Title
Predictive simulation of grain boundary evolution on a finite element mesh

Author
Eren, Erdem

Publication Date
2022

Predictive simulation of grain boundary evolution on a finite element mesh

By

ERDEM EREN
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Materials Science and Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Jeremy K. Mason, Chair

Jeffery C. Gibeling

Michael R. Hill

Committee in Charge

2022

To my parents and all my teachers,
for planting and nourishing the seed of curiosity.

CONTENTS

List of Figures

List of Tables

ABSTRACT

Predictive simulation of grain boundary evolution on a finite element mesh

The topological transitions of the grain boundary network during grain growth under uniform grain boundary energies are believed to be known. However, this is not true for more realistic materials with varying grain boundary energies that, in principle, allow many different grain boundary configurations. A grain growth simulation for such a material therefore requires a procedure to enumerate all possible topological transitions and select the most energetically favorable one. Such a procedure is developed and implemented here for a microstructure represented by a volumetric finite element mesh. The method is implemented as a C++ library called VDlib based on SCOREC, an open source massively parallelizable library for finite element simulations with adaptive meshing. To test the implementation, a Voronoi tessellated microstructure composed of one hundred grains is generated and evolved under constant grain boundary properties until half of the grains remained. The evolution of the individual grains is compared to what is expected from the MacPherson-Srolovitz relation and is found to be in good match.

As with all numerical techniques, it is important to identify systematic sources of error. Such studies for grain growth simulations are either missing from or not uniform across the literature. To address this issue and enable comparison across grain growth simulations, a set of benchmark cases, one for each boundary type of surfaces, lines, and points with analytical solutions are identified. These are used to compare a recently developed discrete-interface method for microstructure evolution to a state of the art diffuse interface (multiphase field method). In each case, the discrete method is found to meet or outperform the multiphase field method in terms of accuracy for comparable levels of refinement, demonstrating its potential efficacy as a numerical approach for microstructure evolution.

Last, a novel invariant of ideal grain growth process is defined that ideally can be used to calculate the number of topological entities in a microstructure and only changes as the topology changes. VDlib is used to test the invariant on a simulated microstructure composed of a periodic arrangement of Kelvin cells. A convergence study reveals that the invariant converges to the ideal case with increasing levels of refinement, however predicting the number of entities without bounds on number of entities proves unpractical. Two error measures based on this novel invariant are proposed which can potentially be used to quantitavely measure how much simulations deviate from ideal grain growth process and experimental microstructures.

ACKNOWLEDGMENTS

A dissertation requires a great amount of work, but without the efforts of our predecessors and the support of collaborators, friends, and family it is simply impossible.

I'm deeply grateful to my advisor Jeremy Mason for all his support in this long journey, his expertise, the advice he gave on all matters relating to research, and for being a friend. He is one of the most attentive, hard-working, and compassionate persons I have known and I'm extremely fortunate to have met him. I would like to thank Prof. Jeffery Gibeling and Prof. Michael Hill for following the dissertation progress over the years and their suggestions. I will always remember their advice on the importance of preserving the personal connections formed during this journey. I'm grateful to Prof. Ricardo Castro, Prof. Susan Gentry, and Prof. Natarajan Sukumar for being on my qualifying exam committee, for the discussions, and their helpful suggestions. I would also like to thank Prof. Brandon Runnels who despite the circumstances brought by the pandemic was an excellent collaborator with his expertise, responsiveness, and optimism.

I would like to thank Prof. Yayoi Takamura for her help during my transfer from Turkey to the department and for her friendly manner in her interactions with us students in the department. In addition, I would like to thank her and Mr. Jeffrey Erhardt for their fellowship. I hope to follow their example in the future to foster a similar working environment for my colleagues and support students in need.

I'm grateful for having to know all my friends who shared this journey with me. I consider myself very fortunate for the amicable atmosphere I experienced in both Davis and Boğaziçi. Despite all the wonderful friends and the welcoming culture in Davis, one cannot help but miss home. I'm grateful for all my Turkish friends, Çiğdem Hanım, and Zeynep with whom I was able to feel back in Turkey whenever I needed.

Last, I would like to thank my family; I can't imagine this process without them. My parents, Şeref Eren and Nuran Eren were always there for me with their unconditional love and support. My brother Onur Eren and I were on different sides of the Atlantic all these years before I moved to the US, but he was always by my side when I needed him the most. I’m fortunate to have my spouse Oyku share this journey with me through thick and thin. Even when thousands of miles away, feeling her presence helped me to overcome the obstacles I faced.

Chapter 1

Introduction[1]

1.1 Motivation

One of the overarching goals of integrated computational materials engineering (ICME) [2] is to accurately predict microstructure and property evolution during thermomechanical processing. At a minimum this would require a simulation incorporating crystal plasticity and grain boundary motion, and ideally interactions involving multiple phases and other material physics. Such simulations would benefit from recent advances in three-dimensional microscopy [3], and specifically three-dimensional X-ray diffraction microscopy (3DXRD) that enables non-destructive three-dimensional imaging of millimeter-sized samples [4, 5]. These could both provide initial conditions for and allow verification of the output of predictive simulations of microstructure evolution.

Historically, one of the major difficulties with simulations of microstructure evolution has been the use of unrealistic grain boundary energy (GBE) functions. Such functions are difficult to determine experimentally due to the number of independent variables, but Morawiec recently suggested a procedure to estimate the GBE from distributions of grain boundary angles around triple junctions [6]. Saylor et al. subsequently used a related technique to estimate the GBE from EBSD analysis of the surface of aluminum samples [7, 8]. While explicit functions for the grain boundary energy are not yet widely available (with a few exceptions [9, 10]), this will likely change in the near future. When

---

[1] The majority of this chapter has been previously published in Ref. [1].

that happens, a code for microstructure evolution that is able to make full use of them would ideally already be available. Existing simulations of microstructure evolution include Monte Carlo (MC) Potts, cellular automata (CA), phase field (PF) and front tracking models, though these do not satisfy the requirements of the problem as is described in more detail below.

1.2 State of the art

1.2.1 Implicit interface models

The MC Potts [11, 12] and CA [13–15] methods are popular partly because of their low computational complexity and ease of implementation, but suffer from two relevant shortcomings. First, the underlying voxel lattice introduces an artificial anisotropy that can be difficult to overcome [16, 17], and a predictive model should respect the isotropy of space with kinetics that are relatively independent of the underlying grid. Second, it can be difficult to connect the MC Potts and CA models with physical units of measure. Zhang et al. scaled quantities defining characteristic time, length and energy but observed that the grid size affected the bulk energy driving force [18]. Mason established spatial and temporal dimensions in a CA model using the Turnbull relation and a uniform grain boundary energy, but the technique is not easily generalized to other situations [17].

The phase field method is an implicit boundary approach that was initially developed to study phase transitions [19], and can be modified to include small deformations and mildly anisotropic interface energies [20]. One drawback is the high memory and computational demand associated with representing grains by continuous fields, since numerical instabilities associated with steep gradients limit the time step. Modern implementations often use sparse data structures [21–23] and adaptive meshing [24] to address this issue. Still, finite deformations and arbitrary boundary energies that can depend on the grain boundary plane pose difficulties. Moreover, the use of diffuse boundaries can complicate the study of topological aspects of the grain boundary network and can introduce subtle numerical errors. Jin et al. compared the accuracy of level set and phase field methods coupled with the Finite Element Method (FEM) in representing the motion of triple lines during isotropic and anisotropic grain growth [25]. They observed that under proper grid and time refinement, both methods performed similarly for the isotropic case. For anisotropic grain growth though they observed 14.2% error in triple junction velocity for the level set method and as much as 68.7% error for the PF method. Some recent variants allow for anisotropic grain boundary properties (GBP) [26], but the modeling of finite mechanical deformation has still not been addressed.

1.2.2 Explicit interface models

Early front tracking methods had the advantage of concentrating computational resources just on the boundaries, and were often used to study mean curvature flow [27, 28]. FEM-based approaches are a natural extension of these that can support additional physics, e.g., boundary energies can be explicitly defined and volumetric meshes allow for crystal plasticity [29]. However, FEM-based methods introduce additional challenges with scalability and require explicit handling of the topology and mesh. These complexities have encouraged use of an MC Potts, CA or PF method in conjunction with a FEM solver. Such hybrid schemes use an implicit boundary representation to model grain growth, and transfer the resulting microstructure to the FEM to model deformation. Sequential evolution is achieved by transferring the microstructure back and forth [30–32]. This does not resolve accuracy concerns though, since transferring the solution potentially introduces information loss and increases computational complexity.

1.2.2.1 Finite element based techniques

Of the purely FEM-based approaches, Kuprat developed a three-dimensional gradient-weighted moving finite element (GWFE) method and implemented GRAIN3D, a serial finite element framework for microstructure modeling of grain growth [33]. The code uses an element regularization scheme to improve low-quality elements, handles changes in the microstructure as boundaries evolve, and supports volumetric physics. While the initial implementation only supported constant grain boundary energies, more general energies were investigated by Gruber et al. [34]. There are two main concerns with using this for general purpose simulations of microstructure evolution though. First, Kuprat implemented the topological transitions by switching the last remaining set of elements of a collapsing boundary segment or volume to the appropriate neighboring volumes [33]. This is not necessarily physical, and the relabeling can cause a substantial and artificial perturbation of the boundaries. Although the likely changes to the overall evolution are limited for an isotropic grain boundary energy, this could substantially affect the microstructure trajectory for the anisotropic case. Second, the existing implementation of the implicit finite element solver is serial. This prohibits simulating microstructures on physically relevant scales, such as the 1 mm^3 cylindrical copper sample imaged using 3DXRD by Li et al. [5].

Using a surface mesh representation, Syha and Weygand studied the effects of an anisotropic grain boundary energy [35]. They proposed to decompose topological transitions into simpler sequential operations and used a force-based criteria to select changes to the grain boundary network. While this could accommodate anisotropic grain boundary energies, decomposing a topological transition into a sequence of simpler ones could alter the eventual trajectory of microstructure evolution. Moreover, the implementation is not volumetric and cannot support volumetric physics.

Lazar et al. studied ideal grain growth by using a surface mesh representation, a fixed set of topological transitions applicable for uniform grain boundary energy, and evolving the microstructure with a discretized version of the MacPherson–Srolovitz relation [36, 37]. This approach provided insight into ideal grain growth but is not applicable to general microstructure evolution for two reasons. First, the boundary evolution formulation assumes that the microstructure is composed of quadruple points and triple junctions at all times except for the moments where transitions occur. While this is generally applicable for ideal grain growth, it does not hold for experimental microstructures. For instance, highly twinned microstructures often contain stable junction lines joining four grain boundaries, and accommodating such configurations would require implementing more general topological transitions. Second, the implementation doesn't support volumetric physics and is only intended to model ideal grain growth.

A general-purpose FEM code for ICME would ideally be able to handle substantial volumes of material since many grains are required to accurately reflect variations in the local deformation response and model stochastic processes like recrystallization. Tucker et al. studied the convergence of large scale crack propagation simulations as a function of the number of grains and mesh refinement in microstructures with abnormal grains [38]. They observed that the overall damage response was not significantly affected by mesh resolution, but that more than 200 grains were required in the sample microstructure the local response to converge. This shows that a scalable framework is necessary to accurately capture the local response during deformation.

In summary, existing implementations of FEM-based grain growth codes are limited in several respects. First, they are generally serial, prohibiting large scale simulations [33, 35, 36]. Second, topological transitions are achieved by merging mesh entities with one of the neighboring grains [33], by sequentially splitting points [35], or selecting from a restricted set of operations [36], all of which could substantially change the microstructure evolution trajectory. That is, a general-purpose FEM code to study grain growth and deformation at physically relevant scales does not appear to exist.

1.3 Summary of contributions

The main goal of this dissertation is to enable predictive simulations of grain boundary evolution on a volumetric finite element mesh. Chapter 2 describes the underlying topological and mesh representations. Chapter 3 gives the details of the method and of the resulting implementation. Chapter 4 describes a set of benchmarks that can be used to quantify the convergence of error when modeling grain growth and to enable comparative studies of different simulation methods. The benchmarks are used to compare our discrete interface method with a state-of-the-art diffuse interface method. Chapter 5 proposes a novel invariant of motion for ideal grain growth processes that only changes with topological transitions and ideally can be used to uniquely calculate the number of topological entities. Chapter 6 describes possible extensions and future research directions.

Chapter 2

Computational approach

The vision for a general purpose microstructure evolution framework pursued in this dissertation is based on a finite element mesh with three key requirements:

1. scalability to reach representative volumes of material,
2. an adaptive meshing capability to preserve element quality as grain boundaries move, and
3. a volumetric mesh to support arbitrary physics (i.e. crystal plasticity).

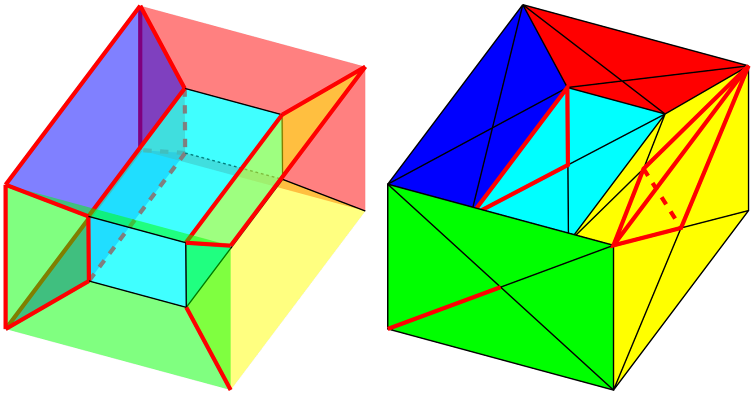
SCOREC is an open source massively parallel finite element management library [39] that satisfies all three of these requirements, but does not natively allow any changes to the grain boundary network. Section 2.1 describes the topological and the underlying mesh representations. Section 2.2 describes the mesh generation approach, Sec. 2.3 describes the mesh partitioning process, and Sec. 2.4 describes the mesh adaptation approach that SCOREC uses. Section 2.5 describes the VDlib code structure that follows the flow of a typical simulation.

2.1 Microstructure representation[1]

Our purpose is to simulate microstructure evolution at a scale that resolves the grain structure. It will be useful in the following to introduce specific terminology to identify the various microstructure components. A grain will be called a volume, a boundary

---

[1] The content of this section has been previously published in Ref. [1].

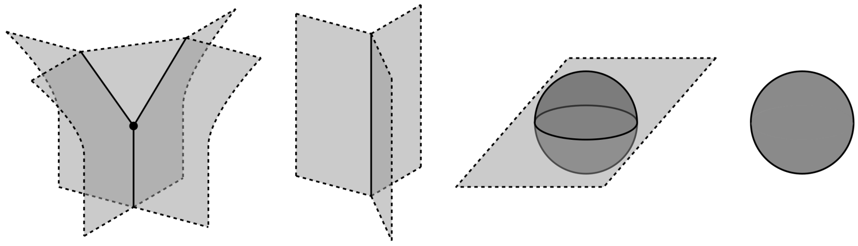


(a) (b)

Figure 2.1. (a) An example grain structure composed of seven grains. A central rectangular grain is surrounded by six grains, with examples of a volume, surface, and line outlined in red. (b) A finite element representation of this grain structure where tetrahedra, triangles, edges and vertices are used to discretize volumes, surfaces, lines and points. Examples of a tetrahedron, triangle and edge are outlined in red.

a surface, a boundary junction line a line, and a boundary junction point a point. A microstructure where each of these components is outlined in red is shown in Fig. 2.1a. The volumes, surfaces, lines, and points composing the microstructure formally comprise a stratified space, and for that reason the microstructure components will occasionally be referred to as d-strata where d is the dimension of the stratum. The connectivity of the topological components of the microstructure is defined by the adjacencies of d-strata and $(d-1)$-strata; that is, a volume is bounded by surfaces, surfaces by lines, and lines by points.

A point is required to bound at least three lines (Fig. 2.2a), a line at least three surfaces (Fig. 2.2b), and a surface exactly two volumes. One can show that any topological component not satisfying these relationships is spurious and it can be removed by merging the adjacent components of the next higher dimension. There are no constraints imposed on the number of adjacent components of the next lower dimension; this allows, e.g., a small spherical volume to be embedded in the middle of a surface (Fig. 2.2c), or a ball to be embedded in the interior of a volume (Fig. 2.2d).



(a) (b) (c) (d)

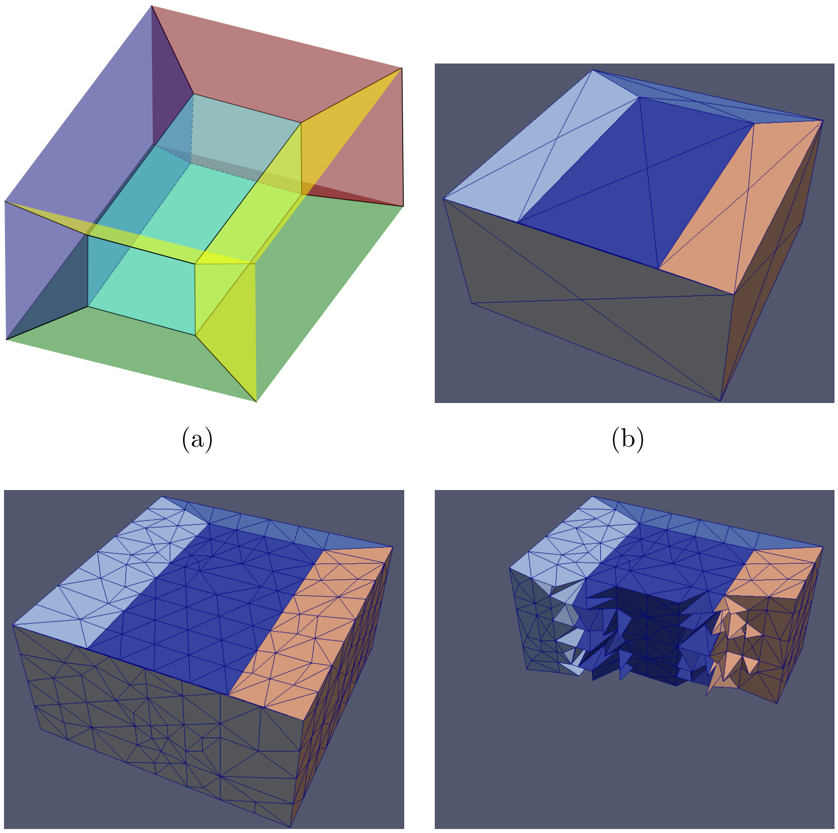
Figure 2.2. Examples indicating adjacency rules. (a) A point should bound at least three lines. This point bounds three lines and four volumes, two on the left and right and two in front of and behind the page. (b) A line should bound at least three surfaces. (c) A surface separating a top and a bottom volume and ball embedded in the surface. The line of intersection has no bounding points. (d) A sphere inside another volume, with a surface that has no bounding lines.

2.2 Mesh generation

Any synthetic microstructure can be used to test the method proposed in Ch. 3. Since SCOREC has no inherent capacity to generate microstructures or meshes, two methods of microstructure generation are employed within this project. First, Neper [40] is an open source microstructure generation toolbox that can generate regularized Voronoi tessellations that resemble grain microstructures. Second, custom MATLAB scripts can be used to generate test microstructures compliant with the Neper file format, with one example being the microstructure shown in Fig. 2.1a. VDlib has the ability to generate a mesh from a Neper microstructure file by using a simple routine that places a single vertex on the interior of each boundary line, surface, and volume, and subsequently connects these vertices to form edges, triangles and tetrahedra. SCOREC can subsequently refine this mesh based on a general anisotropic metric. An open source software called DREAM.3D [41] is also capable of generating synthetic microstructures based on empirical statistics or experimental EBSD data, but it is not used as part of this project.

2.3 Mesh partitioning

For optimal parallelized performance it is important to maintain an equitable mesh partitioning. The steps of microstructure meshing, refinement, and partitioning are shown in Fig. 2.3 on a microstructure containing a rectangular prismatic grain surrounded



(a) (b)

(c) (d)

Figure 2.3. a) A sample rectangular prismatic microstructure. b) The mesh generated by the custom script. c) The refined mesh obtained using MeshAdapt. d) One part of the partitioned mesh.

by six other grains. Although partioning is necessary to reach representative material volumes, there are certain limitations on the computational gains that can be obtained by parallelization; Amdahl's argument indicates that the theoretical performance improvement is limited by the portion of the process that needs to be executed serially [42]. Moreover, Amdahl's argument doesn't consider the cost of communication between parallel processes, further limiting the useful degree of parallelization in a simulation. Parallelization also complicates the implementations of mesh adaptation, physical processes, and topological transitions.

SCOREC is fully parallelized [43] and represents the mesh as a set of partitions with overlapping regions, with one partition shown in Fig. 2.3d. Although VDlib is built on SCOREC, the topological transitions could not be fully parallelized within the time frame of this dissertation. Parallel implementations of physical processes or equations of motion using SCOREC's interface were achieved in some test programs, but the necessary changes were not included in VDlib.

2.4 Mesh adaptation

An important concern of mesh management is the preservation of the mesh or element quality. The movement of vertices in a deformable mesh changes the shape and size of the elements over time, leading to two undesired effects. First, the change in element shape can lower element quality, increasing the stiffness of the differential formulation and reducing the stability and floating point accuracy of the calculations. Second, the change in element size can affect the fidelity of the simulation as certain regions become over-refined and others become coarsened.

Remeshing is the standard solution to these difficulties. One approach is to parameterize the grain boundary representation and globally remesh the structure. Although this can allow better control over the positions and alignment of mesh vertices, global remeshing is costly and can perturb the grain boundary mesh if not done carefully. The other approach is to identify regions that require remeshing and apply local changes which are less costly, easier to scale, and minimally perturb the grain boundaries. SCOREC implements local adaptive meshing through a MeshAdapt module. This is further broken up into three main operations applied in sequence through an adaptation loop: coarsening, refinement, and quality improvement [39].

SCOREC decides to locally coarsen or refine the mesh by means of a metric on edges. Usually an edge is coarsened (refined) if its length is less than (greater than) a multiple of a length threshold that can be anisotropic and vary over the space. The criterion for coarsening is not restricted to use of a metric on edges though, and can be customized to include topological information and fields defined over elements.

After the coarsening and refinement operations, MeshAdapt iterates over tetrahedra and detects those below a quality threshold. To improve the quality of a tetrahedron, a cavity or collection of immediate entities around the tetrahedron is formed. The mesh

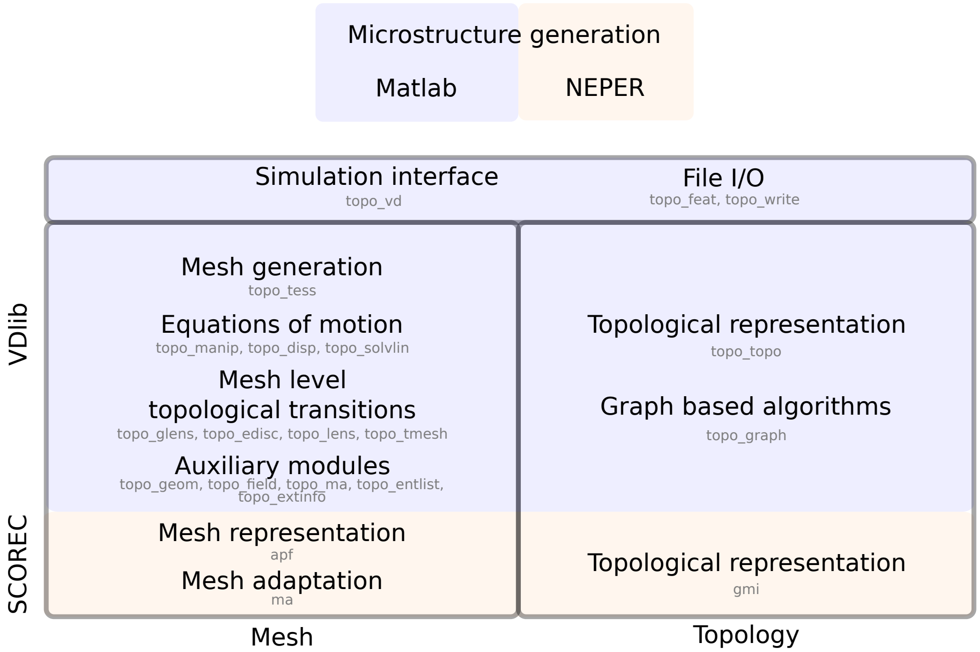


Figure 2.4. The hierarchy of modules within VDlib. The code developed within the project is colored in blue, and the code that is interfaced with is colored in orange.

within the cavity is modified if the quality can be improved by any of the following operations: edge removal, 2-to-3 face swapping, edge collapse, edge split, face slip, or region split. However, MeshAdapt sometimes creates extremely low quality, pancake-like elements near boundaries with small curvatures that can lock the simulation. Although not described here, additional operations are implemented within VDlib to remedy such low quality elements. These are mainly achieved by refining the interior of the low quality elements and their neighbors such that MeshAdapt can find cavities to improve overall element quality. When these don't resolve the element quality, an operation is applied to collapse the low quality tetrahedron to a boundary along its shortest edge, along the altitude of the smallest area triangle, or along the altitude of the tetrahedron itself depending on the geometry.

2.5 Code structure

The structure of the code used in a typical simulation is shown in Fig. 2.4. The simulation starts by the generation of a microstructure using either MATLAB scripts or Neper. The simulation interface accepts a microstructure representation (either a Neper microstructure file or saved SCOREC mesh instances) together with option files for mesh adaptation, equations of motion, and information extraction. A corresponding mesh is generated by VDlib (if necessary) and evolves under the user-specified equations of motion. Any topological transitions are detected and handled at the topological and mesh levels. These operations require auxiliary modules for geometric calculations and to communicate with SCOREC.

Chapter 3

Method[1]

This chapter describes the method developed to handle topological transitions involving the grain boundaries on a simplicial finite element representation of a microstructure. The resulting method and the corresponding implementation makes four practical contributions to the literature. First, a method for finding all possible topological transitions that can occur around junction points during grain growth is proposed. Second, operations on the simplicial mesh have been developed to modify the mesh corresponding to these topological transitions. Third, a criterion based on the energy dissipation rate is used to compare different topological transitions, providing an unambiguous selection criterion. Fourth, a discrete formulation to simulate grain boundary motion has been implemented that allows for effectively arbitrary grain boundary properties [44]. The formulation is explicit and solves for the motion of each vertex individually, reducing the computational complexity compared to the weak formulation of the FEM at the cost of increased error. An open-source C++ library called VDlib [45, 46] implements all these operations and interfaces with SCOREC [39, 43], a massively parallel mesh management library with local adaptive re-meshing.

The process of code development and the capabilities of VDlib described below directly resulted in three scientific contributions. First, the process of finding all possible topological transitions for a canonical configuration of five grains around a junction point revealed two possibilities that are likely not mentioned anywhere in the literature, and that are

---

[1] The content of this chapter has been previously published in Ref. [1].

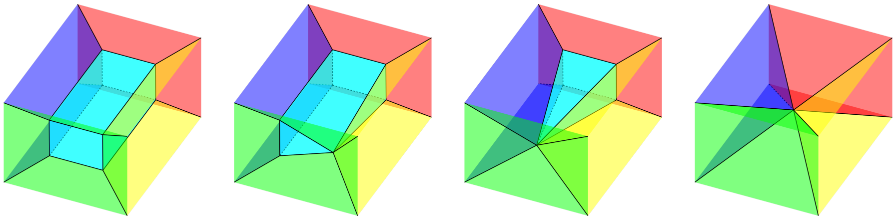

(a) (b) (c) (d)

Figure 3.1. The cases of collapse shown on the rectangular prism example. (a) The initial microstructure. (b) Line collapse. (c) Surface collapse. (d) Volume collapse.

named mixed removal and digon insertion type-II in Sec. 3.5. Second, the spurious stratum insertion discussed in Sec. 3.2.3 is established as one of the essential topological transitions for the progress of a grain growth simulation. Third, Sec. 3.4 generalizes the MacPherson–Srolovitz relation [37] to allow for fixed boundary conditions. Since our purpose is to provide the foundations for large-scale simulations of realistic microstructure evolution, we expect that the capabilities of our open-source library will enable the community to realize further related scientific advances in the future.

Sections 7.2.1 - 7.2.12 in the appendix provide detailed explanation of the topological algorithms, mesh level operations, modifications to handle exterior insertions, insertions on a six grain configuration, spurious insertions, and the scaling of the computational resource usage with number of grains.

3.1 Operations on the microstructure

Over the course of grain growth, grain boundaries migrate to reduce the energy of the microstructure. Occasionally a surface or volume will shrink to a point or will expand from a point to participate in the subsequent evolution; such events are called topological transitions. From the standpoint of the finite element mesh the corresponding operations are either collapses, where disappearing boundary segments or volumes are removed, or insertions, where new boundary segments are introduced to allow the microstructure evolution to continue.

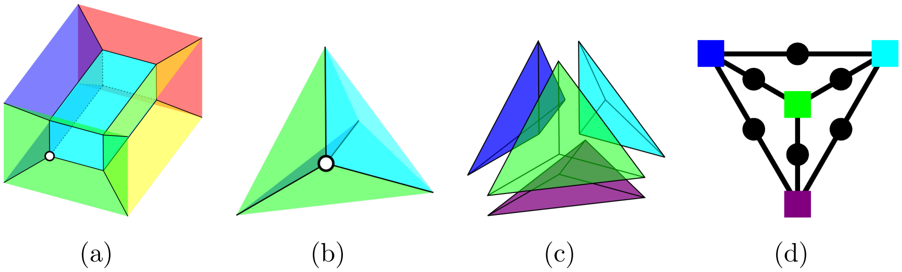

(a) (b) (c) (d)

Figure 3.2. (a) Consider the point at the bottom left corner of the central volume. (b) The neighborhood of the point shows the relationships with the surrounding surfaces and volumes. (c) The volumes in an exploded view. (d) The adjacency graph showing the volumes as squares and the surfaces as disks. In this figure, volumes and squares are the same color.

3.1.1 Stratum collapses

The average grain size increases during grain growth, meaning that the general trend is for components of the grain boundary network to vanish. The criterion for this topological transition in practice is that the length of a line, area of a surface, or volume of a grain is shrinking and passes below a threshold related to a characteristic microstructural length scale, e.g., the average grain diameter. The collapse is effected by merging all of the bounding points and adjusting the adjacency lists of the surrounding components as appropriate. Examples of this operation are shown in Fig. 3.1, with several specifics of the algorithm given in Sec. 7.2.2 of the appendix.

3.1.2 Stratum insertions

Often the configuration resulting from a stratum collapse is unstable and the energy could be lowered by splitting the point to insert a line or a surface. There are usually many such possible insertions, and selecting the most likely one necessarily involves first identifying the various possibilities. This analysis can be performed using the adjacency graph of surfaces and volumes. The adjacency graph is constructed by placing a node for each volume and surface and an edge between adjacent volumes and surfaces. The steps involved are shown in Fig. 3.2 for a particular point. Formally, for non-periodic microstructures, there is a volume surrounding the simulation cell that is connected to the surfaces bounding the simulation cell. For the purpose of identifying the possible insertions, this is treated similarly to the volumes within the simulation cell, with the

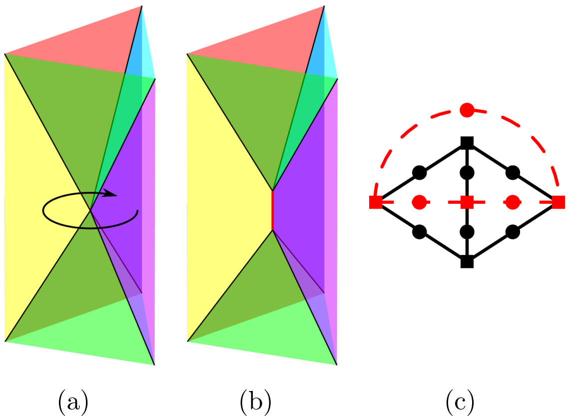


Figure 3.3. A line insertion corresponds to a circuit on the adjacency graph. (a) A five grain configuration and a circuit going around the point. (b) Every surface punctured by the circuit is extended by adding the inserted line to their adjacency lists. (c) The adjacency graph around the point. Edges along the circuit are dashed.

specifics given in Sec. 7.2.8 of the appendix.

3.1.2.1 Line insertions

Every possible line insertion corresponds to a circuit on the associated adjacency graph, with an example shown in Fig. 3.3. This configuration frequently occurs for isotropic grain boundary energies, e.g., when a triple line collapses and two quadruple points are merged. The circuit shown in Fig. 3.3a passes through the front, left and right volumes, and every surface that is punctured by the circuit is adjacent to the inserted line. The circuit in Fig. 3.3a precisely corresponds to the circuit in Fig. 3.3c, and enumerating all possible line insertions is equivalent to enumerating all circuits on the adjacency graph. Algorithms to identify the circuits on a graph are available in the literature [47, 48]. Not all possible circuits need to be considered though; if removing the nodes and edges of the circuit from the adjacency graph leaves only a single connected component, then the line insertion would create a spurious line and point that would subsequently be removed. The resulting algorithm is described in detail in Sec. 7.2.3 of the appendix.

3.1.2.2 Surface insertions

Around a point a surface can only be inserted between two disconnected volumes. Given a pair of such volumes, the inserted surface is connected to the surrounding surfaces by some set of inserted lines. Each line corresponds to a path that starts on one of the disconnected

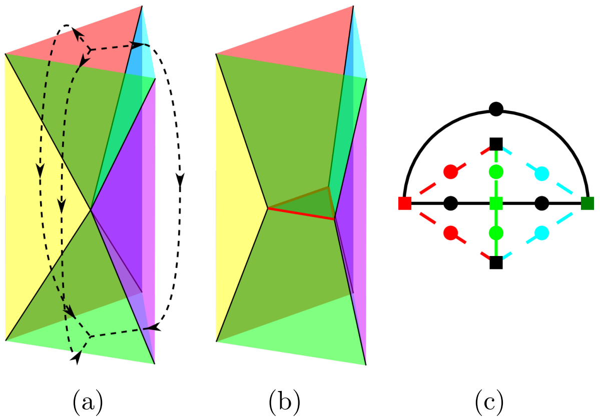

(a) (b) (c)

Figure 3.4. A surface insertion corresponds to a set of paths on the adjacency graph. (a) A five grain configuration, showing a set of three non-intersecting paths connecting the disconnected (top and bottom) volumes. (b) A surface is inserted between the disconnected volumes with one bounding line for each path. Each line is added to the adjacency lists of the surfaces punctured by the corresponding path. (c) The adjacency graph around the point. The color of punctured surfaces and edges on the graph match on (a) and (c).

volumes and ends on the other, as in Fig. 3.4a. A set of such paths completely specifies the topology around the inserted surface. Every surface punctured by a path is adjacent to the corresponding inserted line, as in Fig. 3.4b. The set of all possible surface insertions can be found by constructing all possible sets of non-intersecting paths between the nodes of the adjacency graph corresponding to the disconnected volumes. These paths can be found using a standard depth first search algorithm on the adjacency graph. Unlike line insertions, paths along surfaces that share a common edge are still acceptable, as the newly inserted line will bound the inserted surface and will be topologically different from any preexisting line. The resulting algorithm is described in detail in Sec. 7.2.4 of the appendix.

3.1.3 Other considerations

The algorithms described in this section are conjectured to result in sets of topological transitions that include all those that occur during grain growth for a generic initial condition, even with anisotropic grain boundary energies. A generic initial condition is one for which the type of topological transition shown in Fig. 3.5 does not occur. That is, the only allowed topological transitions are those for which the length of a line, the area

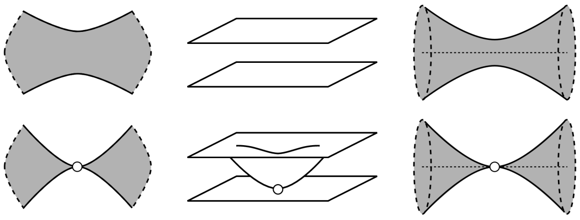

(a) (b) (c)

Figure 3.5. Topological transitions not considered here. (a) Two lines bounding the same surface meet to form a new point. (b) Two bounding surfaces of a volume meet to form a new point. (c) The cross section of a cylindrical volume is reduced to a point.

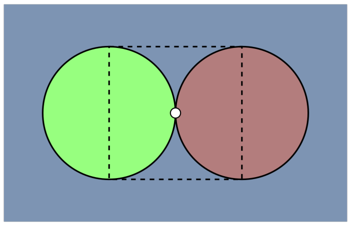


Figure 3.6. A point connected to two spherical grains, and two grains in front of and behind the page. The neighborhood of the point is outlined by a dashed line. The surface in the page is represented twice in the neighborhood of the point.

of a surface, or the volume of a grain passes through zero. This is not believed to be a serious constraint though, since the topological transitions in Fig. 3.5 are not expected to occur during grain growth in a generic physical system.

There are some situations where the adjacency graph of the strata does not accurately reflect the topology around a point. For example, a single point could appear on the boundary of a surface more than once, as in Fig. 3.6. This is the reason that the adjacency graph is constructed using only the microstructure components in a small neighborhood of the point. This can allow spurious insertions (in the sense of Sec. 2.1) that are nevertheless required by the physical system, and any spurious strata can easily be removed after the topological transition is complete. The detection algorithm for spurious strata is provided in Sec. 7.2.10 of the appendix. The construction of a small neighborhood necessarily involves the mesh, and will be considered further in Sec. 3.2.3.

3.2 Operations on the mesh

The Scientific Computation Research Center (SCOREC) is based at the Rensselaer Polytechnic Institute, and develops mesh-based codes for multiscale systems design and engineering [39, 43]. Since the core SCOREC library does not natively support changes to the topology of the finite element mesh, a set of fundamental and localized operations are proposed and implemented. Given that the microstructure is represented by means of a finite element mesh, the individual microstructure components are comprised of sets of simplicial mesh elements. These mesh elements will be referred to as tetrahedra, triangles, edges, and vertices, or occasionally as d-simplices when that is simpler. The distinction between the topological and geometric components of the microstructure is reinforced in Fig. 2.1.

Applying the stratum collapse and insertion operations described in Sec. 3.1 on a simplicial finite element mesh requires some mesh modifications, both to prepare the mesh for these changes and to execute them. The two basic operations are lens collapse and lens expansion, associated with stratum collapse and insertion, respectively. The lens split is an additional operation used to prepare the mesh around a stratum before stratum collapse or in the neighborhood of a point before stratum insertion. While the actual collapse and insertion operations are more complex than those described below, the underlying approach is the same.

Recalling that the set of volumes, faces, lines and points and their connections compromise a topological structure called a stratified space, microstructural components will be called strata in this section, i.e., a volume will be called a 3-stratum, a surface will be called a 2-stratum, a line will be called a 1-stratum, and a point will be called a 0-stratum. For brevity, S^d will denote a d-stratum and S^d_i more specifically the ith d-stratum.

3.2.1 Stratum collapse

An S^d with $d > 0$ is represented by a collection of e-dimensional mesh entities with $e = 0, 1, \ldots, d$. Collapsing an S^d is equivalent to collapsing its constituent entities onto a single vertex. This can be further simplified to collapsing all edges within the S^d and its bounding strata, giving the central idea of stratum collapse. For simplicity, this section

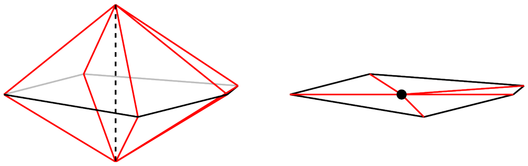


Figure 3.7. Lens collapse operation. Left, the lens composed of tetrahedra and triangles bounded by the collapsing dashed edge. Right, the disc obtained by collapsing the lens.

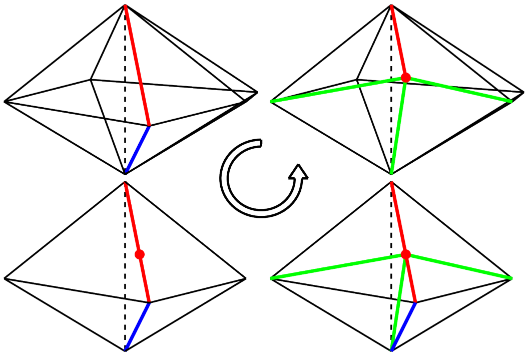
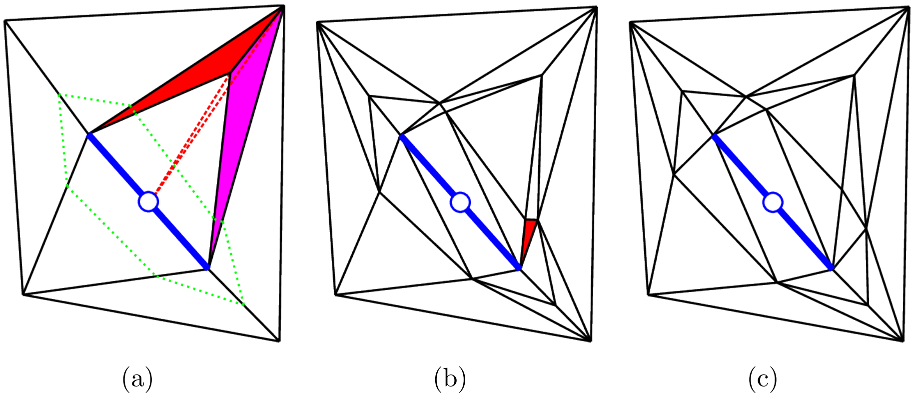


Figure 3.8. Edge split operation during preconditioning. The thicker edges in red and blue belong to strata S_i^d and S_j^e, respectively. If S_i^d and S_j^e are not the same and one doesn't bound the other, collapse of the dashed vertical edge is not allowed. Splitting the red edge and all entities that are bounded by that edge into two creates new entities which by construction either belong to S_i^d or strata bounded by S_i^d.

describes the procedure for a single collapsing edge, with the extension to stratum collapses involving multiple collapsing edges in Sec. 7.2.7 of the appendix.

On a simplicial mesh, an edge bounds a collection of tetrahedra and triangles forming a lens around that edge. As shown in Fig. 3.7, the entities that are bounded by the collapsing edge will also collapse and need to be removed. For each collapsing triangle, the other two bounding edges form a merging couple. For each collapsing tetrahedron, the two triangles that are not collapsing form a merging couple. After the collapse, a new entity is generated for each merging couple. Such an entity belongs to the lower dimensional stratum of the merging couple, assuming the merging entities belong to the same or adjacent strata.

Three issues could arise that would invalidate the mesh during a stratum collapse. First, the collapse could cause an additional topological transition if any of the merging entities do not belong to the same or adjacent strata. Applying the edge split operation shown in Fig. 3.8 to one of the edges of the problematic couple resolves this situation.



(a) (b) (c)

Figure 3.9. The effect of preconditioning for an S^1 collapse on a two-dimensional mesh. (a) Collapsing the blue S^1 and moving the vertices to the blue node would invert the red triangle and merge it with the purple triangle. The resulting triangle is shown in dashed lines. (b) The splitting procedure resolves this problem, but yields the red triangle that could invert during collapse. (c) Relaxation allows the S^1 to be collapsed without inverting any elements.

Second, it is possible that two d-dimensional entities could unintentionally merge. This could occur even if they do not belong to the the collapsing lens, but requires that they share $d - 1$ vertices and that the remaining vertex of each be a distinct merging vertex as in Fig. 3.9a. The edge split procedure can also resolve this by isolating the collapsing entity, as shown in Fig. 3.9b. A third issue that would invalidate the mesh is inversion of one of the surrounding entities during a collapse. This could occur if the initial and final positions of a merging vertex lie on distinct sides of the plane containing the opposite triangle of an adjacent tetrahedron.

The three-dimensional equivalent of the preconditioning operation in Fig. 3.9 is applied to edges that are adjacent to a single merging vertex to avoid all three situations. First, the midpoints of all edges emanating from the merging vertices are collected to compute their convex hull, and the emanating edges are split where they intersect the convex hull. This resolves the first two issues and yields a hull of triangles surrounding the collapsing stratum. While it is still possible for a surrounding tetrahedron to invert during the collapse, a relaxation procedure analogous to that in Fig. 3.9c and described in Sec. 7.2.6 of the appendix is applied to vertices on the hull to avoid such an event. After preconditioning, the stratum memberships of the new entities associated with the merging entities are found. A new entity belongs to the lowest dimensional stratum that owns one of the merging entities; the preconditioning certifies that there is a single stratum of the lowest dimension.

During the course of microstructure evolution, the criterion for collapsing a stratum is decided at the mesh level with a two step algorithm. First, the diameter of a stratum is approximated as that of an edge, square or cube with the same length, area or volume, respectively. If the diameter of a S^d is smaller than a threshold, then the time rate of change of the total length, area, or volume of the collapsing stratum is calculated using the velocities associated with the bounding vertices. If this is negative, then the stratum is collapsed.

3.2.2 Stratum insertion

As described in Sec. 3.1.2, the insertion of a S^1 or S^2 around a central S^0 initially involves finding circuits or paths in the adjacency graph of surfaces and volumes. For this to work on the mesh level, there should be at least one internal edge in each of the surrounding S^2 and S^3. This is ensured by two operations. First, a lens expansion is applied to each connected set of tetrahedra belonging to the same S^3. The S^2 triangles bounding such a set and adjacent to the S^0 form a disc that can be expanded. The expansion forms a new vertex, a new edge and a set of new triangles and tetrahedra corresponding to the disc triangles, all belonging to the specified S^3. Second, if there are any sets of connected triangles belonging to a S^2 that consist of a single triangle, the edge opposite the S^0 is split. Next, the split operation is applied to the edges bounded by the central vertex belonging to the S^0. The vertices created by these split operations are positioned on a sphere centered at the S^0 vertex location. The radius ρ of the sphere is smaller than the distance to the closest triangle opposite the central vertex in any surrounding tetrahedron.

Preconditioning achieves three things. First, it ensures that corresponding sets of triangles and edges can be found for each circuit associated with a S^1 insertion and each path associated with a S^2 insertion. These sets of triangles and edges form disc- or fin-like structures. Second, it forms a convex cavity of triangles, preventing element inversion after the insertion. Third, it reduces the size disparity of the surrounding triangles and

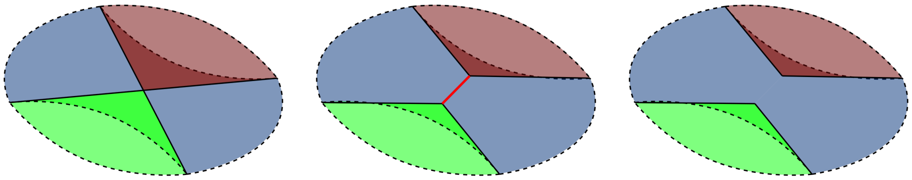

(a) (b) (c)

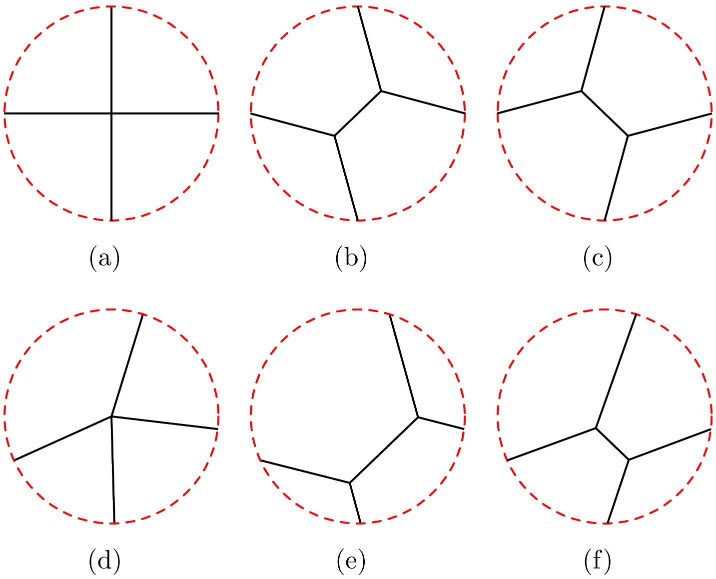
Figure 3.10. Steps of a spurious line insertion. (a) A point, connected to the red and green volumes above and below and two volumes in front of and behind the page. (b) Insertion of the spurious line, adjacent to two surfaces both separating the same volumes. (c) The spurious line is removed and the two surfaces are merged.

the associated bias in the numerical scheme for vertex velocities.

Stratum insertion requires expansion of a disc/fin, creation of triangles and tetrahedra with the same stratum memberships as the edges and triangles on the disc/fin, and creation of edges and triangles belonging to the new strata. In the case of a S^1 insertion, a new S^0 vertex and a new S^1 vertex to be positioned on the interior of the new line are created. The disc associated with the circuit is used to create three discs, one for the old S^0 vertex, one for the new S^1 vertex, and one for the new S^0 vertex such that the disc entities belong to the same strata as in the initial disc. Two new S^1 edges are created to connect the S^1 vertex to the bounding S^0 vertices. The volume between the discs and around the new S^1 edges is filled by triangles and tetrahedra corresponding to edges and triangles on the discs. In the case of a S^2 insertion, the entities bounded by the new S^2 entities need to be generated. A triangle belonging to the new S^2 is generated for each new S^1 edge, and a new tetrahedron belonging to the adjoining S^3 is generated for each new S^2. When inserting strata on a S^0 on the boundary of the simulation, the algorithm skips the creation of entities for the exterior S^3. The final step of the insertion is the relaxation described in Sec. 3.3.

3.2.3 Spurious stratum detection and insertion

If an inserted stratum has fewer than the minimum number of higher-dimensional adjacencies, it is spurious and is removed by merging the higher-dimensional adjacencies. An example is given in Fig. 3.10. This operation is sometimes necessary, e.g., when a S^0 is connected to multiple disjoint sets of triangles belonging to the same S^2 or disjoint sets


(a) (b) (c)

(d) (e) (f)

Figure 3.11. The choice of insertion can change the overall trajectory of the system. (a) A two-dimensional degenerate configuration with four grains could transition to either (b) or (c) since they are energetically equivalent. For (d), (e) and (f) both lower the energy, but (e) more so.

of tetrahedra belonging to the same S^3. In this situation, the global connectivity of the stratification is not representative of the possible local insertions around the vertex. A local stratification of disjoint sets of entities belonging to the same stratum is generated, and the set of all possible insertions is found with the same circuit and path detection methods as described above. A practical situation where spurious insertions are necessary for the progress of a grain growth simulation is described in Sec. 7.2.10 of the appendix.

3.3 Boundary evolution and energy criteria

When inserting a new stratum, it is important that the geometry of the stratum maximizes the energy dissipation rate as the stratum expands. This is especially important when there is more than one possible stable insertion, as shown in Fig. 3.11. Even for a constant grain boundary energy, inaccurate calculations of the geometry could change the selected insertion and drastically alter the subsequent evolution of the system.

The calculation of the geometry of an inserted stratum begins by isolating the mesh around the old S^0 vertex and applying the relaxation algorithm, as shown in Fig. 3.12 for

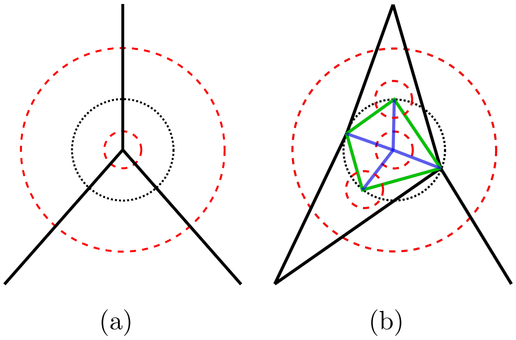

(a) (b) (c) (d)

Figure 3.12. The steps of mesh level insertion and reorientation for a digon insertion. (a) Fins of triangles along paths, shown in bold black. (b) Insertion of the new digon, where S^1 edges are shown as green lines and S^2 edges are shown as blue lines. (c) The vertices are allowed to move until one of the ending criteria is reached. (d) The digon is scaled to be within the projection sphere, and relaxation continues until the energies converge.

a digon insertion. The bold black lines in Fig. 3.12a represent the fins of triangles on the paths. A new digon is inserted by expanding the two selected fins, changing the topology as shown in Fig. 3.12b. The projection sphere of radius r is represented by the black dotted circle and the inner (one for each S^1 and S^2 vertex) and outer bounding spheres are represented by red dashed circles. The vertices are then allowed to move according to the equations of motion (Fig. 3.12c) until a minimum energy is reached or one of the moving vertices intersects an inner or outer bounding sphere. If one of the inner spheres is intersected, the insertion is discarded. If the outer sphere is intersected, the inserted stratum is scaled to be contained within the projection sphere. The steps in Fig. 3.12c and 3.12d are repeated until both the energy at the intersection and the energy after the scaling converge to the final and initial energies E_f and E_i.

Since the thermodynamically-driven system follows a gradient flow of the energy, the physical system will transition to the state with the the highest energy dissipation rate. After the process converges, the energy dissipation rate is calculated for the expanding insertions at the singular configuration where all the new vertices are positioned at the old vertex position. Assuming the contributions of the newly generated strata to the forces acting on the vertices are vanishingly small in this configuration, the dissipation rate of initial expansion is given by

$$W = -\sum_i \boldsymbol{F}_i \cdot \boldsymbol{v}_i$$

where F_i and v_i are the force acting on and the velocity of vertex i and the sum is over all newly inserted bounding vertices.

Our energy dissipation rate criterion is similar to the depinning force which Shya and Weygand use to repeatedly split a node by edge insertions [35]. The difference is that our approach instead compares all possible single stratum insertions at once using the energy dissipation rate criterion, presumably more closely following the evolution of the physical system. Moreover, the relaxation algorithm discards insertions that do not expand, allowing for stable high valency junctions that could form, e.g., at intersecting deformation twins in TWIP steels.

3.4 Modified MacPherson–Srolovitz relation

All numerical approaches should be verified against experimental or analytical results. One possibility for polycrystalline microstructures evolving under constant grain boundary energy is the MacPherson–Srolovitz (MS) relation [37], the three-dimensional extension of the von Neumann–Mullins relation [49, 50]. For a constant grain boundary energy, this relation should be satisfied by each grain at every moment in time except for when a topological transition occurs.

The MS relation [37] governing the rates of change of volumes is given by:

$$\frac{dV(D)}{dt} = -2\pi\mu\gamma \left[\mathcal{L}(D) - \frac{1}{6}\mathcal{M}(D)\right], \tag{3.1}$$

where μ is the constant grain boundary mobility, γ is the constant grain boundary energy, $\mathcal{L}(D)$ is the mean width which measures the the total mean curvature of grain D, and $\mathcal{M}(D)$ is the total length of the triple lines of grain D. Lazar et al. describe a discretized form of the MS relation that can be used to calculate the rate of volume change for grains composed of discretized linear elements [36]. For this case, $\mathcal{L}(D)$ and $\mathcal{M}(D)$ reduce to

$$\mathcal{L}(D) = \frac{1}{2\pi} \sum_i e_i \alpha_i,$$
$$\mathcal{M}(D) = \sum_j l_j,$$

where e_i is the length of the ith boundary edge, α_i is the exterior angle around the ith boundary edge with respect to grain D, and l_j is the length of the jth triple line edge.

The coefficient of $\mathcal{M}(D)$ in Eq. (3.1) is related to the equilibrium exterior angle of $\pi/3$. For periodic boundary conditions and when all junctions are composed of triple junctions and quadruple points, this is the expected exterior angle everywhere. As will be further discussed in Sec. 3.5 though, when using an exterior boundary or allowing higher valency junctions due to the discretized mesh, the MS relation needs to be modified to include more general exterior angle conditions. Specifically,

$$\frac{dV(D)}{dt} = -\mu\gamma \left[2\pi\mathcal{L}(D) - \mathcal{N}(D)\right], \tag{3.2}$$
$$\mathcal{N}(D) = \sum_j \beta_j l_j, \tag{3.3}$$

where β_j is the equilibrium exterior angle around the jth junction line edge. This is determined by the equation

$$(\pi - \beta_j)n = \xi_j$$

where n is the number of grains and ξ_j is the total interior angle available for all grains around the jth junction line edge. For a stable interior S^1, $\xi_j = 2\pi$, $n = 3$, $\beta_j = \pi/3$ and Eq. (3.2) reduces to Eq. (3.1). Assuming a cubic simulation cell, the stable configuration of a S^1 on a simulation cell edge has $n = 1$, $\xi_j = \pi/2$ and $\beta_j = \pi/2$, and the stable configuration of a S^1 on a simulation cell face has $n = 2$, $\xi_j = \pi$ and $\beta_j = \pi/2$. It is possible to have unstable junctions with n larger than that for the stable configurations.

3.5 Results and Discussion

This section describes several tests of our implementation of the preceding ideas in situations where the expected result are known. All possible insertions are identified for

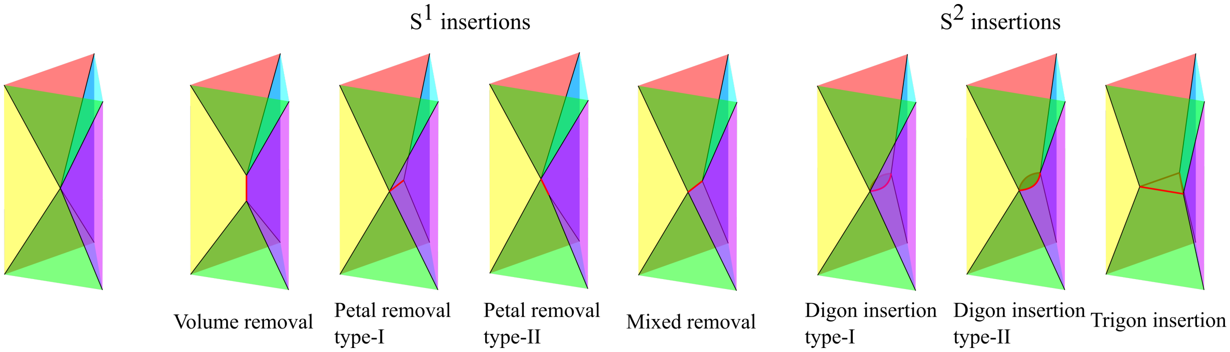
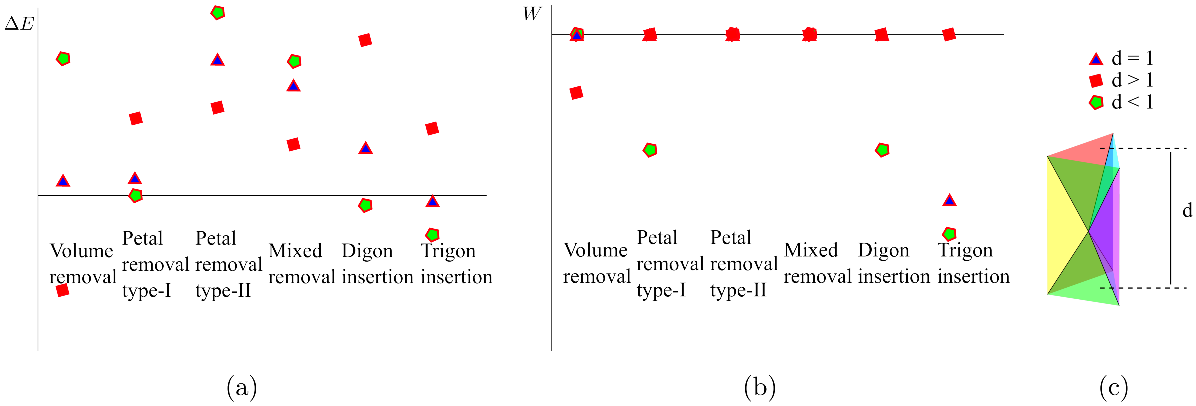


Figure 3.13. All possible insertions for the canonical configuration, classified by symmetry groups. Observe that digon insertions can be obtained by decomposing circuits containing disconnected 3-stratum couples into two paths connecting the couples and using these to insert a 2-stratum. Digon insertion type-I is related to petal removal type-I and digon insertion type-II is related to mixed removal.

the canonical configuration of five volumes around a single point, and include both the insertion of a line and triangle that are considered in previous FEM-based methods and a number of insertions that are not. The effect of the geometric configuration of the surrounding surfaces on the type and geometry of the insertion that maximizes the energy dissipation rate is explored. Finally, a simulation of a trial microstructure is performed as a demonstration of the capabilities of our implementation.

Consider the five grain configuration previously described in Fig. 3.3a. All possible insertions can be found by applying the circuit and path detection algorithms, and these are shown in Fig. 3.13 (classified by their symmetries). There are four classes of S^1 insertions and three classes of S^2 insertions. The volume removal and trigon insertion are generally handled by all grain growth codes, but the other insertions are usually not since a S^1 collapse is always followed by a trigon insertion when the boundary energy is a constant. Digons can also be inserted, with the two types shown in Fig. 3.13.

To be specific, there is one volume removal, three petal removal type-Is, six petal removal type-IIs, and six mixed removals possible, all of which are found by circuit analysis. There are three type-I digon, six type-II digon, and one trigon insertions possible as well. Note that digon insertion type-I and type-II use paths that can be constructed by decomposing the circuits of petal removal type-I or mixed removal, respectively. When discussing the energy dissipation rates, it will be shown that these additional operations



(a) (b) (c)

Figure 3.14. (a) The variation in energy change of insertion with the surrounding boundary configuration. Blue triangles show the energies for the configuration when the S^1 angles in (c) are tetrahedral angles. Red squares denote the energies for the stretched case, and the green pentagons show the compressed case. (b) The dissipation rates for the expanding insertions at the singular configuration, where the volume removal and the trigon insertion are energetically favorable for the stretched and compressed cases, respectively.

could be relevant depending on the grain boundary energy function.

Depending on the geometry of the surrounding boundaries, each insertion has a different energy dissipation rate associated with the subsequent evolution. The energy dissipation rate criterion states that the insertion with the highest positive dissipation rate is the one that will be realized. As a test of this criterion, a mesh was generated for the configuration in Fig. 3.13. If the geometry is such that the three S^1s on top and three S^1s on the bottom are separated by the tetrahedral angle, a degenerate configuration is created where any insertion results in an unstable configuration with increased energy. If the angles between the S^1s are instead larger than the tetrahedral angle, a trigon insertion should be favored. Conversely, if the angles between the S^1s are smaller than the tetrahedral angle, a volume removal should be favored.

The results of this simple test are shown in Fig. 3.14. The energy changes in Fig. 3.14a are calculated with the new vertices on the outer projection sphere. For the compressed case where trigon insertion is favored, it is significant that the digon insertion is also energy decreasing and the petal removal type-I is nearly energy neutral. The dissipation rates associated with the expanding insertions are shown in Fig. 3.14b, and correctly identify

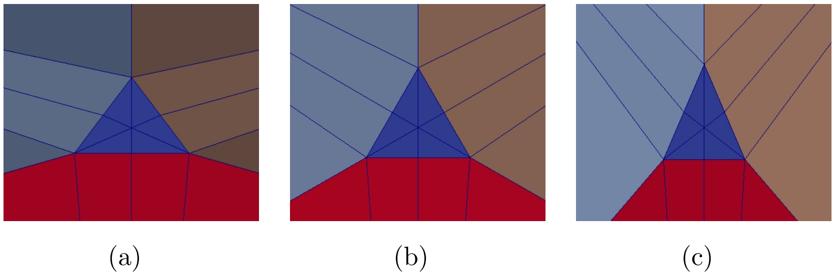


Figure 3.15. The effect of orthogonal stretching on the trigon shape. (b) Starting configuration, where dihedral angles between surfaces separating the surrounding S^3 are equal. (a)-(c) After stretching (compressing) the configuration in the lateral direction, running the relaxation yields a laterally stretched (compressed) S^2.

the most energetically favorable insertions.

The current scheme applies smaller forces on the inserted triangles than the surrounding triangles due to the discretized equations of motion, and a small bias towards trigon insertions in the degenerate configuration is visible in Fig. 3.14. The bias depends on the selection of the ratio of the radii of the inner and outer spheres in Fig. 3.12. By increasing the ratio, smaller radius insertions are discarded, effectively creating a range of grain configurations around the degenerate case where no insertion is valid. However, that can also make high aspect ratio S^2 insertions hit the inner sphere and be discarded until their aspect ratio lowers on the consecutive time steps.

Whereas the vertical stretch changes which insertion is energetically favored, lateral stretches change the energy-minimizing shape of the inserted stratum as is reflected by the relaxation scheme. Without this, insertions of equilateral S^2 could increase the energy artificially and cause a physically favorable insertion to be overlooked. Relaxation mitigates the problem, and as shown in Fig. 3.15, the shape of the inserted S^2 changes along with the surrounding geometry.

Finally, we simulate the evolution of an artificial microstructure of 100 grains generated as a Voronoi tesselation using Neper [40]. The simulation cell is a cube with unit edge length but is not periodic, requiring that a local volume preservation constraint be imposed on the exterior vertices. This relaxes the connectivity constraint on grain surfaces on the exterior, and requires some additional operations described in Sec. 7.2.8 of the appendix.

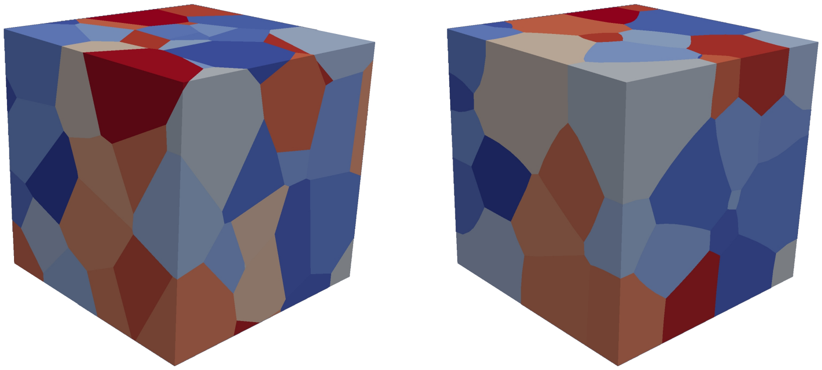

(a) (b)

Figure 3.16. Simulation of a microstructure composed of 100 grains under isotropic grain boundary energy. (a) Initial configuration. (b) The number of grains is about one half of the starting number.

The mesh is adaptively refined, with a target edge length set to a fraction of the median of the cube-equivalent grain diameters. The S^1 are additionaly required to contain at least two edges to provide sufficient degrees of freedom. The microstructure is evolved using equations of motion by Mason [44] with unit surface drag coefficient and grain boundary energy. The time iteration is implemented by a second order Runge-Kutta scheme with the time step at each iteration given by $\min(t_{\text{inv}}/20, t_{\text{fixed}})$, where t_{inv} is the shortest time step to invert any element and t_{fixed} is the maximum fixed time step of 5.0×10^{-5}. One iteration loop involves nine sub-iterations of the equations of motion, checking for and implementing collapses, followed by checking for and implementing insertions. Some snapshots from the resulting system evolution are shown in Fig. 3.16, with the discretization of several grains visible in Fig. 7.8 of the appendix. For reference, this simulation required about a week on a PC with 16 GB of 2400 MHz DDR4 RAM and an Intel(R) Core(TM) i7-7700HQ CPU at 2.80GHz. More details of the computational resource use and scaling are provided in Sec. 7.2.12 of the appendix.

The modified MS relation in Sec. 3.4 is used to calculate the rate of volume change of grains composed of discretized linear elements. The resulting actual rates of volume change for a select number of grains and the predictions of the modified MS relation are given in Fig. 3.17. The initial discrepancy is mainly due to the deviation from the

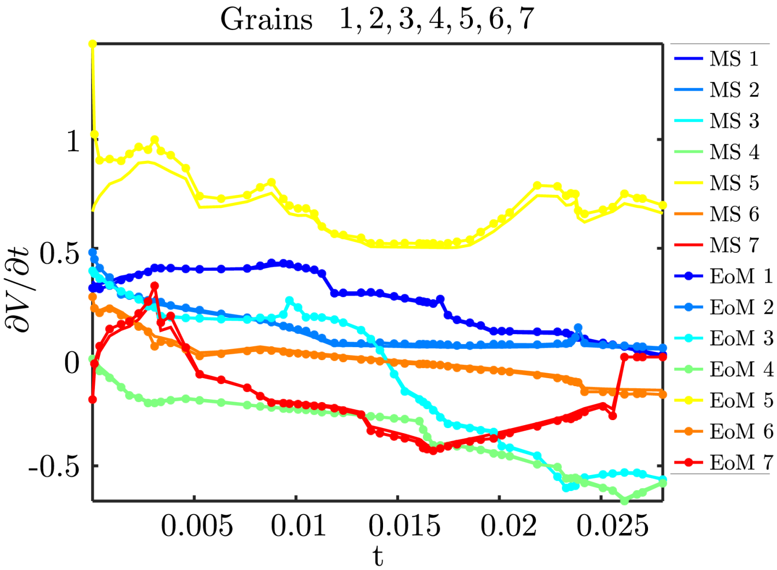


Figure 3.17. The rates of volume change for example grains as calculated by the modified MacPherson–Srolovitz (MS) relation, and first-order approximation using the equations of motion (EoM).

equilibrium angle conditions in the initial condition. The discrepancy decreases as the initial microstructure evolves and the angles around the junction lines approach the equilibrium values. Topological transitions can also cause temporary deviations (e.g., grain 5 around $t = 0.003$ in Fig. 3.17) which decrease with time. Despite using linear elements and an explicit time integration scheme, there is overall good agreement with the MS relation.

3.6 Conclusion

A computational framework with an explicit grain boundary representation is proposed to predict grain growth for anisotropic grain boundary energies and mobilities. This establishes the foundations of a massively parallelizable general-purpose framework to model microstructure evolution during, e.g., high-temperature and finite-strain processes. There does not appear to be any other software with these capabilities, that uses an explicit boundary representation, and that supports general changes to the grain boundary network.

Predictive simulations of microstructure evolution during thermomechanical processing require the ability to represent features such as stable quadruple junction lines in low stacking-fault energy metals. This in turn requires the ability to handle anisotropic properties and more general topologies than usually assumed in the literature. Moreover, the mesh should be partitioned across multiple processing units to reach physically relevant scales, and the equations of motion should be local to keep the computational cost linearly proportional to the number of grains. The discrete equations of motion proposed by Mason [44] can accommodate anisotropic grain boundary energies and drag coefficients. They are local and scalable, and have been implemented to describe the boundary motion.

A generic method to identify the possible singular transitions is proposed and implemented. An insertion selection criterion based on the energy dissipation rate is proposed and implemented. The method can utilize models for anisotropic energies, and once experimental grain boundary energy functions are available, the framework will be used to simulate grain growth under these conditions. Finally, the work is done in the context of a massively parallelizable finite element based library that can support volumetric physics.

Chapter 4

Benchmarks[1]

Quantifying accuracy is essential to any simulation efforts. This chapter introduces a set of three test cases to evaluate the relative accuracy and numerical cost of simulations of grain boundary motion, and uses this set to compare the discrete interface and multiphase field methods. The three cases correspond to several of the simplest configurations involving the motion of a grain boundary surface, a triple junction (TJ), and a quadruple point (QP). The grain boundary properties are assumed to be isotropic; a coarsening grain structure with isotropic grain boundary properties is said to be the ideal grain growth system, and provides a basis for comparison of all other evolving grain structures. Analytical forms for the evolving geometries are known for the spherical surfaces and TJ cases [50, 52], and the TJ and QP configurations have well-defined steady-state geometries. It is also of interest whether the two methods converge to the same geometries in situations for which analytical solutions are not known, since there is likely no other way to verify the simulations in such cases. While several of these configurations have been studied before, they are not usually considered in conjunction despite the benefits of doing so. Namely, the increasing complexity of the grain boundary configurations among the three test cases introduces different sources of systematic error to the grain boundary motion, and these errors can be more easily identified by comparing the test cases to one another.

It is desirable to establish the natures of any systematic errors and the accuracy of the simulation methods for a system with isotropic grain boundary properties before

---

[1] The content of this chapter is under review and is available in Ref. [51].

attempting to do so with more general grain boundary energy and mobility functions. The two methods considered in the present work will be capable of simulating the motion of grain boundaries with anisotropic properties when such functions become available. The discrete boundary method uses equations of motion that allow for general grain boundary properties and grain boundary lines that join an arbitrary number of grains [44]. The multiphase field model was developed to simulate the faceting of grain boundaries with energies that depend on boundary plane orientation, though this requires calculating a fourth-order derivative of the order parameters [26].

This chapter is structured as follows. We begin with a discussion of the discrete interface method and its implementation, followed by an analogous discussion for the multiphase field/diffuse interface counterpart. We then apply both methods to a set of three test cases: a two grain system (shrinking sphere), a three grain system (triple junction), and a four grain system (quadruple point). The behavior of the discrete and diffuse models is compared for each of the examples *vis-à-vis* analytic predictions and the models' internal length scales. The performance of the discrete interface case is briefly discussed, and then we conclude with a general discussion of the behaviors of the two models and a summary of recommendations for best practice in discrete interface modeling.

4.1 Methods

Assuming that grain boundary properties are independent of grain boundary crystallography implies that the grain boundary network evolves along the negative gradient of the total boundary area. This is usually expressed as the Turnbull equation [53]

$$\boldsymbol{v} = \mu\gamma K\hat{n}, \tag{4.1}$$

governing the motion of each boundary patch where $\boldsymbol{v}$ is the velocity, μ and γ are the mobility and energy per unit area, K is the mean curvature (the sum of the principle curvatures), and $\hat{n}$ is the unit normal vector.

While this is sufficient to determine the time evolution of a closed surface, the Turnbull equation does not specify what happens at the TJs or QPs of the grain boundary network. One of the essential differences between discrete and diffuse interface models is the governing equations for precisely these locations. Discrete interface models generally represent the TJs and QPs as distinct entities with explicit geometries, and sometimes provide additional governing equations specific to these locations [36]. This is in contrast to the implicit approach of most diffuse interface methods which do not track TJs or QPs explicitly (while some diffuse interface methods do include higher order terms to account for the distinct behavior of line or point defects, these can come at extreme computational cost). In diffuse interface methods, each surface evolves according to the Turnbull equation with the geometric singularities at the TJs and QPs regularized by the diffuse interfaces. This difference in the handling of TJs is significant since the TJs define the geometric conditions at a grain boundary's edges and thereby constrain the evolution of the grain boundary surface. Therefore, the microstructure trajectory is likely sensitive to the geometry around and motion of the TJs. It is for this reason that the angles between adjoining grain boundary surfaces are often used as simple scalar measures of the simulation accuracy in Section 4.2 below.

4.1.1 Discrete interface model

As implied by the name, every discrete interface model uses a discrete representation of the grain boundary network. A discrete representation entails that the grain boundary network geometry is represented by a collection of simple geometric objects, or elements, along with a description of how to join those elements together. The result is known as a surface mesh in three dimensions, and can be advantageously extended to a volumetric mesh to provide a discrete representation of the grain interiors as well. VDlib [1, 45] is a C++ library based on SCOREC [54] that represents a grain structure by means of a volumetric mesh containing tetrahedra, triangles, edges, and vertices.

There are two operations involved in updating the mesh to evolve the microstructure. The first moves the vertices of the mesh according to established equations of motion [44] that allow for anisotropic surface energies and arbitrary drag coefficients (the counterpart to the usual grain boundary mobility). The idea is that the velocity $\boldsymbol{v}$ of any given vertex should be such that the driving force $\boldsymbol{F}$ on the vertex is precisely balanced by the sum of drag forces $\boldsymbol{Dv}$ resulting from the motion of the adjoining grain boundary elements, where

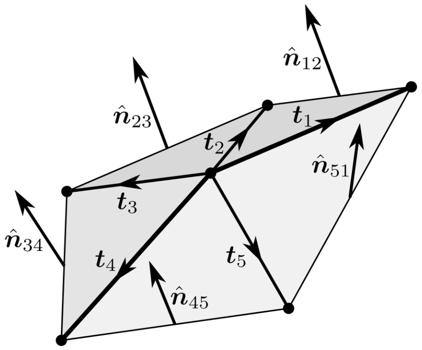


Figure 4.1. Vectors describing the geometry around a vertex of the surface mesh. The central vertex is connected to five edges $\boldsymbol{t}_i$ and five triangles with unit normal vectors $\hat{\boldsymbol{n}}_{ij}$. The TJ along the edges $\boldsymbol{t}_1$ and $\boldsymbol{t}_4$ is shown in bold.

$\boldsymbol{D}$ is the drag tensor and $\boldsymbol{v}$ is the grain boundary velocity. The capillary force acting on the vertex is given by

$$\boldsymbol{F} = \sum_i \hat{\boldsymbol{t}}_i \tau_l(\hat{\boldsymbol{t}}_i) + \frac{1}{2}||\boldsymbol{t}_i|| \sum_{j:\{i,j\}\in\Delta} (\hat{\boldsymbol{n}}_{ij} \times \hat{\boldsymbol{t}}_i)\gamma(\hat{\boldsymbol{n}}_{ij}) + \hat{\boldsymbol{n}}_{ij} \left.\frac{\partial\gamma}{\partial\phi_i}\right|_{\hat{\boldsymbol{n}}_{ij}}, \tag{4.2}$$

where τ_l and γ are the line and surface energy functions, $\boldsymbol{t}_i$ the vector along edge i starting at the vertex and $\hat{\boldsymbol{t}}_i$ is the corresponding unit vector, $\hat{\boldsymbol{n}}_{ij}$ is the normal of the triangle formed by edges i and j, $j : \{i, j\} \in \Delta$ indicates an edge j starting at the vertex such that edges i and j span a triangle Δ, and ϕ defines the surface orientation around edge i; Fig. 4.1 shows several of these quantities for a generic vertex of a surface mesh. At force equilibrium the capillary forces are balanced by the drag forces $\boldsymbol{Dv}$ applied by the moving boundaries with

$$\boldsymbol{D} = \delta_0 \boldsymbol{I} + \frac{1}{2}\sum_i \delta_1(\hat{\boldsymbol{t}}_i)||\boldsymbol{t}_i||(\boldsymbol{I} - \hat{\boldsymbol{t}}_i \otimes \hat{\boldsymbol{t}}_i) + \frac{1}{6}\sum_{i,j\in\Delta} \delta_2(\hat{\boldsymbol{n}}_{ij})||\boldsymbol{t}_i \times \boldsymbol{t}_j||(\hat{\boldsymbol{n}}_{ij} \otimes \hat{\boldsymbol{n}}_{ij}) \tag{4.3}$$

and where δ_k is the drag term associated with the k-dimensional simplicial boundary element. The resulting boundary vertex velocity $\boldsymbol{v}$ is given by

$$\boldsymbol{v} = \boldsymbol{D}^{-1}\boldsymbol{F}. \tag{4.4}$$

One advantage of this formulation is that the motion of every boundary vertex is governed by the same equation, including those on the interiors of surfaces, along TJs, and at QPs. If the point and line drag terms are zero, $\boldsymbol{Dv}$ reduces to the sum of the drag forces exerted by the neighboring triangles along the triangle normal directions for a given velocity $\boldsymbol{v}$. Moreover, if the grain boundary properties are constant, then $\delta_2 = 3/\mu$ and this further reduces to a discrete version of Eq. (4.1) with an accuracy that depends on the product of the edge length and the mean curvature of the surface. The explicit nondimensionalized forms are given in 7.3.1.

Apart from the motion of the mesh vertices, the accuracy of the discrete interface model is highly dependent on the element quality, where low-quality elements do not resemble equilateral triangles or tetrahedra [55, 56]. Without regular intervention and adaptation of the mesh, the quality of mesh elements generically degrades with grain boundary motion, even to the point of elements inverting. The discrete interface method handles this by using MeshAdapt [57] to locally remesh where the element quality falls below a threshold value, and coarsening or refining edges with lengths below or above a threshold value. The target edge length ℓ_e is constant in time and space for any given simulation, and an edge is coarsened or refined if the edge length l is outside the interval $0.7\ell_e \leq l \leq 1.5\ell_e$. These operations are used sparingly though, since apart from the computational expense local remeshing can perturb the grain boundary geometry. Specifically, these operations are the source of the discontinuous jumps observed in the discrete interface model results in Section 4.2.

4.1.2 Diffuse interface model

Comparison to a standardized diffuse interface model provides verification of the discrete interface model, as well as a benchmark for performance and code efficiency. In this work we apply the multiphase field model implemented following the presentation in Refs. [58, 59], which are general references for this section. A brief overview is provided here. For a system in region $\Omega \subset \mathbb{R}^3$ with N grains, N order parameters (denoted as the vector of functions $\boldsymbol{\eta} = \{\eta_1, \ldots, \eta_N\} \subset C_2(\Omega)$) are defined such that the region occupied by grain i is precisely the support of η_i. The free energy of the system is then defined to be

$$W[\boldsymbol{\eta}] = \int_\Omega \left(w(\boldsymbol{\eta}) + \frac{1}{2} \sum_n k \left|\nabla \eta_n\right|^2 \right) d\boldsymbol{x}, \tag{4.5}$$

where w is the chemical potential and k is a model parameter to be discussed subsequently (the use of functional brackets should be understood to indicate dependence on the argument and any temporal or spatial derivatives). The following polynomial form is used for the chemical potential:

$$w(\boldsymbol{\eta}) = m\sum_n \left(\frac{1}{4}\eta_n^4 - \frac{1}{2}\eta_n^2 + \frac{3}{4}\sum_{m>n}\eta_m^2\eta_n^2\right), \quad m = 3.26. \tag{4.6}$$

The coefficient for the boundary term is related to the grain boundary energy σ by

$$k = \frac{3\ell_{GB}}{4}\sigma, \tag{4.7}$$

where ℓ_{GB} is the diffuse boundary width. The evolution of $\boldsymbol{\eta}$, which determines the overall evolution of the microstructure, follows an L^2 gradient descent to minimize Eq. (4.5). The resulting kinetic evolution equation, expressed in terms of the variational derivative, is

$$\frac{\partial\eta_n}{\partial t} = -L\frac{\delta W}{\delta\eta_n}, \tag{4.8}$$

where the rate coefficient L is related to the traditional boundary mobility μ by

$$L = \frac{4}{3}\frac{m}{\ell_{GB}}. \tag{4.9}$$

All diffuse boundary calculations are performed using Alamo, a high performance multiphysics code that uses block-structured adaptive mesh refinement (BSAMR) with a strong-form elasticity solver to perform diffuse interface calculations [60]. Alamo is built on the AMReX package, developed by Lawrence Berkeley National Laboratory [61]. All of the results presented here were run on a desktop computer and generally completed in less than an hour (depending on the chosen parameters). Of particular interest is the convergence of the solution with respect to the boundary width, ℓ_{GB}, which determines the diffuse boundary length scale. The exact solution is recovered as $\ell_{GB} \to 0$, but this comes at the expense of increased computational cost. In this work we are particularly interested in the relationship between ℓ_{GB} and the discrete interface model counterpart.

4.2 Results and discussion

Three cases are considered in this section to quantify the systematic error of the discrete interface and multiphase field methods of simulating grain boundary motion. The first is a spherical grain which evolves in a self-similar way. This is a standard configuration that is often used in the literature to verify that the Turnbull equation is obeyed in the absence of complicating factors [33, 62–64]. The second is a TJ that migrates along a semi-infinite grain boundary [25, 63, 65], eventually reaching a steady state configuration with a known profile and velocity [50]. The quantity considered below is the angle of the grain boundaries at the TJ, though in principle a stricter validation scheme could involve evaluating the simulation's ability to precisely reproduce the expected geometric evolution. The third is a columnar hexagonal grain configuration that migrates along semi-infinite grain boundaries to allow a study of the steady state evolution of a QP [63, 66]. Perhaps the reason this case appears less often in the literature is that an analytical solution for the boundary profile is not known; instead, the angles between grain boundary traces on two cross-sections are evaluated for convergence and used to compare the two simulation methods. The grain boundary geometries for the three test cases are described in their respective sections, have Neumann boundary conditions, and are constructed to make the grain boundary curvatures comparable.

It is expected that the accuracy of both the discrete and diffuse interface models will increase with decreasing internal length scale ℓ, denoted as $\ell = \ell_e$ for the discrete model and $\ell = \ell_{GB}$ for the diffuse. However, the accuracy should not depend on any absolute length scale, since then the accuracy could be improved simply by uniformly scaling the grain structure. The accuracy therefore depends on ℓ relative to a second length scale that is characteristic of the evolving interface. Since the accuracy should be invariant to the isometries of Euclidean space, the inverse of the interface's mean curvature is the natural candidate for the second length scale, and the accuracy of both models is expected to depend on the dimensionless product of ℓ and interface's mean curvature. More precisely, all of the errors reported in this section are expected to be power laws in ℓ, with the prefactor depending on the mean curvature and implementation details in a way that is difficult to parameterize (only the spherical grain has the same mean curvature everywhere). For this reason, only the exponent of ℓ is generally reported in the following.

Many of the quantities reported below are nondimensionalized following the procedure

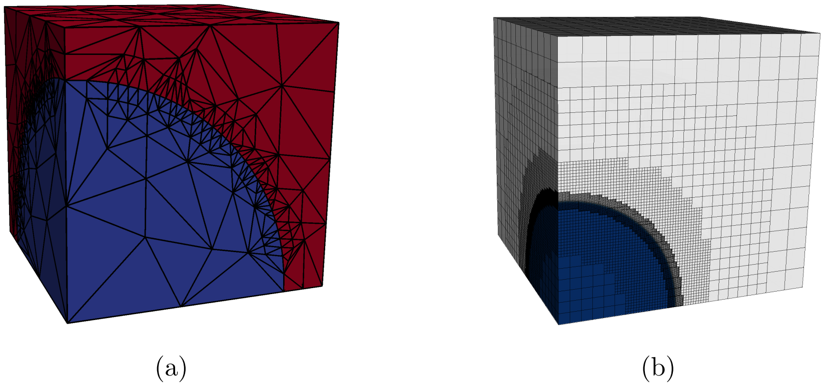

(a)
(b)

Figure 4.2. Initial geometries of the shrinking spherical grain within another grain for the (a) discrete and (b) diffuse interface methods.

in 7.3.1 to facilitate the comparison of the discrete interface and multiphase field methods. A tilde indicates a nondimensionalized variable (with the exception of ℓ_e and ℓ_{GB} which are always nondimensionalized) and an analytical prediction is denoted by the subscript t, e.g., $\tilde{r}_t(\tilde{t})$ is the analytical prediction for the nondimensionalized radius of the sphere as a function of nondimensionalized time. The equations of motion of the discrete interface method were integrated using a second order Runge–Kutta scheme with a maximum time step of 1.25×10^{-5}.

4.2.1 Spherical grain

The spherical grain case is intended to reveal the error when modeling surface motion in the absence of confounding effects from other grain boundary network components. One advantage of this particular choice is that, provided the grain boundary properties are constant and isotropic, the evolution of a spherical grain is known analytically. As derived in 7.3.2, the sphere shrinks uniformly with radius

$$r_t(t) = \sqrt{r_0^2 - 4\mu\gamma t} \tag{4.10}$$

as a function of time. Nondimensionalizing this equation reveals that a sphere starting with a radius of $\tilde{r}_t(\tilde{t}) = 1$ vanishes at $\tilde{t} = 0.25$.

The actual simulations deviate from Eq. (4.10) both because the initial geometries shown in Fig. 4.2 are not precisely spheres and because the Turnbull equation in Eq. (4.1) is not precisely followed, though these sources of error are reduced as the ℓ are made smaller.

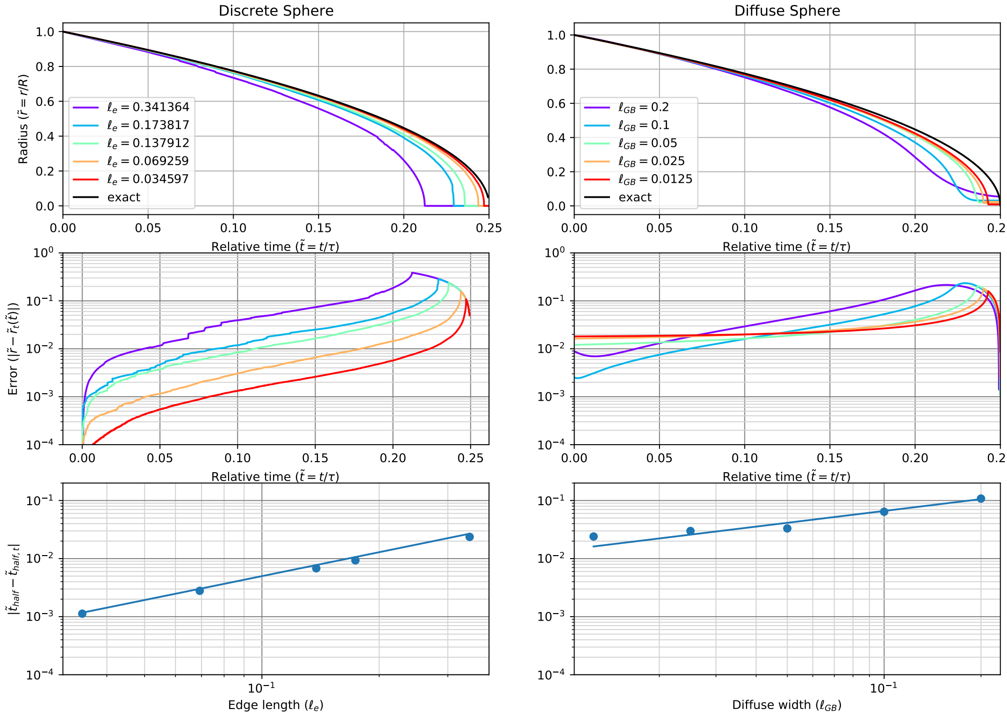


Figure 4.3. Comparison of shrinking spherical grain results for the discrete model (left) and the diffuse model (right); all quantities are nondimensionalized. (Top row) Plot of radius vs time, with color indicating the length scale and the exact solution in black. (Middle row) Plot of relative error in the radius vs time, with color indicating the length scale. (Bottom row) Plot of half-life error magnitude as a function of length scale.

Since the diffuse interface model does not perform well when the radius of the sphere approaches the grain boundary width, the magnitude of the error for the shrinking grain is quantified by the deviation of the sphere half-life t_{half} from the analytical prediction

$$t_{half,t} = \frac{3r_0^2}{16\mu\gamma}. \tag{4.11}$$

When nondimensionalized, the right hand side reduces to $\tilde{t}_{half,t} = 3/16$.

Figure 4.3 shows the performance of the two models, with the discrete interface model on the left and the diffuse interface model on the right. The top row shows the radius of the sphere as a function of time, where the color indicates the internal length scale and the exact solution is in black. The roughness of the curves for the discrete interface model is due to remeshing to preserve the element quality, and the velocity in the diffuse interface model falls as the radius approaches the grain boundary width. The magnitude of the relative error in the radius as a function of time is shown in the middle row. The error for the discrete interface model is caused by the magnitudes of the surface vertex velocities being larger than predicted by the analytical solution, perhaps as a consequence of the equations of motion being explicit and uncoupled. That the accumulation of error accelerates with decreasing radius supports the hypothesis that the error generally depends on the product of ℓ_e and the mean curvature. Meanwhile, there are likely two sources of error that contribute to the results for the diffuse interface model. The error at early times is a postprocessing artifact that occurs when constructing isocontours to identify the location of the grain boundary, effectively resulting in an offset to the sphere radius. The other source of error relates to the order parameter gradient at a grain boundary patch being affected by the presence of nearby patches. This is most visible when the grain is about to collapse and grain boundary patches on opposite sides of the grain interact, reducing the gradient magnitude and the grain boundary velocity. Conversely, the mean curvature of the surface causes neighboring grain boundary patches to interact, increasing the gradient magnitude and the grain boundary velocity at earlier times. As with the discrete interface model, the magnitude of this effect at earlier times is proportional to the product of ℓ_{GB} and the mean curvature.

The bottom row of Fig. 4.3 shows the half-life error $|\tilde{t}_{half} - \tilde{t}_{half,t}|$ as a function of the internal length scale. A conjugate gradient minimization algorithm and bootstrapping were used to fit $|\tilde{t}_{half} - \tilde{t}_{half,t}|$ to a power law in the internal length scale ℓ. This gives an exponent of 1.37 ± 0.21 for the discrete interface model and 0.678 ± 0.085 for the diffuse interface model, where the values are the medians and the uncertainties are half the interquartile range. While the exponents could suggest that the error of the diffuse interface model decays slower than that of the discrete interface model with decreasing internal length scale, the errors in the apparent grain radius due to isocontour construction during postprocessing do not actually affect the microstructure trajectory. This could motivate using the two-grain configuration with self-similar evolution analyzed by Mullins

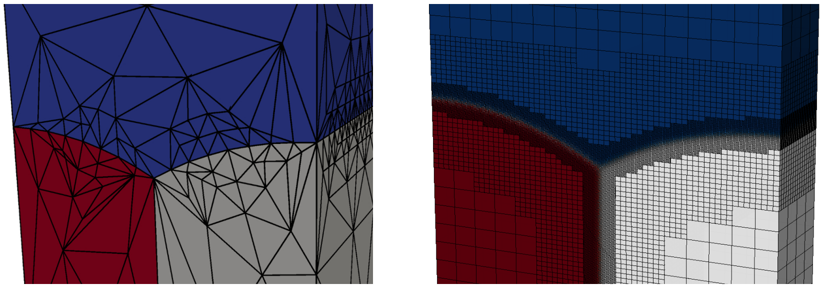

(a) (b)

Figure 4.4. Initial geometries of the TJs for the (a) discrete and (b) diffuse interface methods. The structures have mirror boundary conditions in the lateral directions.

[50] in the future since such postprocessing errors would likely not affect the long-time behavior.

4.2.2 Triple junction

The purpose of the TJ case is to include a TJ in the moving boundary while keeping the grain configuration as simple as possible, ideally allowing the error of the equations of motion for the TJ to be identified by comparing the results to those for the spherical grain. The initial geometries of the grain configuration are shown in Fig. 4.4, are constant in the out-of-plane direction, and have mirror boundary conditions in the lateral directions. The rate of volume change of the top grain can be derived by applying the von Neumann-Mullins equation [49, 50] to the two-dimensional grain configuration in a plane perpendicular to the TJ. Since there is one triple point per simulation cell in this plane, the rate of cross-sectional area change of the top grain per simulation cell width L is $\pi m\gamma/3$, and the rate of volume change of the top grain can be found by multiplying by the TJ length. Mullins actually went further and solved for the steady-state profile of the moving boundary assuming constant and isotropic grain boundary properties [50]. If x is distance from the left edge of the simulation cell and y is height from the top of the red grain, then the steady-state profile of the grain boundary between the red and blue grains is

$$y(x) = -\ln[\cos(\pi x)]/\pi. \tag{4.12}$$

The width of the simulation cell is defined by the above equation as $L = 1/3$ which can be scaled to match the dimensions of the simulation cell.

The dihedral angle θ_{TJ} between the two boundaries of the blue grain is perhaps the simplest way to evaluate the accuracy of the geometry of the moving boundary in the vicinity of the TJ. A force balance argument for constant and isotropic grain boundary properties (and the absence of any TJ drag) leads to the condition $\theta_{TJ,t} = 2\pi/3$. This is roughly enforced in the initial conditions by defining the two parts of the moving boundary to be the appropriate sections of of cylinders; while this is not the steady-state profile given by Mullins, it is sufficiently close for a short initial transient and rapid convergence to the steady-state condition as is visible in Fig. 4.5.

As before, results for the discrete interface model are on the left and those for the diffuse interface model are on the right. The top row shows θ_{TJ} as a function of time, where the color indicates the internal length scale and the exact solution $2\pi/3$ is in black. The roughness of the curves for the discrete interface model is due to the remeshing required to maintain element quality, and the periodic spikes that appear for the diffuse interface model are due to the interaction of the adaptive mesh refinement and the construction of the isocontours. The error in θ_{TJ} (measured as the median of the second half of the time series) is shown in the bottom row, with the dependence of the steady-state angle on ℓ_e for the discrete interface model being a consequence of the linear elements forcing the grain boundary curvature to be concentrated at the vertices and edges of the mesh. Specifically, the grain boundary curvature that is distributed to the TJ edges causes the deviation of θ_{TJ} from the expected value, with the magnitude of the deviation depending on the product of ℓ_e and the mean curvature of the adjoining grain boundary. Identifying the precise location of the TJ and the value of θ_{TJ} is more difficult for the diffuse interface model since the grain boundary geometry is implicit. The procedure followed here involves fitting third- and fourth-order polynomial approximations to each side of the isocontour where the order parameter for the top grain is 0.5. The triple point location in the plane is then defined to be the point of intersection of the polynomials, and θ_{TJ} is the angle between the tangent vectors at the point of intersection. This process works well in the sharp interface limit, but is very sensitive to perturbations in the solution for larger ℓ_{GB} since there is substantially more error in the predicted location of the TJ with respect

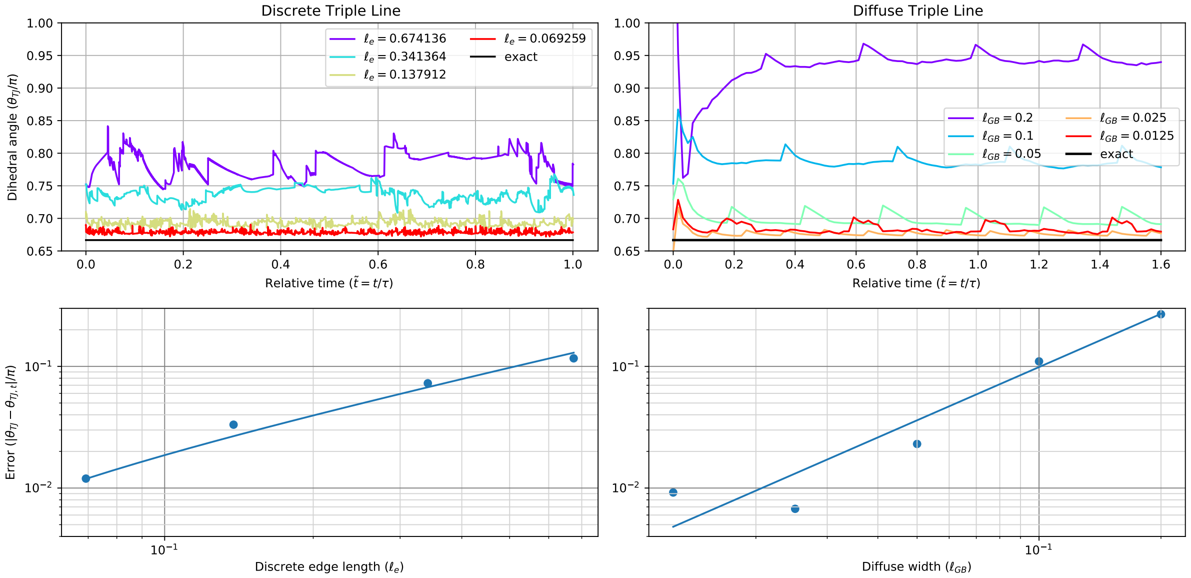


Figure 4.5. Comparison of θ_{TJ} for discrete model (left) and diffuse model (right); all quantities are nondimensionalized. (Top row) Plot of θ_{TJ} vs time, with color indicating the length scale and the exact solution in black. (Bottom row) Plot of the relative error vs length scale.

to the simulation size. The occasional deviations that are observed in the steady-state correspond to BSAMR re-gridding events.

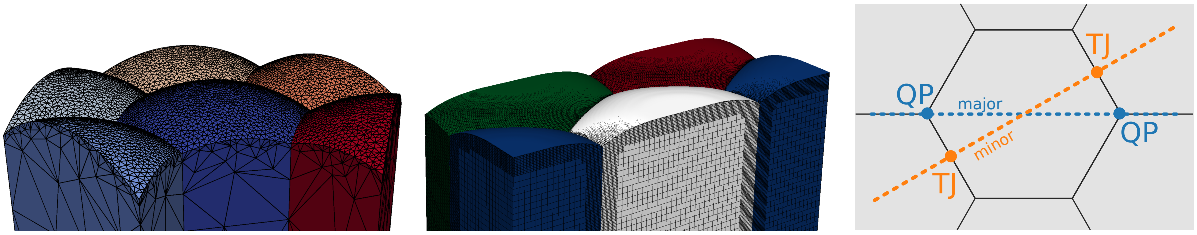
Fitting a power law in the internal length scale ℓ to $|\theta_{TJ} - \theta_{TJ,t}|/\pi$ gives an exponent of 0.91 ± 0.20 for the discrete interface model and 1.45 ± 0.13 for the diffuse interface model, where the values are the medians and the uncertainties are half the interquartile range. The additive offset of -0.005 ± 0.011 for the discrete interface model is entirely consistent with θ_{TJ} converging to the equilibrium angle in the $\ell_e \to 0$ limit, though at a lower rate than the half-life error magnitude in Fig. 4.3. This is not unexpected though, since the TJ can be thought of as a jump condition in the tangent plane to the grain boundary that is both difficult to accurately reproduce with a finite element mesh and is not present in the spherical grain case. While the exponent for the diffuse case is nominally higher, this is not reflective of the trend observed for small ℓ_{GB} where the saturation in the error is likely the result of inaccuracy in the postprocess calculation of the angle. The higher exponent therefore does not necessarily indicate better convergence.

4.2.3 Quadruple point

As with the TJ case, the grain structure for the QP case consists of a top grain above several columnar grains. The grain boundaries of the top grain migrate down the simulation cell, consuming the columnar grains and eventually reaching a steady-state profile, though an analytical solution for this profile is not known. The configurations of columnar grains for the discrete and diffuse interface models are shown in Figs. 4.6a and 4.6b respectively, with the hexagonal cross-sections of the columnar grains clearest for the discrete interface model; the BSAMR mesh makes simulations of rectilinear domains like the one in the figure strongly preferable for the diffuse interface model.

Following the initial transient, the steady-state profile is examined on the two planes indicated in Fig. 4.6c, one along a minor diameter of the central grain and bisecting a TJ, the other along a major diameter of the central hexagonal grain and containing a QP. The angles along these profiles at the intersections with the TJ and the QP are reported in Fig. 4.7 and Fig. 4.8. While the equilibrium angle at the TJ should be $2\pi/3$ (the same as for the TJ in Section 4.2.2), the curvature of the grain boundaries in both principal directions could change the rate of convergence to $2\pi/3$ with decreasing ℓ compared to the TJ case. As for the equilibrium angle at the QP, an infinitesimal neighborhood of the QP will contain triple junction lines in a tetrahedral configuration connected by flat grain boundary surfaces provided the principal curvatures of the grain boundaries are finite. This allows the equilibrium angle of $\cos^{-1}(-1/\sqrt{3}) \approx 0.696\pi$ at the QP along the major diameter to found by geometrical considerations.

Starting with the TJ angle, observe that the data points for the TJ angle error along the minor axis in the bottom row of Fig. 4.7 closely resemble those for the TJ angle error in the bottom row of Fig. 4.5. This indicates that the nonzero second principal curvature of the grain boundaries along the TJ lines in the QP case does not have a significant effect on the error in the equations of motion, and is consistent with the expectation that the error should scale with the mean curvature (the sum of the principal curvatures). Fitting a power law in the internal length scale ℓ to $|\theta_{TJ} - \theta_{TJ,t}|/\pi$ gives an exponent of 0.927 ± 0.223 for the discrete interface model and 0.85 ± 0.50 for the diffuse interface



(a) (b) (c)

Figure 4.6. QP mesh configurations and schematic. (a) Hexagonal columnar grain mesh for the discrete interface model. (b) Hexagonal columnar grain BSAMR mesh in a rectilinear domain for the diffuse interface model. (c) Locations of QP and TJ along major and minor lines.

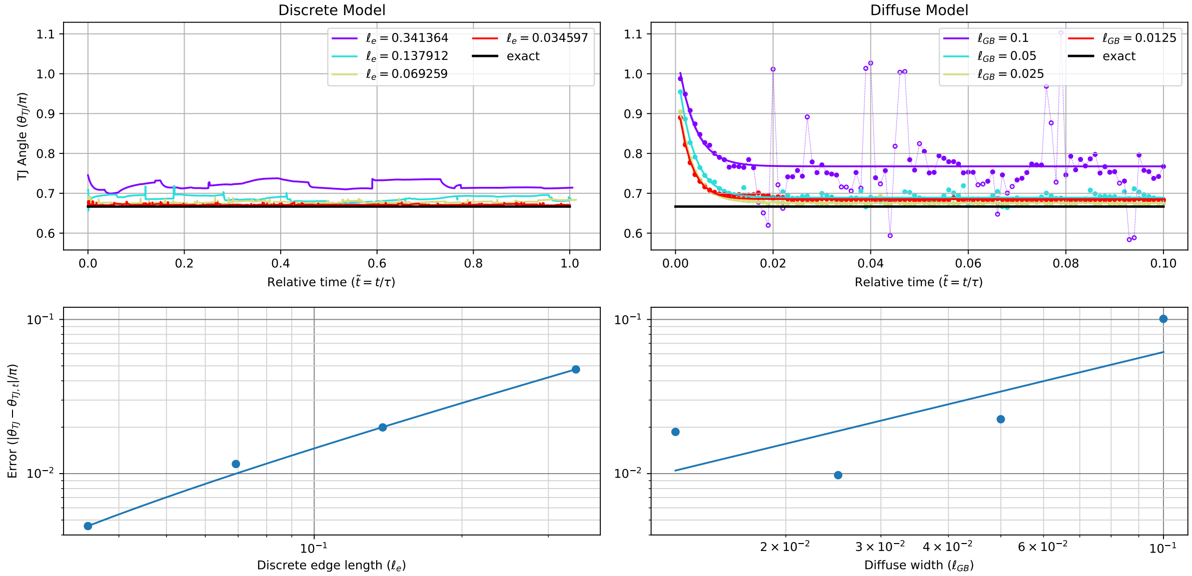


Figure 4.7. Comparison of minor axis results for the QP case for the discrete model (left) and diffuse model (right); all quantities are nondimensionalized. (Top row) Plot of the measured TJ (minor diameter) angle, with color indicating the length scale and the exact solution in black. (Bottom row) Plot of the relative error in the TJ angle with respect to length scale.

model, with both models converging to the expected value. While the exponent for the discrete interface model is nearly identical to that for the TJ case, the lower exponent for the diffuse interface model is likely a consequence of a power law fitting the data relatively poorly; observe that the TJ angle error for the diffuse interface model does not fall on a line on a log-log plot, and instead seems to saturate at a lower bound set by the angle estimation procedure in postprocessing.

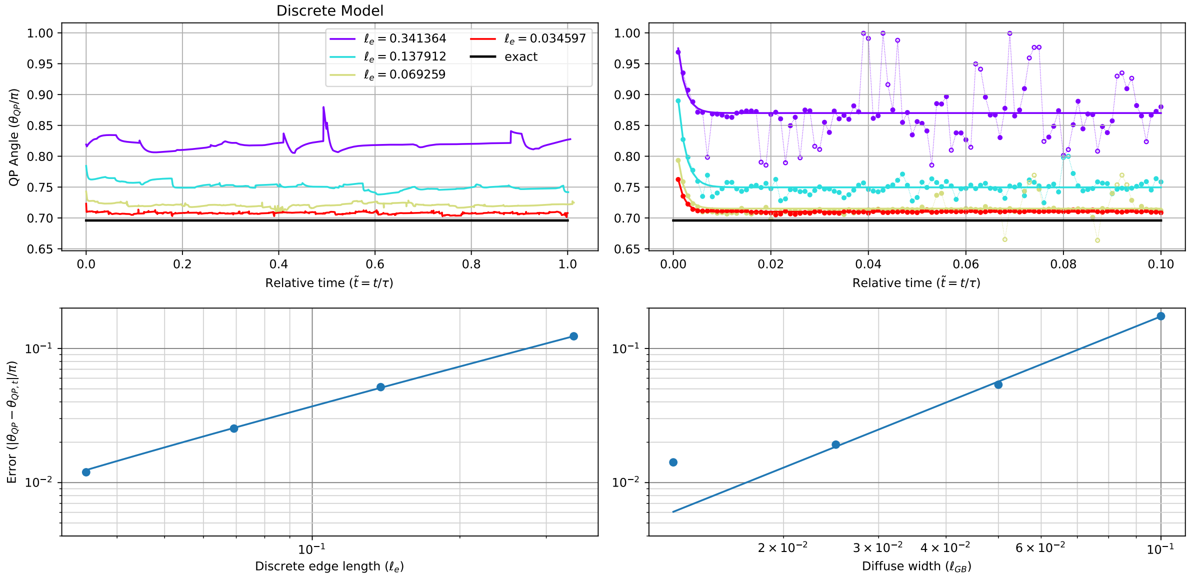


Figure 4.8. Comparison of major axis results for the QP case for the discrete model (left) and diffuse model (right); all quantities are nondimensionalized. (Top row) Plot of the measured QP (major diameter) angle, with color indicating the length scale and the exact solution in black. (Bottom row) Plot of the relative error in the QP angle with respect to length scale.

For the QP angle, the final values for the discrete interface model follow a power law in ℓ that converges to an angle of $(0.694 \pm 0.001)\pi$ with an exponent of 0.958 ± 0.026, whereas the respective values for the diffuse interface model are $(0.707 \pm 0.011)\pi$ and 0.85 ± 0.45; the limiting values for both the discrete and diffuse interface models effectively coincide with the exact value. It is significant that the errors for all of the discrete interface results in Secs. 4.2.2 and 4.2.3 decay with exponents that are close to one. The discrete interface method uses linear elements that approximate the grain boundary geometry with first-order accuracy, meaning that an exponent of one is the best possible result. It is likely that higher-order elements would need to be used to substantially increase the rate of error decay with ℓ_e. The irregularity in the exponents for the diffuse interface model in Secs. 4.2.2 and 4.2.3 is attributed to the error in the polynomial algorithm used to extract the grain boundary profile. Examination of Figs. 4.5 and 4.7 indicates that this functions as a source of random error that is larger for highly diffuse boundaries but vanishes in the sharp boundary limit.

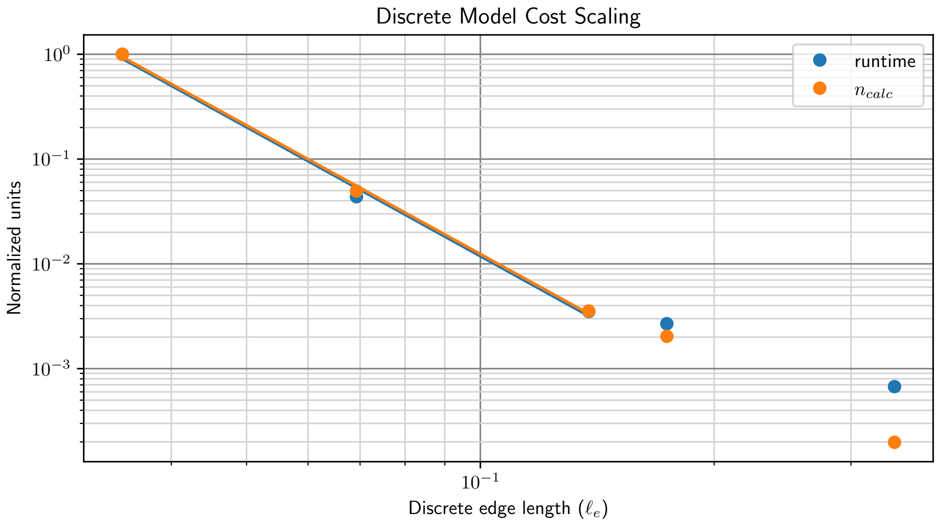


Figure 4.9. The scaling of the normalized runtime and the normalized number of grain boundary vertex calculations for the spherical grain case as a function of ℓ_e.

4.3 Performance

When selecting a numerical method in practice, computational cost is often nearly as much a concern as the accuracy of the simulated behavior. This section specifically considers the dependence of the discrete interface method's computational cost on the internal length scale ℓ_e. Suppose that the main contribution to the computational cost is evaluating the equations of motion for the grain boundary vertices. The number of such vertices is expected to depend on the internal length scale as ℓ_e^{-2}. If the velocity of the vertices is independent of ℓ_e, then the time step length should decrease as ℓ_e to keep the vertex displacement shorter than the characteristic edge length and prevent mesh element inversion. This would imply that the overall computational cost should scale with ℓ_e^{-3}, or as the product of the number of grain boundary mesh vertices and the number of time steps for a given overall simulation time.

Figure 4.9 shows the scaling of the normalized runtime cost and normalized number of grain boundary vertex calculations $n_{calc} = \sum_j n_{v_b,j}$, where $n_{v_b,j}$ is the number of grain boundary vertices at time step j, for the discrete interface method. These scale as $\ell_e^{-4.088}$ and $\ell_e^{-4.081}$, respectively, for small ℓ_e where the computational cost of the vertex calculations is expected to dominate. This confirms that the overhead of the discrete interface method (mesh management, enumeration of topological transitions, etc.) is relatively small compared to the evaluation of the equations of motion, but is not consistent with the ℓ_e^{-3} scaling expected in the previous paragraph. This discrepancy is a result of the length of the median time step scaling as $\ell_e^{1.916}$ instead of linearly; the underlying cause for this time step scaling is investigated further in 7.3.

4.4 Conclusion

The purpose of this work has been to establish the validity and performance of a recently-developed discrete interface method by comparison to analytic solutions and a well-established multiphase field method used to model grain boundary motion. More specifically, the evolution of the simplest configurations involving surfaces, triple lines, and quadruple points with self-similar behavior given constant and isotropic grain boundary properties are used to quantify the error in position and junction angles as a function of the degree of refinement. The boundary types are simple enough to be amenable to analysis, yet complex enough to introduce different systematic errors over the course of their evolution. Despite the approaches for simulating boundary motion being distinctly different, our results indicate that both of the discrete and diffuse interface methods converge to the same junction angles with similar rates. The most significant difference is that when predicting the half life of the shrinking sphere, the convergence rate of the diffuse interface method appears to be about half of that of the discrete interface method.

Although this work assumes constant and isotropic grain boundary properties, both methods were developed with the intention of performing simulations for anisotropic grain boundary properties. There are reasons to believe that the necessary data could be available in the future. Morawiec [6] suggested that the grain boundary energy could be experimentally obtained as a function of the grain boundary crystallography by applying the Herring condition [67, 68] to triple junctions imaged by three-dimensional microscopy techniques [3, 4]. Alternatively, molecular dynamics simulations allow direct evaluation of grain boundary properties in bicrystals [69]. While the excessive number of points required to adequately sample the five-dimensional grain boundary space of has precluded the availability of general grain boundary energy and mobility functions in the literature, there has been progress for particular subsets of grain boundaries [9, 10]. The absence of general grain boundary energy and mobility functions has practically limited existing simulations of grain boundary motion to the canonical system for which the grain boundary properties are assumed to be independent of grain boundary crystallography.

The performance of the discrete interface model lends confidence in its ability to yield accurate results for more general and complex microstructures for which there is no known analytic solution. Moreover, the performance with unoptimized code indicates reasonable scaling behavior that is close to the ideal scaling and comparable to that of alternative methods.

Acknowledgements

EE and JKM were supported by the National Science Foundation under Grant No. DMR 1839370. BR was supported by the Office of Naval Research, grant #N00014-21-1-2113. This work used the INCLINE cluster at the University of Colorado Colorado Springs. INCLINE is supported by the National Science Foundation, grant #2017917.

4.5 Data availability

The ALAMO (`https://github.com/solidsgroup/alamo`) and VDlib (`https://github.com/erdemeren/VDlib`) libraries used to generate these results are available as open source. The processed data required to reproduce these findings are available to download from [`https://arxiv.org/abs/2203.03167`].

Chapter 5

Constant of motion[1]

The grain structure of polycrystalline materials is deceptively simple, and for that reason has been the subject of intense and ongoing study. For specificity, consider a model system where the grain boundary energy and mobility are constants, i.e., do not depend on grain misorientation or the boundary plane normal. The phenomenological Turnbull equation [71] relates the normal velocity of a grain boundary in such a system to the driving pressure, and along with the Young–Laplace equation [72] suggests that the normal velocity is directly proportional to the mean curvature of the grain boundary. The migration of individual boundaries induces the evolution of the grain structure, a process known as grain growth, where the total area of grain boundaries and the number of grains decrease with time.

There are surprisingly few rigorous results known about grain structures, even for the two-dimensional version of this system. Energy considerations require that grain boundaries only meet at triple junctions with internal angles of $2\pi/3$ [73, 74]. A consequence of this and curvature-driven grain growth is that a grain's area changes at a rate that depends only on the number of bounding vertices [49, 50]. Globally, topological arguments require that the average number of such bounding vertices be precisely six [75]. There are natural analogues to several, but not all, of these results in three dimensions. Grain boundaries only meet at triple junction lines with dihedral angles of $2\pi/3$, and triple junction lines only meet at quadruple junction points with angles of $\cos^{-1}(-1/3)$ [73, 74]. The rate of

---

[1] The content of this chapter has been previously published in Ref. [70].

volume change of a grain depends not only on the total length of the bounding triple lines, but on a measure of the linear dimension known as the mean width [37, 76]. For both the two- and three-dimensional systems, the hypothesis that the structure reaches a statistically self-similar state implies that that the average grain diameter increases as the square-root of time [77, 78]. This is the effective extent of current knowledge.

There have been a variety of inexact relationships proposed as well, usually for grain structures in the conjectured self-similar state [79]. Ones that relate to the global properties of the three-dimensional system include proposed distributions for the effective radius of a grain [80–82] and the number of faces bounding a grain [82]. Recent advances in several microscopy techniques promise to make three-dimensional grain structure data more readily available, possibly allowing such relationships to be further refined. Three-dimensional electron backscatter diffraction [83, 84] destructively images the grain structure by a serial sectioning process, whereas three-dimensional X-ray diffraction microscopy [4, 85] is non-destructive but generally offers poorer spatial resolution. Given this situation, additional rigorous results for the global properties of the grain structure of the model three-dimensional system would be valuable, both to measure deviations of the experimental systems from the model one, and to verify the accuracy of grain structures generated by computational means. This article proves one such result, relating the integral of the Gaussian curvature over the grain boundaries to the numbers of grains and quadruple junction points, and thereby to the numbers of grain boundaries and triple junction lines.

Let Ω be a space-filling grain structure composed of grains that meet in twos on grain boundaries, grain boundaries that meet in threes at triple junction lines, and triple junction lines that meet in fours at quadruple junction points, as in Fig. 5.1. Further suppose that Ω satisfies Plateau's laws (i.e., grain boundaries meet at dihedral angles of $2\pi/3$ and triple junction lines meet at angles of $\cos^{-1}(-1/3)$), and that Ω is defined in a three-dimensional region with periodic boundary conditions. If G is a grain in Ω, then our main result is that the expectation value of the Gaussian curvature K integrated over the interiors of the grain boundaries of G and the expectation value of the number of quadruple junction points $f_0(G)$ of G are related by:

$$\left\langle \int_{\partial G} K \, \mathrm{d}A \right\rangle = 4\pi - \alpha \langle f_0(G) \rangle. \tag{5.1}$$

The angle brackets indicate an average performed over all grains in Ω, ∂G indicates the interiors of the grain boundaries of G, and $\alpha = 2\pi - 3\cos^{-1}(-1/3)$ is the angular defect at a quadruple junction point. This result is exact (given a few technical assumptions that are usually satisfied and are discussed in Sec. 7.4 of the appendix, and to our knowledge does not appear in the literature; a related result by Kusner [86] requires that all the grain boundaries be minimal surfaces, and one by Glicksman [87] applies only to unconstructable grain structures of average n-polyhedra.

The Gaussian curvature of a surface is defined as the product of the principal curvatures at any point. The appearance of this quantity in Eq. 5.1 could be surprising, since the mean curvature (the sum of the principal curvatures) is the one that controls the dynamics of the grain boundary network [53, 72]. That said, the Gaussian curvature is in some ways the more fundamental of the two quantities, being an intrinsic property of the surface that does not depend on the way the surface is embedded in Euclidean space. For example, the

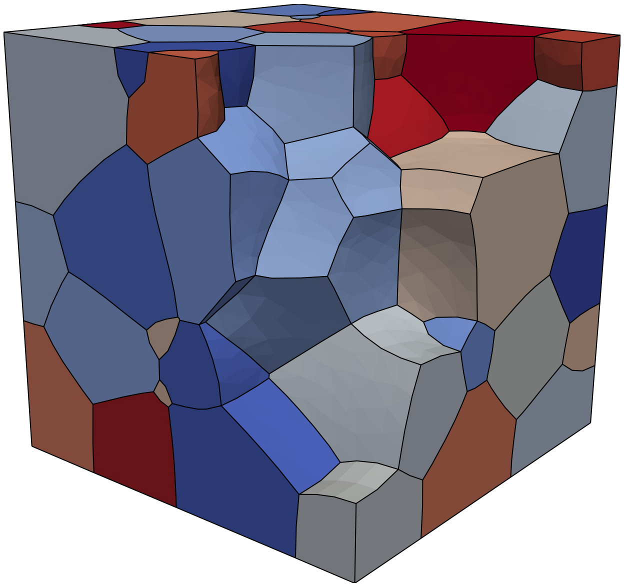


Figure 5.1. A grain structure in a cubic volume, with several grains removed to reveal the interior. Color indicates the individual grains, internal curved surfaces are grain boundaries, internal black lines are triple junction lines, and four triple junction lines intersect at quadruple junction points.

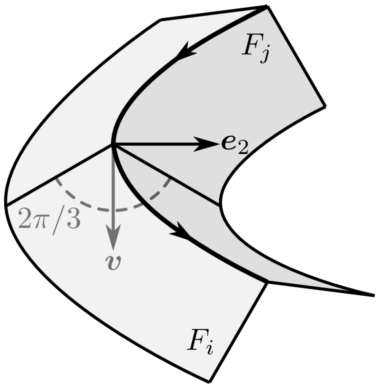


Figure 5.2. Grain boundaries F_i and F_j meet at the triple junction line in bold, and $\boldsymbol{v}$ bisects the dihedral angle between F_i and F_j. Arrows indicate the tangent direction, and the second Frenet vector $\boldsymbol{e}_2$ points along the triple junction line's normal direction.

Gaussian curvature of a sheet of paper is zero at every point whether the sheet is laid flat or rolled up, though the same is not true for the mean curvature. This invariance to the embedding is reflected in the celebrated Gauss–Bonnet theorem:

$$\int_{\partial G} K \,\mathrm{d}A + \sum_{i=1}^{f_2(G)} \int_{\partial F_i} \kappa_g \,\mathrm{d}s + \sum_{i=1}^{f_0(G)} \alpha_i = 2\pi\chi(\partial G).$$

While this version specifically applies to the surface of a grain, all versions relate the integrated Gaussian curvature of a surface to its Euler characteristic $\chi(\partial G)$ (equal to two when the surface can be smoothly deformed into a sphere without cutting or gluing). The terms on the left include the integrated Gaussian curvature over the grain boundary interiors, the sum of the integrated geodesic curvature κ_g over the interiors of the bounding triple junction lines ∂F_i of all grain boundaries F_i, and the sum of the angular defects α_i of the quadruple junction points of G.

If G can be smoothly deformed into a sphere and belongs to a grain structure Ω that obeys Plateau's rules, then this can be simplified to:

$$\int_{\partial G} K \,\mathrm{d}A + \sum_{i=1}^{f_1(G)} \int_{E_i} \kappa \boldsymbol{e}_2 \cdot \boldsymbol{v} \,\mathrm{d}s + \alpha f_0(G) = 4\pi$$

where the most significant change is to the middle term on the left; this is now the sum of the integrated curvature of the triple junction lines of G, weighted by the dot product of the second Frenet vector $\boldsymbol{e}_2$ of the curve and a unit vector $\boldsymbol{v}$ that bisects the dihedral angle between the adjoining grain boundaries; see Fig. 5.2. Summing this equation over all grains in Ω results in a remarkable cancellation (previously noted by DeHoff [88]) where the contribution of the second term on the left vanishes. Specifically, every triple junction line is integrated over three times, once for each adjoining grain. $\kappa \boldsymbol{e}_2$ is an inherent quantity of the triple junction line that is the same for all three integrals, but the three $\boldsymbol{v}$ are all unit vectors in a plane with mutual angles of $2\pi/3$. That is, the sum of the three $\boldsymbol{v}$ vanishes identically for each triple junction line, leaving an alternative version of the main result:

$$\sum_{i=1}^{f_2(\Omega)} \int_{F_i} K \, \mathrm{d}A = 2\pi f_3(\Omega) - 2\alpha f_0(\Omega) \tag{5.2}$$

where F_i is the ith grain boundary of Ω and $f_0(\Omega)$, $f_2(\Omega)$ and $f_3(\Omega)$ are the numbers of quadruple junction points, grain boundaries, and grains of Ω. Dividing through by $f_3(\Omega)$ and multiplying by a constant gives Eq. 5.1. More detailed derivations of both Eqs. 5.1 and 5.2 are provided in Sec. 7.4 of the appendix.

Although Eq. 5.1 appears to be simpler, there are at least two observations that are more clearly made by means of Eq. 5.2. The first is that the integral of K over the grain boundaries of Ω depends only on the numbers of grains and quadruple junction points of Ω, and not on the geometry of the grain structure. That is, the left-hand side of Eq. 5.2 is invariant to any deformation of Ω that preserves the numbers of grains and quadruple junction points. The second is that a sufficiently accurate measurement of the integral of K over the grain boundaries of Ω in principle specifies the numbers of all components of Ω. Observe that since there is no rational number that relates the coefficients of $f_3(\Omega)$ and $f_0(\Omega)$ in Eq. 5.2, the numbers of grains and quadruple junctions can be inferred if the left-hand side is known sufficiently accurately. The number of triple junction lines can then be found from $2f_1(\Omega) = 4f_0(\Omega)$ by a counting argument, and the number of grains from $0 = f_0(\Omega) - f_1(\Omega) + f_2(\Omega) - f_3(\Omega)$ which follows from the domain of Ω being a three-torus with $\chi(\Omega) = 0$. The necessary modifications to Eqs. 5.1 and 5.2 for grain structures in other domains (e.g., ones with free boundaries) are discussed in Sec. 7.4 of the appendix.

As numerical confirmation of Eq. 5.1, consider a grain structure consisting of periodic

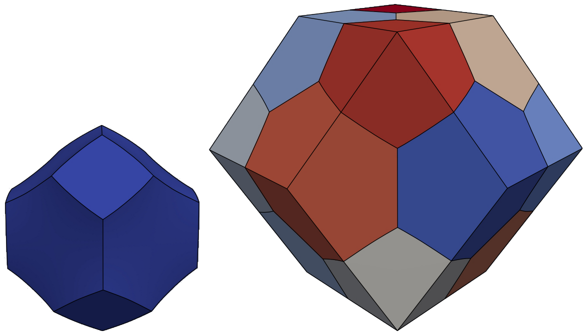


Figure 5.3. An infinite periodic grain structure that satisfies Plateau's laws can be constructed by repeating a relaxed truncated octahedron (left). This grain was found by starting with a periodic unit of a grain structure consisting of unrelaxed truncated octahedra, fixing the location of the interior quadruple junction points, and minimizing the grain boundary area (right).

truncated octahedra, relaxed under the action of surface tension to satisfy Plateau's laws; one such grain is shown on the left of Fig. 5.3. The shape of this grain was found by starting with a grain structure consisting of unrelaxed truncated octahedra and constructing the periodic unit shown on the right of Fig. 5.3, with a single grain at the center and corners at the centers of the neighboring grains. The periodic unit was computationally represented by a volumetric finite element mesh with linear elements, with the average number of triangles per hexagonal face n_t depending on a characteristic length and the details of the mesh adaptation algorithm. The locations of the interior quadruple junction points were fixed, and the structure was relaxed by allowing the vertices on grain surfaces to move according to equations of motion known to reproduce curvature-driven grain growth [44] until the magnitude of the vertex forces fell below a threshold. The grain structure did not reach a steady-state configuration when the locations of the interior quadruple junction points were not fixed, owing to a known instability of this grain structure to volumetric perturbations [89]. While quadratic elements would allow the steady-state geometry to be more accurately represented, a convergence analysis with an increasing number of linear elements is sufficient for the present purpose.

The simulations were performed with a modified version of a recently-developed microstructure evolution code [1] that usually uses SCOREC [54] for mesh management and maintenance, but the mesh adaptation operations were found to interfere with the

Table 5.1. The percent grain boundary area change of the relaxed truncated octahedron relative to the unrelaxed one and the integrated Gaussian curvature of the relaxed truncated octahedron as functions of n_t.

n_t	61	129	477	665	1885	2610	3250	4314	5322	6532	7649	8906
$\Delta A(\%)$	0.0863	0.129	0.135	0.136	0.136	0.137	0.136	0.136	0.136	0.136	0.136	0.136
$\int K \, \mathrm{d}A$	-0.219	-0.349	-0.461	-0.477	-0.544	-0.554	-0.563	-0.568	-0.578	-0.588	-0.593	-0.597

convergence of the grain geometry. Instead, artificial vertex forces defined by Kuprat [33] were used to maintain the mesh element quality during the structure relaxation. Since the artificial forces only acted on vertices on the grain interiors, it is expected that they did not substantially affect the grain geometry. The boundary conditions were defined to make the simulation cell behave as a periodic unit in a grain structure consisting of periodic truncated octahedra. Whereas the grain boundaries on the simulation cell interior had a constant nonzero energy per unit area, the external surfaces of the simulation cell were assigned zero energy per unit area; this is consistent with viewing them as the result of intersecting grains in the underlying grain structure with the boundary of the periodic unit. Vertices on the external surfaces were constrained to remain on the external surfaces during relaxation by projecting away any displacement in the normal direction, effectively imposing a Neumann boundary condition. The integrated Gaussian curvature was calculated as the sum of the angular defects at the vertices on the grain boundary interiors, where the angular defect is defined as 2π minus the sum of the interior angles of the grain boundary triangles meeting at the vertex.

Table 5.1 shows the results of this analysis for increasing refinement of the mesh, i.e., as a function of n_t. The geometric accuracy of the representation can be evaluated by means of the percent reduction in grain boundary area ΔA of the relaxed truncated octahedron

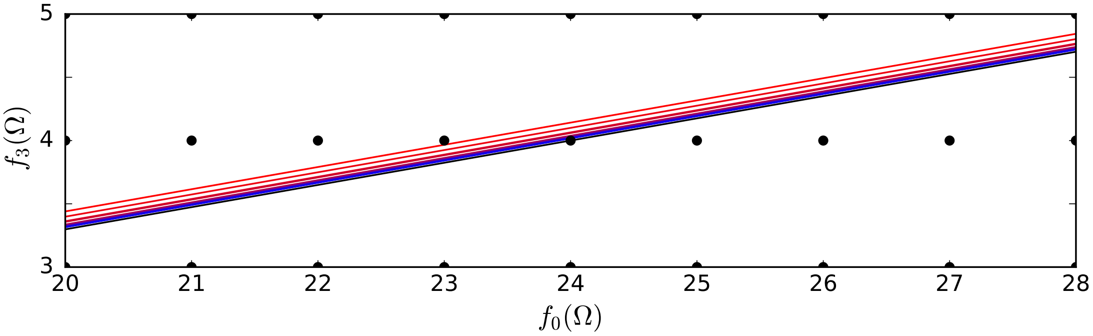


Figure 5.4. The lines defined by Eq. 5.2 for the data in Table 5.1 in the feasible region $3 \leq f_3(\Omega) \leq 5$ and $20 \leq f_0(\Omega) \leq 28$. The lines are colored from red to blue with decreasing error and the black line passing through $(24, 4)$ corresponds to the exact solution. The closest integer lattice point to the line is $(24, 4)$ for $n_t \geq 477$.

relative to the unrelaxed one. A detailed analysis [90] suggests a value of 0.159% for the continuous system; that ΔA does not converge to this value is likely due to the irregularity of the mesh. As for the integrated Gaussian curvature, the average quantities in Eq. 5.1 are equivalent to those for a single grain by periodicity. This implies that the integral of the Gaussian curvature over the interiors of the grain boundaries should be:

$$\int_{\partial G} K \, \mathrm{d}A = 4\pi - 24\alpha \approx -0.664484.$$

A conjugate gradient minimization algorithm and bootstrapping were used to fit the model $\int K \mathrm{d}A = a + b n_t^c$ to the data in Table 5.1, giving $a = -0.661 \pm 0.022$, $b = 2.08 \pm 0.51$, and $c = -0.378 \pm 0.060$ (reported as the medians and half the interquartile range). This implies that the integrated Gaussian curvature would be -0.661 ± 0.022 in the $n_t \to \infty$ limit, and is interpreted as numerically confirming Eq. 5.1 given the degree of approximation of the grain geometry. That the integrated Gaussian curvature converges to the expected value even though the percent area reduction does not confirms the assertion that Eqs. 5.1 and 5.2 are invariant to geometric perturbations of the structure, provided the numbers of grains and quadruple junction points remain the same and Plateau's laws are satisfied.

Alternatively, one could consider the feasibility of inferring $f_3(\Omega)$ and $f_0(\Omega)$ by means of a sufficiently accurate measurement of the integral of K over the grain boundaries of Ω in Eq. 5.2. This can be done by a graphical construction in the plane with $f_3(\Omega)$ and $f_0(\Omega)$ on the vertical and horizontal axes. Given the integral of the Gaussian curvature over the grain boundaries of Ω, Eq. 5.2 defines a line in this plane with an irrational slope.

Since the actual values of $f_3(\Omega)$ and $f_0(\Omega)$ are necessarily positive integers, this line passes through exactly one point on the integer lattice in the positive quadrant. In practice, any error in the measurement of the integrated Gaussian curvature would change the intercept with the vertical axis and shift the line off of the lattice point; whether this is an issue or not depends on the magnitude of the error and any a priori bounds that can be placed on $f_3(\Omega)$ and $f_0(\Omega)$. For example, Fig. 5.4 shows this construction for the data in Table 5.1 with the constraints $3 \leq f_3(\Omega) \leq 5$ and $20 \leq f_0(\Omega) \leq 28$. Since the magnitude of the error is assumed to be unknown, it is reasonable to suppose that the correct values of $f_3(\Omega)$ and $f_0(\Omega)$ correspond to the integer lattice point closest to the line within the feasible region. This procedure correctly identifies the relevant integer lattice point as $(24, 4)$ for $n_t \geq 477$; in general, the effect of integrated Gaussian curvature error is reduced as the area of the feasible region decreases.

Apart from advancing our fundamental understanding of grain structures, there remains the question of the practical utility of Eqs. 5.1 and 5.2 (and the analogues in Sec. 7.4 of the appendix). This question is made more pressing by there being few materials that actually evolve by the relevant ideal grain growth process; the grain boundary energy and mobility do generally depend on the grain misorientation and boundary plane normal, but in a way that is not precisely known even for simple model systems. The only way to conceivably measure grain boundary mobility is by a time series of three-dimensional microstructures, visualized using a non-destructive technique. While such techniques exist [4, 85], fitting to such data has not yet yielded single-valued mobilities [91], perhaps due to insufficient experimental resolution or confounding variables. As for grain boundary energies, despite the seminal work of Morawiec [6] indicating how the grain boundary energies could in principle be extracted from a three-dimensional microstructure, the grain boundary energies for, e.g., simple metals are not yet widely available in the literature. Finally, simulations of grain boundary properties [69, 92–94] out of necessity only consider a small subset of possible grain boundary characters, and cannot be validated in the absence of reliable experimental data.

Given this situation, the authors propose two possible applications of this chapter's results based on the differences of the left and right sides of Eqs. 5.1 and 5.2:

$$e_1 = \left\langle \int_{\partial G} K \, \mathrm{d}A \right\rangle - 4\pi + \alpha \langle f_0(G) \rangle$$

$$e_2 = \sum_{i=1}^{f_2(\Omega)} \int_{F_i} K \, \mathrm{d}A - 2\pi f_3(\Omega) + 2\alpha f_0(\Omega)$$

First, e_1 and e_2 could be used as rough measures of the deviation of a physical system from an ideal one (along with other quantities like the grain growth exponent and the grain size distribution), thereby contributing to the ongoing investigation of how severe is the assumption of ideal grain growth in practice. Second, there is widespread interest in generating microstructures (for use in, e.g., integrated computational materials engineering) by means of physics-based simulations of microstructure evolution. Faced with the absence of reliable grain boundary data, such simulations generally assume that grain boundary properties are constants, making this chapter's results relevant to the vast majority of contemporary microstructure evolution codes. In this context, e_1 and e_2 could be used to evaluate the accuracy of the geometric representation of a grain structure; the derivation above suggests that these quantities should be particularly sensitive to the geometry around triple junction lines and quadruple junction points. Since the angle conditions around triple junction lines are directly implicated in the rates of area and volume change of two-dimensional [49, 50] and three-dimensional [37, 76] grains, any deviations from Eqs. 5.1 and 5.2 could function as bounds on the maximum achievable accuracy of simulations of mean-curvature driven grain growth.

Chapter 6

Future work

As stated in Ch. 1, predictive modeling of thermomechanical deformation is an important part of the intetgrated computational materials engineering (ICME) vision. A natural extension of this dissertation project would be to include deformation physics (i.e., crystal plasticity) to investigate thermomechanical processes at the microstructural level and explore fabrication routes for advanced materials. Alternatively, interfacing with existing open source crystal plasticity libraries could enable such simulations without extensive library development.

ExaConstit developed by Lawrence Livermore National Laboratory is an open source crystal plasticity finite element (CPFE) library written in C++ which makes it an ideal candidate for integration [95], only requiring interfacing with the data representations at virtually no computational overhead. More importantly, the deformation and grain growth mechanisms will interact with each other in a single framework, making it easier to seamlessly integrate the numerical models. DAMASK developed by Max Planck Institute fr Eisenforschung is another candidate for integration [96]. However, since it is written in Fortran, it would most likely require the transfer of information between the two programs. Finally, FEpX is an open source finite element library for modeling elastoplastic deformation of polycrystalline solids written in Fortran [97]. If integration with any of these options proves problematic, a package could alternatively be implemented within VDlib. In terms of capabilities, ExaConstit currently only supports Dirichlet boundary conditions, so only function values (e.g., position) can be specified at the boundaries.

DAMASK supports grid-based FEM with mixed boundary conditions, but this would require mesh conversion and would likely employ a file-based interface which is expected to be less efficient; support for unstructured meshes is currently a development goal. FEpX supports unstructured meshes and both strain and load boundary conditions, but the possibility of integration has not been as extensively explored as the other options.

Adding crystal plasticity would enable simulations of microstructure evolution during thermomechanical processing. To our knowledge, a three dimensional, purely FEM code with support for deformation and anisotropic grain boundary properties (GBP) does not exist in the literature. Adding grain nucleation mechanisms would allow simulations of both static and dynamic recrystallization within a purely finite element framework. The platform could then be further extended to study additional physics and processing routes. For example, understanding mechanical twinning during high-strain rate deformation, developing models of damage accumulation and fracture initiation to be used in continuum codes, and searching the space of additive manufacturing processing parameters to reduce residual porosity are all high impact research directions for advanced materials All of these require microstructural information and are naturally studied at the microscale, i.e., at the length scales considered here.

Specific proposed studies include, listed roughly in order of difficulty, the following:

1. Implementing a strain gradient crystal plasticity model would allow better prediction of the local deformation response and more accurate investigation of thermomechamical processing and high temperature failure mechanisms.
2. An explicit model for deformation twinning [98] would allow comparison of simulations of microstructural twinning with in situ DIC measurements [99].
3. Twin boundaries are low energy boundaries and can thereby help to stabilize the microstructure to grain growth [100]. Realistic anisotropic grain boundary energy (GBE) functions obtained by three dimensional microscopy could be used to test the feasibility of a grain boundary engineering application on digital or experimentally obtained microstructures.
4. Hot isostatic pressure (HIP) is a post process that aims to reduce porosity of additively

manufactured materials by applying high pressures and elevated temperatures for long time periods. The relevant processes take place at the microstructure level, and accurate simulation of HIP to predict the ideal processing conditions essentially would require models working at the microstructure level [101] that include anisotropic GBE, are capable of representing multiple phases and phase separation, and could handle finite deformations.

Chapter 7

Appendix

7.1 Summary

Section 7.2 is the appendix to Ch. 3 and describes the fundamental operations on the topology and mesh that are involved in our approach. Sections 7.2.2 - 7.2.5 describe the algorithms related specifically to the topological transitions. Section 7.2.6 describes the relaxation procedure to prepare the mesh for a collapse and Sec. 7.2.7 describes the generalization of the collapse algorithm described in Sec. 3.2.1. Section 7.2.8 describes the local volume preservation procedure for simulating microstructures without periodic boundary conditions. An six grain configuration is used to test the insertion detection algorithms and the detected insertions are shown in Sec. 7.2.9. Section 7.2.10 gives an example configuration that shows that spurious insertions can be physically necessary even when grain boundary properties are constant and isotropic. Section 7.2.11 shows isolated grains to highlight the differences between the initial Voronoi tesselation and grains from an evolved microstructure. Finally, Sec. 7.2.12 discusses the scaling of the computational resource usage as a function of number of grains.

Section 7.3 is the appendix to Ch. 4 which is about the benchmarking of grain growth models. Section 7.3.1 describes the non-dimensionalization of the equations of motion and Sec. 7.3.2 shows the derivation for the analytical solution for the shrinking sphere case. Section 7.3.3 investigates the higher than expected scaling of the computational cost for the discrete equations of motion, identifies the cause as the stability of the equations of motion, and describes the remedies explored as part of this study.

Section 7.4 is the appendix to Ch. 5 which describes the constant of motion. Section 7.4.1 shows the steps of the derivation for the constant of motion, while Sec. 7.4.2 extends the derivation to the situation where the boundary conditions are not periodic.

7.2 Method

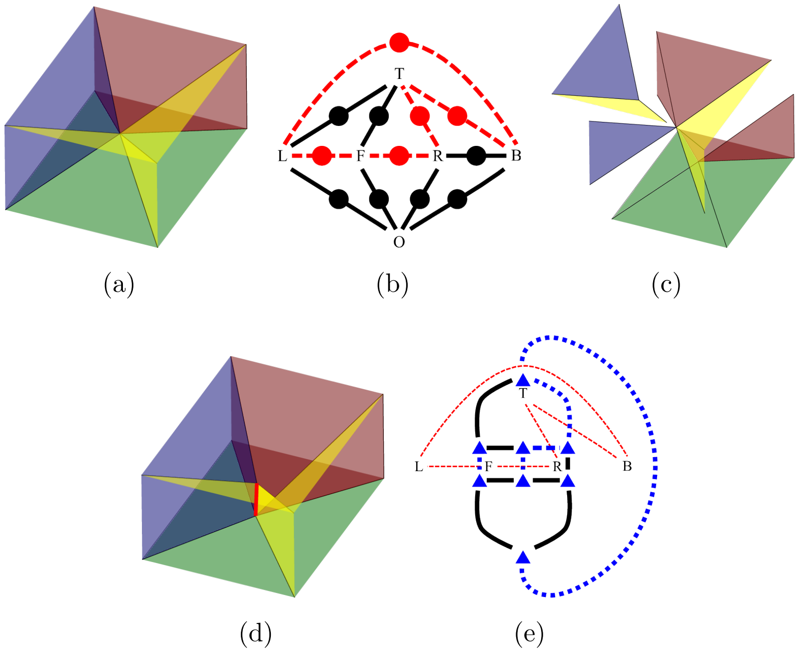
7.2.1 Notation

For brevity, let $\mathbf{S}^d$ be the set of d-dimensional strata and S_i^d the ith d-stratum. $|A|$ is the number of elements in the set A, $A^e(S_i^d)$ is the collection of S^e adjacent to S_i^d, and $A_j^e(S_i^d)$ the jth S^e adjacent to S_i^d. $A^{f,e}(S_i^d)$ are the f-dimensional strata adjacent to the e-dimensional adjacencies of S_i^d. $\tilde{S}_i^d$ is a newly inserted stratum.

The adjacency graph of S^2 and S^3 in the neighborhood of a 0-stratum S_i^0 will be used to enumerate the possible changes to the local microstructure. Let the adjacency graph around S_i^0 be G_i, and have nodes corresponding to the set $A^2(S_i^0) \cup A^3(S_i^0)$ and edges for each incidence of an S^2 and S^3. Similarly, G_i' is the adjacency graph with nodes corresponding to the set $A^1(S_i^0) \cup A^2(S_i^0)$ and edges for each incidence of an S^1 and S^2. Paths and circuits on G_i will be used to find possible S^2 and S^1 insertions around S_i^0, where a path is a sequence of non-repeating nodes connected by edges and a circuit is a path that begins and ends at the same node.

Let Q^i be the set of circuits on G_i and Q_j^i be the jth such circuit. Removing the circuit Q_j^i from the graph leaves two disjoint graphs of nodes which will be denoted as $H_1^{i;j}$ and $H_2^{i;j}$. The $H_k^{i;j}$ could be disconnected over G_i, but the corresponding strata around S_i^0 can always be connected through shared S^1, as shown in Fig. 7.1. If one of the $H_k^{i;j}$ is empty, that implies that the circuit Q_j^i is associated with an existing S^1 and should be discarded.

Let $P^{i;j,k}$ be the set of paths between S_j^3 and S_k^3 in the vicinity of S_i^0, and $P_l^{i;j,k}$ be the lth such path. Let $\wp(P^{i;j,k})$ be the set of sets of paths between S_j^3 and S_k^3, where every set contains at least two paths and none of the paths in the same set intersect. If l is the index for this set, then subgraph $H_m^{i;j,k;l}$ is the mth connected component remaining when the lth set of paths with the end points S_j^3 and S_k^3 is subtracted from G_i.



(a) (b) (c)

(d) (e)

Figure 7.1. A representation of how the circuit Q^i_j divides nodes into disjoint graphs $H^{i;j}_1$ and $H^{i;j}_2$, which are connected over S^1. (a) The initial microstructure consisting of six grains. (b) Removal of the red dashed circuit Q^i_j going through (T)op, (B)ack, (L)eft, (F)ront and (R)ight grains leaves two disjoint graphs. $H^{i;j}_1$ consists of the S^2 connecting the grains T-F and T-L. $H^{i;j}_2$ consists of the b(O)ttom grain, the S^2 bounding grain O in the neighborhood of S^0_i, and the S^2 connecting the grains R-B. (c) At the microstructure level, it is easy to see how the components of $H^{i;j}_1$ are connected by S^1s. (d) The final configuration after the insertion with the associated S^1 colored red. (e) G'_i, where Q^i_j is shown superposed and the dotted blue edges correspond to $S^3 - S^2 - S^3$ components of Q^i_j. The nodes of the two subgraphs of G'_i can be seen to be connected by solid edges.

The mesh is composed of simplicial finite elements, including the 0-dimensional vertices, 1-dimensional edges, 2-dimensional triangles and 3-dimensional tetrahedra. An n-dimensional simplicial element belongs to the stratum of lowest dimension in which it is contained, i.e., a vertex may belong to an S^0, S^1, S^2, or S^3, an edge may belong to an S^1, S^2, or S^3, etc. Similar to the notation for strata, we denote the ith member of the set of d-dimensional simplicial entities as Δ^d_i and use the adjacency operator $A^e(\cdot)$ in the same way to obtain the set of adjacent mesh entities of dimension e. Additionally, the stratum membership of a simplicial entity is indicated as $\Delta^d_i \in S^e_j$, or Δ^d_i belongs to S^e_j. The set of e-dimensional simplicial entities belonging to S^d_i is obtained by the membership operator $M^e(S_i^d)$. $S(\Delta_i^d)$ is the stratum that owns the simplicial entity Δ_i^d. A sample microstructure showing the simplicial entities outlined in red is provided in Fig. 2.1b.

7.2.2 Stratum collapse

Given a stratum S_i^d to collapse and a final point $\hat{S}^0$, recursively collapse the bounding lower dimensional strata and then remove S_i^d. If $\hat{S}^0$ is not specified, it is always possible to pick the first bounding S^0 (otherwise there are no S^0 remaining after collapse). Update the adjacency lists of the surrounding strata.

Algorithm 1 Collapse ($S_i^d, \hat{S}^0 \coloneqq \emptyset$)

```
Implement changes in the stratification when collapsing $S_i^d$.
if $d = 0$ then
    return
if $\hat{S}^0 = \emptyset$ then                                   ▷ Assign $\hat{S}^0$ if not specified.
    if $A^0(S_i^d) \neq \emptyset$ then
        $\hat{S}^0 \coloneqq A_1^0(S_i^d)$
if $d = 1$ then                                               ▷ Replace the merging $S^0$ with $\hat{S}^0$.
    for $S_j^1 \in A^{1,0}(S_i^1)$ do
        for $S_k^0 \in A^0(S_j^1)$ do
            if $S_k^0 \in A^0(S_i^1)$ then
                $A_k^0(A_j^{1,0}(S_i^d)) \coloneqq \hat{S}^0$
        if $|A^0(S_j^1)| = 2$ and $A_1^0(S_j^1) = A_2^0(S_j^1) = \hat{S}^0$ then ▷ If $\hat{S}^0$ is repeated remove one.
            $A^0(S_j^1) \coloneqq \{\hat{S}^0\}$
else                                                          ▷ Collapse the bounding strata.
    for $S_j^d \in A^{d-1}(S_i^d)$ do
        Collapse ($S_j^d, \hat{S}^0$)
if $d < 3$ then                                               ▷ Remove the collapsing strata.
    for $S_j^{d+1} \in A^{d+1}(S_i^d)$ do
        $A^d(S_j^{d+1}) \coloneqq A^d(S_j^{d+1}) \setminus \{S_i^d\}$
```

7.2.3 1-stratum insertion

Given a candidate S_i^0 and a circuit Q_j^i on G_i insert the new stratum $\tilde{S}^1$ corresponding to Q_j^i. Add the new strata $\tilde{S}^0$ and $\tilde{S}^1$ to the stratification and set the S^0 adjacencies of $\tilde{S}^1$ as $\{S_i^0, \tilde{S}^0\}$. Update the adjacency lists of the surrounding strata.

Algorithm 2 S^1 insertion (S_i^0, Q_j^i).

```
Implement changes in the stratification when inserting $\tilde{S}^1$ using $Q_j^i$.
Create new strata $\tilde{S}^1$, $\tilde{S}^0$.
$A^0(\tilde{S}^1) \coloneqq \{S_i^0, \tilde{S}^0\}$                      ▷ Adjacency of $\tilde{S}^1$.
for $S_k^2 \in Q_j^i$ do                      ▷ Add $\tilde{S}^1$ to $A^1(S_k^2)$, for $S^2$ on $Q_j^i$.
    $A^1(S_k^2) \coloneqq A^1(S_k^2) \cup \{\tilde{S}^1\}$
for $S_k^2 \in H_2^{i;j}$ do                      ▷ Replace $S_i^0$ with $\tilde{S}^0$.
    for $S_l^1 \in A^1(S_k^2)$ do
        if $S_i^0 \in A^0(S_l^1)$ then
            $A^0(S_l^1) \coloneqq A^0(S_l^1) \cup \{\tilde{S}^0\} \setminus \{S_i^0\}$
```

7.2.4 2-stratum insertion

Given a candidate S_i^0 and a set of paths $\wp_l(P^{i;j,k})$ on G_i, insert the corresponding new stratum $\tilde{S}^2$. Add the new strata $\tilde{S}^2$, $\tilde{S}_m^0$ for $m = 1 : |\wp_l(P^{i;j,k})| - 1$, and $\tilde{S}_m^1$ for $m = 1 : |\wp_l(P^{i;j,k})|$ to the stratification. Set the $(d-1)$-dimensional adjacency lists of the new strata $\tilde{S}^2$ and $\tilde{S}_m^1$ for $m = 1 : |\wp_l(P^{i;j,k})|$. Update the adjacency lists of the surrounding strata.

Algorithm 3 S^2 insertion ($\wp_l(P^{i;j,k})$).

```
Implement changes in the stratification when inserting a $\tilde{S}^2$ using $\wp_l(P^{i;j,k})$.
Create new stratum $\tilde{S}^2$.
Create new strata $\tilde{S}^0_m$ for $m \coloneqq 1 : |\wp_l(P^{i;j,k})| - 1$.        ▷ In addition to $S^0_i$.
Create new strata $\tilde{S}^1_m$ for $m \coloneqq 1 : |\wp_l(P^{i;j,k})|$.
$A^2(S^3_j) \coloneqq A^2(S^3_j) \cup \{\tilde{S}^2\}$        ▷ Add $\tilde{S}^2$ to $A^2(S^3_j)$.
$A^2(S^3_k) \coloneqq A^2(S^3_k) \cup \{\tilde{S}^2\}$        ▷ Add $\tilde{S}^2$ to $A^2(S^3_k)$.
$A^1(\tilde{S}^2) \coloneqq \{\tilde{S}^1_1, \tilde{S}^1_2, \ldots, \tilde{S}^1_m\}$ with $m \coloneqq |\wp_l(P^{i;j,k})|$        ▷ Set the adjacency of $\tilde{S}^2$.
$A^0(\tilde{S}^1_1) \coloneqq \{S^0_i, \tilde{S}^0_1\}$        ▷ Set the adjacencies of $\tilde{S}^1$.
for $m \coloneqq 2 : |\wp_l(P^{i;j,k})| - 1$ do
    $A^0(\tilde{S}^1_m) \coloneqq \{\tilde{S}^0_{m-1}, \tilde{S}^1_m\}$
$A^0(\tilde{S}^1_m) \coloneqq \{\tilde{S}^0_{m-1}, S^0_i\}$ with $m \coloneqq |\wp_l(P^{i;j,k})|$
for $P^{i;j,k}_m \in \wp_l(P^{i;j,k})$ do        ▷ Add $\tilde{S}^1$ to the adjacency lists of the $S^2$ on path $P^{i;j,k}_m$.
    for $S^2_o \in P^{i;j,k}_m$ do
        $A^1(S^2_o) \coloneqq A^1(S^2_o) \cup \{\tilde{S}^1_m\}$
for $m \coloneqq 2 : |\wp_l(P^{i;j,k})|$ do        ▷ For $S^1$ adjacent to $S^2 \in H^{i;j,k;l}_m$, replace the $S^0_i$ with $\tilde{S}^0_m$.
    for $S^2_o \in H^{i;j,k;l}_m$ do
        for $S^1_p \in A^1(S^2_o)$ do
            if $S^0_i \in A^0(S^1_p)$ then
                $A^0(S^1_p) \coloneqq A^0(S^1_p) \cup \{\tilde{S}^0_{m-1}\} \backslash \{S^0_i\}$
```

7.2.5 Check spurious strata

Spurious stratum insertions can occur for a S^1 insertion if there are two S^2 on the circuit that bound the same S^3, or for a S^2 insertion between two components of the same S^3. An example configuration leading to such an event is shown in Fig. 3.6 of the main text.

Algorithm 4 Check spurious (S_i^d).

```
*Compare the upper adjacencies of $(S_i^d)$ to check if it is spurious.*
if $d = 0$ or $d = 1$ then                      ▷ Higher adjacency rule for valid $S^0$ and $S^1$.
    return $|A^{d+1}(S_i^d)| < 3$
else
    if d = 2 and $|A^{d+1}(S_i^d)| = 2$ then    ▷ If $S^2$ bounds the same $S^3$, it is spurious.
        return $A_1^{d+1}(S_i^d) = A_2^{d+1}(S_i^d)$
return FALSE
```

7.2.6 Relaxation during collapse

The preconditioning operation relaxes the positions of the surrounding vertices to prevent inversions of the surrounding tetrahedra during a collapse. More specifically, the vertices connected to the collapsing strata by an edge form a hull. Positions of the vertices on the hull are found such that the surrounding tetrahedra do not invert during collapse of the stratum by a conjugate gradient search to minimize the positive definite potential ϕ. This is defined as

$$\phi = \sum_{i \in \Delta^0} \phi_i,$$

$$\phi_i = \frac{1}{w} \ln \left\{ \sum_{\Delta_j^3 \in A^3(\Delta_i^0)} \exp\left[-w V_j(\bar{v})/V_t\right] + 1 \right\},$$

where Δ^0 is the set of vertices on the hull, $\bar{v}$ is the position vector of all vertices, V_j is the volume of jth tetrahedron, and w is a weight for scaling the exponent. w is defined as $80\frac{V_t}{\text{abs}(V_n)+\epsilon}$, where V_t is the total starting volume of all tetrahedra surrounding the hull vertices, and $V_n = \min(V_j)$. In practice ϵ is 2.22507×10^{-298}, 10^{10} times the smallest representable double. Using the algorithm shown in Fig. 7.2, the volumes are updated until $V_n > v_{th}$, where v_{th} is the desired volume ratio of the surrounding tetrahedra at the collapsed configuration defined as $v_{th} = (V_m + \epsilon) \times 10^{-5}$ where $V_m = \min(\text{abs}(V_j))$. After each time the positions of the hull vertices are updated, V_n and V_m (but not w or v_{th}) are updated. If the smallest volume V_n is smaller than twice the starting most negative volume

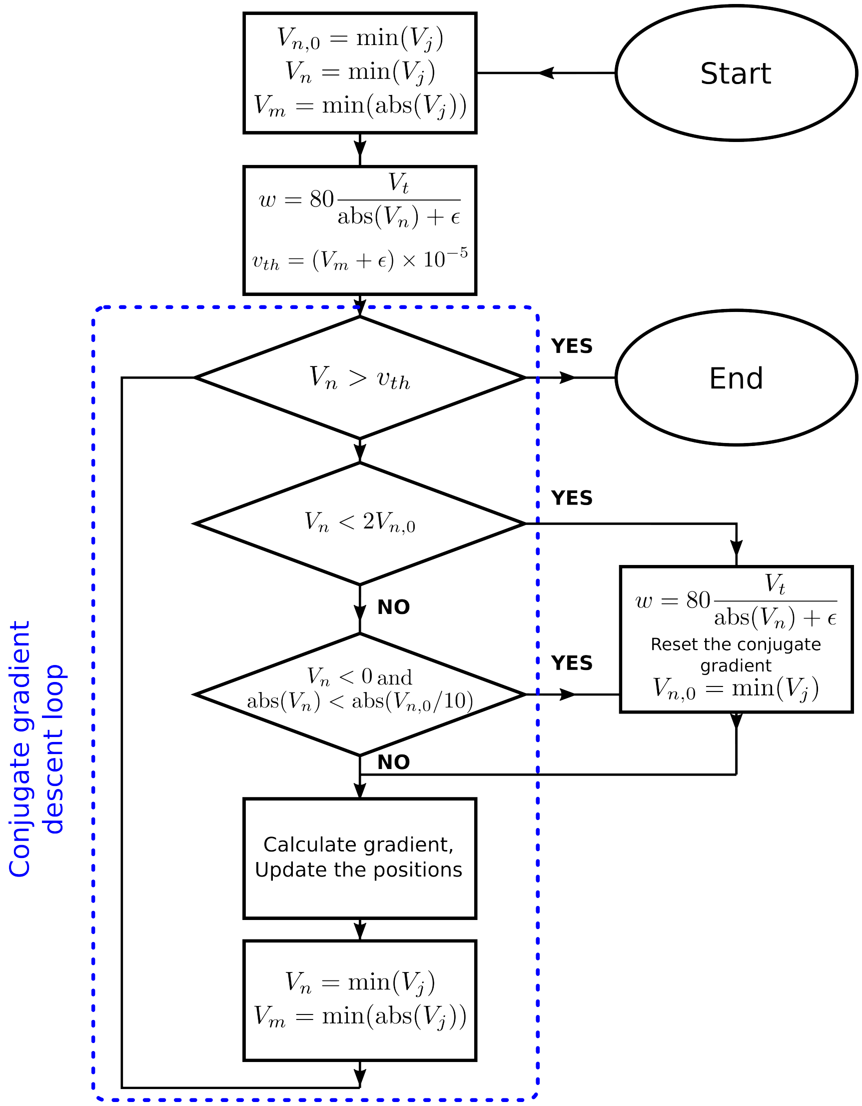


Figure 7.2. The flow chart for the calculation of the volumes and the update of the parameters w, v_{th} required for the conjugate gradient descent calculations.

$V_{n,0}$, or $V_n < 0$ and $\mathrm{abs}(V_n) < \mathrm{abs}(V_{n,0}/10)$, w is updated and the conjugate gradient is reinitialized to increase the convergence rate.

The negative of the gradient of the potential is given by

$$-\overline{\nabla}_i\phi = -\overline{\nabla}_i\phi_i - \sum_{\Delta_j^0 \in A^{0,1}(\Delta_i^0)} \overline{\nabla}_i\phi_j,$$

$$-\overline{\nabla}_i\phi_i = \sum_k \left[\frac{\exp(-wA_k)}{\sum_l \exp(-wA_l)+1}\overline{\nabla}_i V_k\right],$$

where $k, l \in A^3(\Delta_i^0)$ and $A_k = V_k(\bar{v})/V_t$. The form of $-\overline{\nabla}_i\phi_j$ is the same but $k, l \in A^3(\Delta_i^0)\cap$

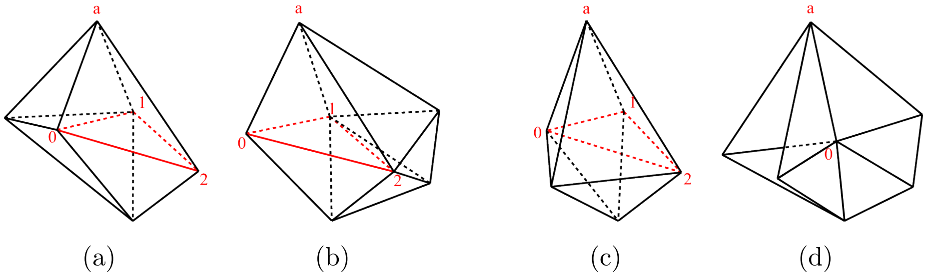


Figure 7.3. The generalized lens collapse corresponding to the triangle $(0, 1, 2)$. Merging entities form sets rather than couples and an entity might be merging in one lens and collapsing in another, and will collapse during the stratum collapse. The lens corresponding to edge (a) $(0, 1)$, (b) $(1, 2)$, (c) $(0, 2)$. In the lenses corresponding to edges $(0, 2)$ and $(1, 2)$ the triangle $(0, 1, a)$ is merging, but since the triangle is collapsing in the lens of edge $(0, 1)$, it is collapsing. Edges $(0, a)$, $(1, a)$, and $(2, a)$ form a merging set. (d) The final configuration after the collapse.

$A^3(\Delta_j^0)$. It is possible that due to the starting geometry a non-inverting configuration with a minimum volume of v_{th} cannot be found. The relaxation continues until either a non-inverting configuration is found or the limit for number of iterations is reached.

7.2.7 Generalized collapse of multiple lenses

The generalized stratum collapse follows the same procedure described in Sec. 3.2.1 for lens collapse, i.e., the main steps are preconditioning the mesh, finding the stratum memberships of the remaining entities, destroying old entities, and regenerating the entities using the last remaining vertex. When there is more than one edge in the collapsing stratum, it is possible that some merging entities form sets rather than couples to form a new entity. Furthermore, an entity can be a merging or a collapsing entity in different lenses, in which case all merging entities in the associated set will collapse as shown in Fig. 7.3. Similar to lens collapse, for each set of merging entities a new entity will be regenerated by replacing the merging vertex with the final vertex and using the new stratum membership.

7.2.8 Using an exterior shell for volume preservation

Evolving down the gradient of surface energy, a non-periodic mesh will not preserve volume without additional constraints. Volume preservation is achieved by creating a stratification composed of the simulation cell corners, edges and surfaces. These strata are called 0-, 1-, and 2-shells, respectively. The shells are determined at the start of the simulation. In

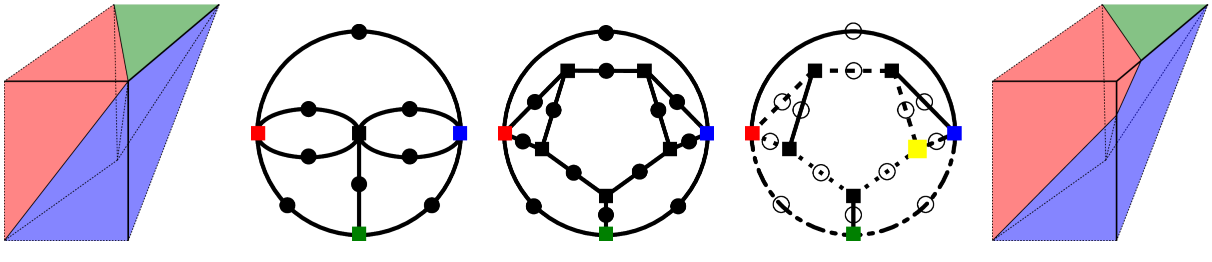

(a) (b) (c) (d) (e)

Figure 7.4. Exterior S^0 insertions require additional exterior S^2 and S^3 in the adjacency graph. (a) The red, blue and green S^3 at the corner of the simulation cell, having two, two, and one surface S^2, respectively. (b) The adjacency graph obtained by including a node for the exterior S^3 which doesn't allow any new surface S^2 insertion. (c) Instead of a single exterior S^3, S^2 and S^3 are included in the adjacency graph for each surface S^1 and S^2, respectively. (d) The augmented adjacency graph in (c) can be used to detect surface S^2 insertions, e.g., a trigon insertion using the dashed, dotted and dash-dotted paths between the exterior and the red grain. (e) The corresponding change in the microstructure.

this section, surface strata will indicate strata on the simulation cell boundary. Each S^0 is first tested to identify those on the simulation cell corners, edges or surfaces, and ones on the corners are attached to 0-shells. Next, the S^1 on the simulation cell boundaries are tested to identify those on the simulation cell edges by a depth first search, and ones on the edges are attached to 1-shells. The 2-shells are constructed similarly. During the simulation the motions of the vertices on these shells are projected onto the corresponding shell to preserve the total volume.

Any newly inserted strata during stratum insertions around exterior S^0 are associated with the appropriate shells. Since the exterior can be multiply connected to the volumes touching the exterior surface, one artificial exterior S^3 is created for each disconnected mesh component of the surface S^2, as shown in Fig. 7.4. The paths and circuits detected on this augmented adjacency graph contain multiplicities as there is actually a single exterior S^3. These are removed by replacing all artificial strata on the paths and circuits with the only exterior S^3 and only allowing uninterrupted segments of the artificial strata on a single circuit or path.

Finally, computing the convex hull for collapses of strata touching the exterior shell requires that the positions of vertices belonging to the collapsing stratum be added to the

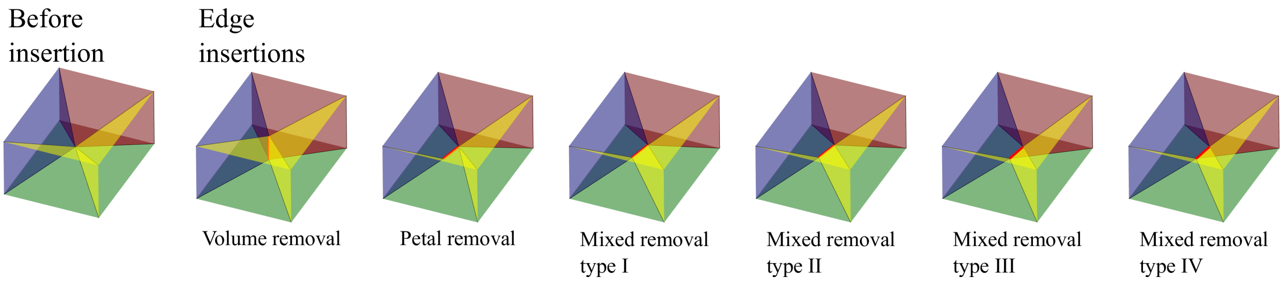


Figure 7.5. To demonstrate the capabilities of the method, a configuration formed by six grains meeting at the center of a cube is generated. The possible classes of insertions are more numerous than in Fig. 3.13. Some examples are shown denoted by the number of 2-strata on the $H_1^{i;j}/Q_j^i/H_2^{i;j}$, though this is not a complete descriptor (e.g., there are three 3/6/3 type insertions). In addition to these, digon, trigon and tetragon insertions between each three disconnected S^3 couples are possible.

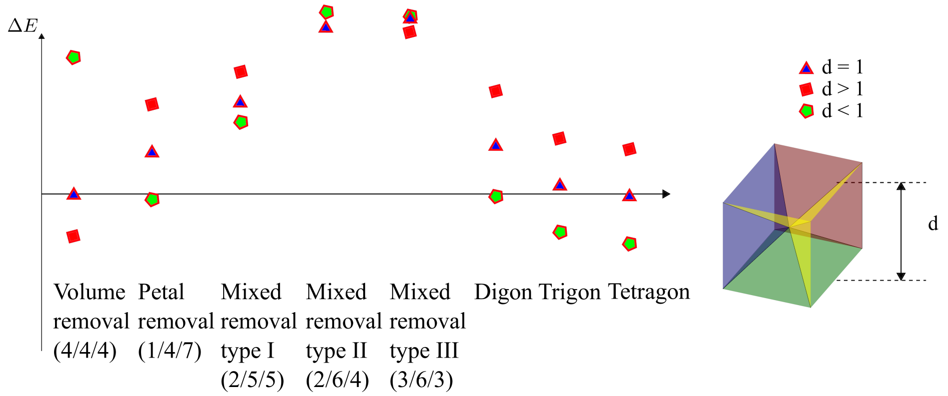
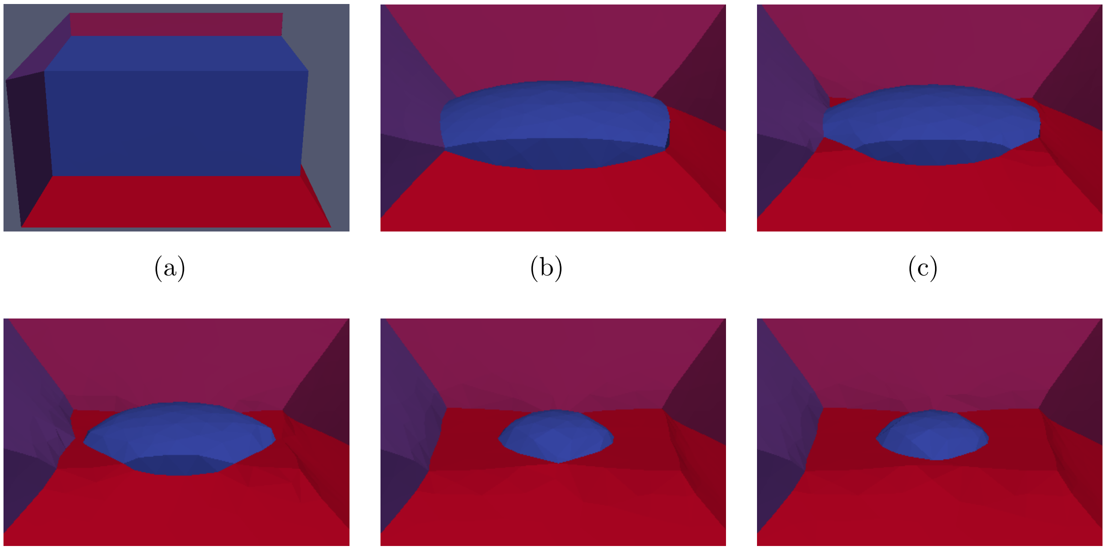


Figure 7.6. The variation of energy with changing dihedral angle configuration. The degenerate configuration is when the outer dimensions correspond to a cube. Red squares denote the case of the cube stretched in one direction and green pentagons denote the compressed case.

set of points.

7.2.9 Six grain configuration

As a further demonstration of the insertion detection, a more complicated configuration with six grains is generated and some of the possible S^1 insertions are shown in Fig. 7.5. This list is not exhaustive, but demonstrates the capability of the detection algorithm. In addition to these, digon, trigon, and tetragon S^2 insertions are possible. Similar to Fig. 3.14, the dependence of the energy change for different insertions on the dihedral angle is shown in Fig. 7.6.



(a) (b) (c)

(d) (e) (f)

Figure 7.7. (a) A symmetric seven grain configuration composed of a rectangular prismatic grain surrounded by six grains, one on each face. The top, front and right grains have been removed. (b) The vertical lines are about to collapse, and will be followed by (c) insertion of four trigons connecting the top and bottom grains. (d) The first two spurious insertions: the digons connecting the right and left grains to the center grain collapse and two spurious lines are inserted. Two surfaces connect the top and bottom grains. (e) After the collapse of the digons connecting the front and back grains to the center grain. (e) Last two spurious insertions: there is a single surface connecting the top and bottom grains and the center grain is embedded in this surface.

7.2.10 Mesh level examples of spurious insertions

Figure 7.7 shows that spurious insertions can be necessary for normal grain growth, though they are not frequently handled by grain growth codes. Specifically, two spurious insertions happen between Figs. 7.7c and 7.7d, and two more between Figs. 7.7e and 7.7f. Moreover, Fig. 7.7f shows a configuration containing a S^2 that cannot be deformed into a disc (the surface in which the blue volume is embedded), and a S^1 with no bounding S^0s (around the equator of the blue volume).

7.2.11 Evolution of isolated grains

Figure 7.8 more clearly shows the discretization of three isolated grains at the start of the simulation and at $t = 0.029$. The target refinement is two edges per median S^1 length, and each S^1 is required to contain at least two edges.

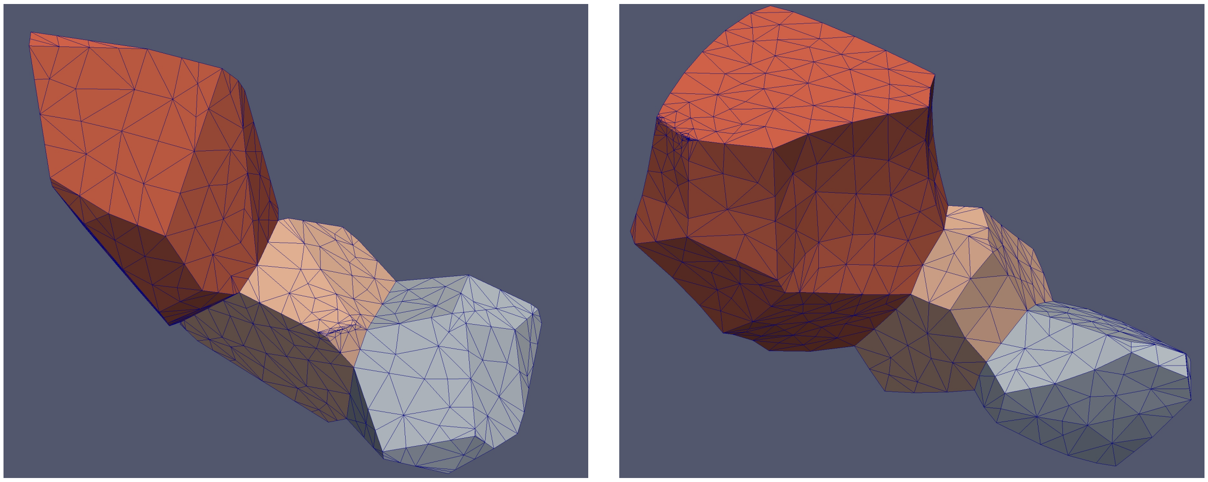

(a) (b)

Figure 7.8. Three selected grains at (a) $t = 0$ and (b) $t = 0.029$. Observe the expansion of the exterior surface of the large orange grain on the top.

7.2.12 Scaling of computational resource usage

Figure 7.9a shows the scaling of the computational resource usage with the number of grains on a PC with 16 GB of 2400MHz DDR4 RAM, and Intel(R) Core(TM) i7-7700HQ CPU during simulations until 5% total boundary area reduction was achieved. The wall clock time for each subprocess unexpectedly scaled roughly quadratically with the the number of grains. The equations of motion calculations and mesh adaptation (the minimum required for the simulation to progress) used $68 - 71\%$ of the total time, and Fig. 7.9b shows that the equations of motion scale linearly with the number of grains for a fixed number of iterations. Hence the source of the overall quadratic scaling is not clear, but likely signifies the potential for further optimization. In addition, the time step was found to reduce by several orders of magnitude around a topological transition; since the number of such events increases with the number of grains, the number of iterations in a fixed interval of time increases as well and likely contributes to the scaling.

The cost of calculating the topological transitions scaled with the cube of the number of grains, but only required between $1 - 6.5\%$ of the total time. The calculations of the rate of volume change using the discretized MacPherson–Srolovitz relation used $25 - 28\%$ of the total time, but would be skipped in a production run. The target edge length is half of the median S^1 length, and every S^1 is represented by at least two edges. The memory usage was tested during a single iteration for numbers of grains ranging from 10 to 1000 and for two target refinement edge lengths, with the results in Fig. 7.9c. The memory usage scaled linearly with the number of grains, with the 1000 grain simulation having a peak simulation memory demand of 983 MB. Halving the target edge length (increasing the resolution of the mesh) increased the memory requirements to 6874 MB.

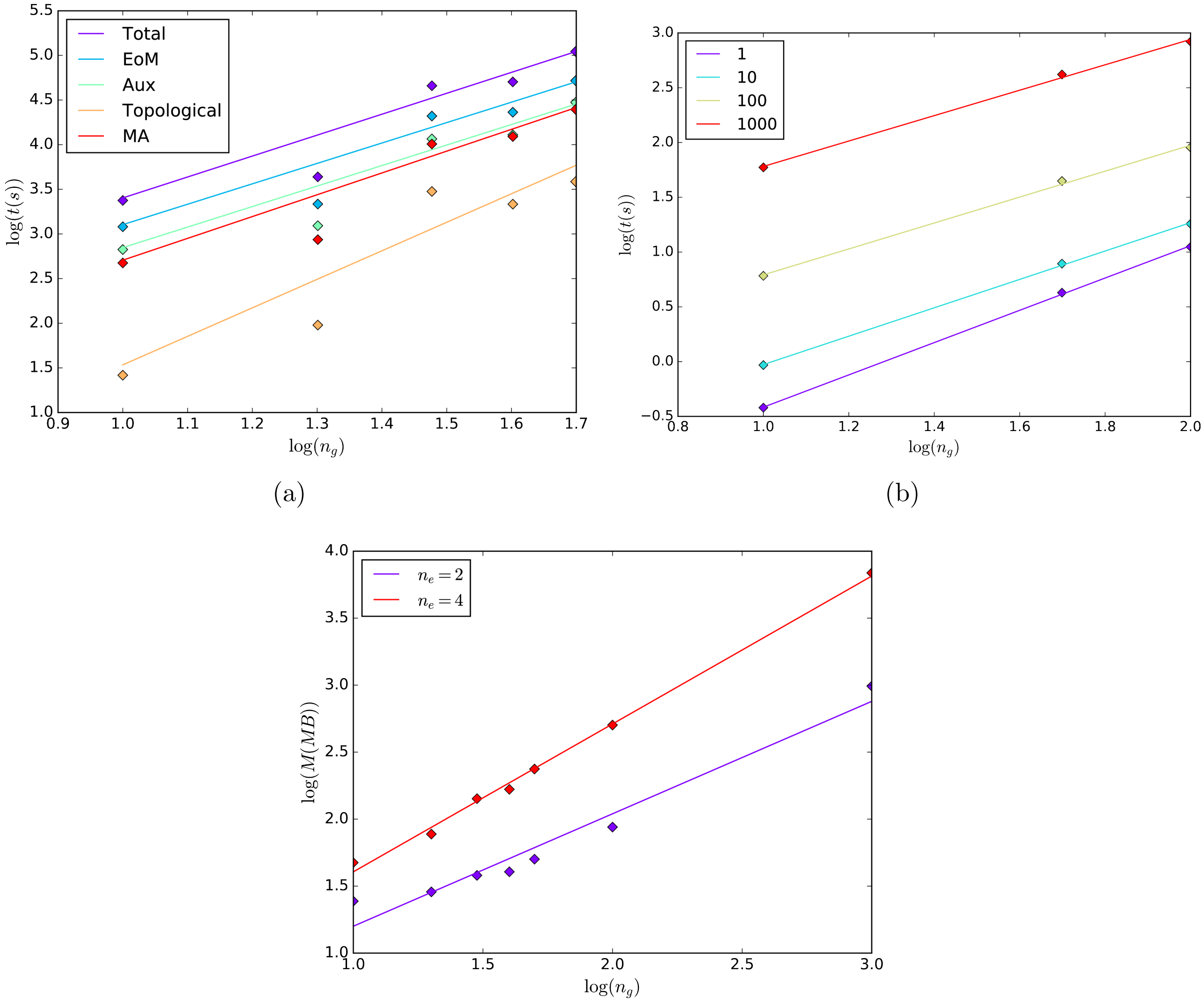
The simulations in Sec. 3.5 of the main manuscript required about a week to reduce the number of grains from 100 to 50 and the total grain boundary area by 35%. The benchmark cases reported in this section used a topological transition threshold of 1/50th of the median S^1 length as compared to 1/10th used in Sec. 3.5 for robustness against perturbations by mesh adaptation. Based on the observed scaling, the duration of an equivalent simulation with a topological transition threshold of 1/50th of the median S^1 length is estimated to be around 27 days on the same workstation.

7.3 Benchmarks

7.3.1 Nondimensionalization

Define the variable L to be the characteristic length scale of the grain structure defined in Section 4.2. For a sphere it is the sphere radius, for the TJ it is the length of the simulation cell in the direction normal to the consumed grain boundary, and for the QP it is the hexagonal grains's minor diameter. The Turnbull equation in Eq. (4.1) suggests that there is a characteristic time scale $\tau = L^2/(m\gamma)$. The simulations are performed with nondimensionalized time $\tilde{t} = t/\tau$, nondimensionalized space $\tilde{x} = x/L$, nondimensionalized rate of volume change $d\tilde{V}/d\tilde{t} = (\tau/L^3)dV/dt$, etc.

With respect to the quantities defined in Section 4.1.1, suppose that $\tau_l(\hat{\boldsymbol{t}}_i) = 0$, $\delta_0 = 0$, $\delta_1(\hat{\boldsymbol{t}}_i) = 0$, and that $\gamma(\hat{\boldsymbol{n}}_{ij})$ and $\delta_2(\hat{\boldsymbol{n}}_{ij})$ are constants. The governing equations of the



(a)

(b)

(c)

Figure 7.9. (a) The scaling of the wall clock time usage for each subprocess: equations of motion calculations (EoM), auxiliary operations (Aux) which here are MacPherson–Srolovitz calculations, topological transitions, and mesh adaptation (MA). (b) The scaling of the wall clock time usage for EoM calculations as a function of the number of grains and a fixed numbers of iterations. (c) The scaling of the memory usage for two different levels of refinement where n_e is the number of edges per median S^1 length.

discrete interface model then reduce to:

$$\boldsymbol{F} = \frac{\gamma}{2} \sum_i ||\boldsymbol{t}_i|| \sum_{j:\{i,j\}\in\Delta} \hat{\boldsymbol{n}}_{ij} \times \hat{\boldsymbol{t}}_i, \tag{7.1}$$

$$\boldsymbol{D} = \frac{\delta_2}{6} \sum_{i,j\in\Delta} ||\boldsymbol{t}_i \times \boldsymbol{t}_j|| (\hat{\boldsymbol{n}}_{ij} \otimes \hat{\boldsymbol{n}}_{ij}), \tag{7.2}$$

$$\boldsymbol{v} = \boldsymbol{D}^{-1}\boldsymbol{F}. \tag{7.3}$$

The nondimensionalized versions of these equations are

$$\tilde{\boldsymbol{F}} = \frac{\boldsymbol{F}}{L\gamma} = \frac{1}{2}\sum_i ||\tilde{\boldsymbol{t}}_i|| \sum_{j:\{i,j\}\in\Delta} \hat{\boldsymbol{n}}_{ij} \times \hat{\boldsymbol{t}}_i, \tag{7.4}$$

$$\tilde{\boldsymbol{D}} = \frac{\boldsymbol{D}}{L^2\delta_2} = \frac{1}{6}\sum_{i,j\in\Delta} ||\tilde{\boldsymbol{t}}_i \times \tilde{\boldsymbol{t}}_j||(\hat{\boldsymbol{n}}_{ij} \otimes \hat{\boldsymbol{n}}_{ij}), \tag{7.5}$$

$$\tilde{\boldsymbol{v}} = \frac{\tau}{L}\boldsymbol{v} = \frac{\tau\gamma}{L^2\delta_2}\tilde{\boldsymbol{D}}^{-1}\tilde{\boldsymbol{F}}, \tag{7.6}$$

where $\delta_2 = 3/\mu$ when the triple line and quadruple point drags vanish; this can be derived by requiring that the limiting behavior of a small spherical cap coincides with the predictions of Eq. (4.1).

The corresponding nondimensionalization of the multiphase field governing equations in Sec. 4.1.2 yields

$$\frac{\partial\eta}{\partial\tilde{t}} = -\tau L\frac{\delta W}{\delta\eta_n} = \frac{\partial\tilde{w}}{\partial\eta_n} + k\tilde{\Delta}\eta_n, \qquad \tilde{w} = w\,\tau\,L, \qquad \tilde{\Delta} = \tau Lk\Delta. \tag{7.7}$$

In this work, all multiphase field calculations are performed with dimensional values and then nondimensionalized for comparison to discrete interface simulations.

7.3.2 Spherical grain

If grain boundary properties are constant and isotropic, then the evolution of a spherical grain is self-similar and can be completely described by the radius $r_t(t)$ as a function of time. Since the mean curvature is $K = 2/r_t$, Eq. (4.1) implies that

$$dr_t/dt = -2\mu\gamma/r_t. \tag{7.8}$$

Setting $t_0 = 0$ and $r_t(0) = r_0$ and integrating gives

$$r_t(t) = \sqrt{r_0^2 - 4\mu\gamma t} \tag{7.9}$$

as the solution to this differential equation. Since the characteristic length scale for a sphere is r_0, nondimensionalizing reduces this to

$$\tilde{r}_t(\tilde{t}) = \sqrt{1 - 4\tilde{t}} \tag{7.10}$$

for the black curve in Fig. 4.3.

7.3.3 Scaling analysis

As described in Sec. 4.3, while the computational cost of the discrete interface method is expected to scale as ℓ_e^{-3}, the actual scaling is instead $\ell_e^{-4.089}$. Closer investigation revealed that the time step could decrease or increase by multiple orders of magnitude depending on the presence of various local mesh configurations. The boundary triangles exert capillary forces only in the boundary plane, yet exert drag forces only in the out-of-plane direction. This allows vertices on nearly-flat grain boundary sections to experience arbitrarily large lateral velocities, slowing the simulation down as the time step is reduced to prevent element inversion. The discrete method simulations in Sec. 4.2.1 include an isotropic contribution $\boldsymbol{D}_{I,d} = A_m^2/(md)\boldsymbol{I}$ to the drag tensor such that $\boldsymbol{v} = (\boldsymbol{D} + \boldsymbol{D}_{I,d})^{-1}\boldsymbol{F}$, where A_m^2 is the mean triangle area over the whole simulation, m is the mobility, and $d = 1000$ is a drag ratio. Decreasing the drag ratio reduces the lateral velocities, but also slows down the actual motion of the boundary and introduces a systematic error.

As an alternative, a contribution to the drag tensor that only acts in the in-plane directions could be constructed as follows. For simplicity, consider a closed disk of coplanar triangles around a vertex. Iterating over each grain boundary triangle Δ_{ij} adjacent to the central vertex, find the relative positions of the other vertices from the central vertex $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ and construct the outer product of the difference $\boldsymbol{p}_i - \boldsymbol{p}_j$ with itself. Let Λ_{max} be the largest eigenvalue of the sum of the outer products, and define the matrix $\boldsymbol{C} = \sum_{i,j \in \Delta} (\boldsymbol{p}_i - \boldsymbol{p}_j) \otimes (\boldsymbol{p}_i - \boldsymbol{p}_j)/\Lambda_{max}$. The anisotropic drag tensor contribution $\boldsymbol{D}_{a,d} = A_m^2/(md)\boldsymbol{C}$ by construction has no effect on the grain boundary motion in the plane normal direction. This should allow the lateral velocities of boundary vertices to be reduced while introducing less systematic error in the motion of non-planar boundaries than for an isotropic drag. An example triple junction mesh configuration is shown in Fig. 7.10 to qualitatively demonstrate the effect of different drag tensor correction terms. Although the velocity associated with $\boldsymbol{D}_{I,d}$ aligns with the force direction faster with increasing d, the velocity term in the vertical direction is also attenuated more compared to $\boldsymbol{D}_{a,d}$.

The difference in the expected and the actual scaling of the cost can largely be attributed

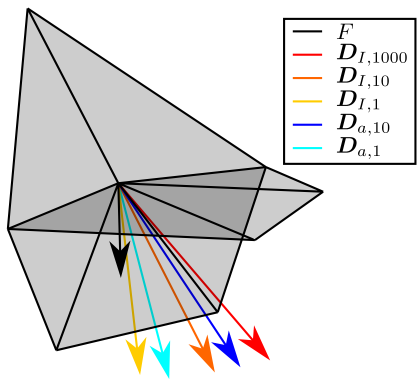


Figure 7.10. The effect of different drag tensor correction terms on the resulting velocity. The capillary force is colored black and the velocities corresponding to different correction terms are differentiated by color. Each vector is scaled relative to the maximum magnitude among the velocities.

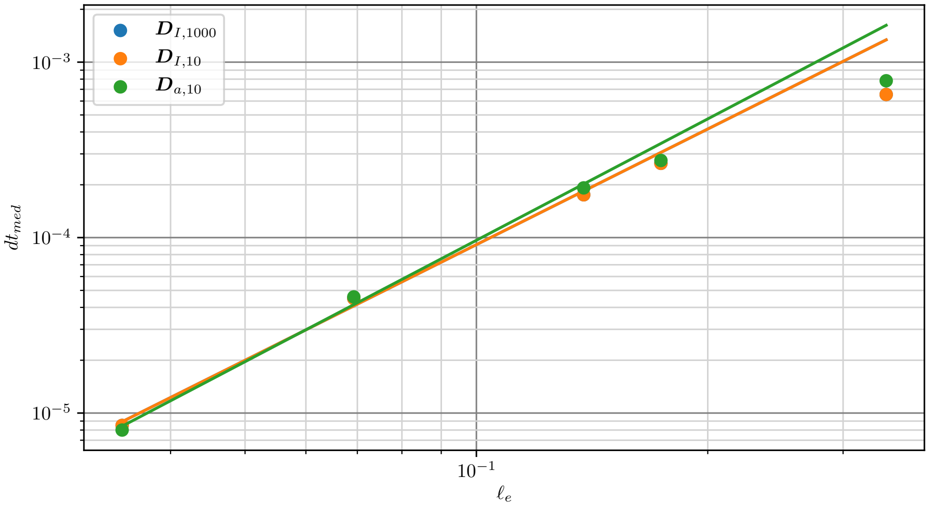


Figure 7.11. The scaling of the median time step with ℓ_e for the three drag tensor correction terms.

to the non-linear scaling of the median time step shown in Fig. 7.11. dt_{med} scales as $\ell_e^{2.010}$ for $\boldsymbol{D}_{a,10}$ and $\ell_e^{1.916}$ for $\boldsymbol{D}_{I,1000}$ and $\boldsymbol{D}_{I,10}$. Overall, $\boldsymbol{D}_{a,10}$ allows larger time steps and has a better accuracy, though the improvement is not significant.

7.4 Constant of motion

7.4.1 Derivation of Eqs. 1 and 2

Suppose that Ω is a grain structure in a domain with periodic boundary conditions, that Ω satisfies Plateau's laws (i.e., grain boundaries meet at dihedral angles of $2\pi/3$ and triple junction lines meet at angles of $\cos^{-1}(-1/3)$), and that in Ω every triple junction line is bounded by two distinct quadruple junction points, every grain boundary is homeomorphic to a disk and bounded by two or more distinct triple junction lines, and every grain is homeomorphic to a ball and bounded by three or more distinct grain boundaries.

The initial objective is to establish the following version of the Gauss–Bonnet theorem for a grain G belonging to Ω:

$$\int_{\partial G} K \,\mathrm{d}A + \sum_{i=1}^{f_2(G)} \int_{\partial F_i} \kappa_g \,\mathrm{d}s + \sum_{i=1}^{f_0(G)} \alpha_i = 2\pi\chi(\partial G) \tag{7.11}$$

where K is the Gaussian curvature of a grain boundary, ∂G indicates the grain boundary interiors, κ_g is the geodesic curvature of a triple junction line, ∂F_i indicates the interiors of the triple junction lines bounding the ith grain boundary, $f_2(G)$ is the number of grain boundaries of G, α_i is the angular defect at the ith quadruple junction point, $f_0(G)$ is the number of quadruple junction points of G, and $\chi(\partial G)$ is the Euler characteristic of the boundary of G. The standard version of the Gauss–Bonnet theorem applies to an oriented surface F_i [102]:

$$\int_{F_i} K \,\mathrm{d}A + \int_{\partial F_i} \kappa_g \,\mathrm{d}s + \sum_{j=1}^{f_0(F_i)} (\pi - \gamma_j) = 2\pi\chi(F_i) \tag{7.12}$$

where γ_j is the interior angle at the jth corner along ∂F_i. As suggested by the notation, let F_i be the ith grain boundary of G. Since every grain boundary is homeomorphic to a disk, $\chi(F_i) = 1$ and summing Eq. 7.12 over the grain boundaries of G gives:

$$\int_{\partial G} K \,\mathrm{d}A + \sum_{i=1}^{f_2(G)} \int_{\partial F_i} \kappa_g \,\mathrm{d}s - \sum_{i=1}^{f_2(G)} \sum_{j=1}^{f_0(F_i)} \gamma_j = 2\pi f_2(G) - \pi f_{0,2}(G)$$

where $f_{0,2}(G)$ is the number of distinct pairs of adjacent quadruple junction points and grain boundaries. Adding $2\pi f_0(G)$ to each side of the above equation, using the identity $f_{0,2}(G) = f_{1,2}(G) = 2f_1(G)$, and applying the definition $\chi(\partial G) = f_0(G) - f_1(G) + f_2(G)$ for the Euler characteristic of a surface gives

$$\int_{\partial G} K \,\mathrm{d}A + \sum_{i=1}^{f_2(G)} \int_{\partial F_i} \kappa_g \,\mathrm{d}s + \left(2\pi f_0(G) - \sum_{i=1}^{f_2(G)} \sum_{j=1}^{f_0(F_i)} \gamma_j\right) = 2\pi\chi(\partial G).$$

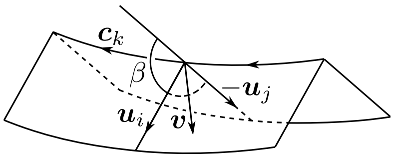


Figure 7.12. The respective arrangement of the curve $\boldsymbol{c}_k(s)$, the unit vectors $\boldsymbol{u}_i$, $\boldsymbol{u}_j$, and $\boldsymbol{v}$, and the angle β at a point along E_k.

Since 2π minus the internal angles around the ith quadruple junction point of G is the angular defect α_i at that point, this reduces to Eq. 7.11 above.

The next objective is to simplify Eq. 7.11 by means of the initial assumptions. Initially observe that $\chi(\partial G) = 2$ since G is homeomorphic to a ball, and that the angular defect at every quadruple junction point is $\alpha = 2\pi - 3\cos^{-1}(-1/3)$; this allows Eq. 7.11 to be reduced to:

$$\int_{\partial G} K \,\mathrm{d}A + \sum_{i=1}^{f_2(G)} \sum_{j=1}^{f_1(F_i)} \int_{E_j} \kappa_g \,\mathrm{d}s + \alpha f_0(G) = 4\pi$$

where E_j is the jth triple junction line and $f_1(F_i)$ is the number of triple junction lines of F_i. Now consider the summation of the integrals of the geodesic curvature over the triple junction lines of G. Every triple junction line is integrated over twice, once for each of the adjoining grain boundaries. Suppose that all of the grain boundaries have outward-pointing unit normal vector fields, that the grain boundaries F_i and F_j intersect on the kth triple junction line E_k of G, and that $\boldsymbol{c}_k(s)$ is the curve parameterized by arc length that travels along E_k in the positive orientation on F_i and the negative orientation on F_j. Let $\boldsymbol{\nu}_i(s)$ and $\boldsymbol{\nu}_j(s)$ be the restrictions of the unit normal vector fields of F_i and F_j to points on $\boldsymbol{c}_k(s)$, and $\boldsymbol{u}_i(s) = \boldsymbol{\nu}_i(s) \times \boldsymbol{c}'_k(s)$ and $\boldsymbol{u}_j(s) = \boldsymbol{\nu}_j(s) \times \boldsymbol{c}'_k(s)$ be the tangent normals of $\boldsymbol{c}_k(s)$ on F_i and F_j as indicated in Fig. 7.12, where the prime indicates differentiation with respect to arclength. Finally, write the geodesic curvatures κ_g of $\boldsymbol{c}_k(s)$ on F_i and F_j as $\boldsymbol{c}''_k(s) \cdot \boldsymbol{u}_i(s)$ and $\boldsymbol{c}''_k(s) \cdot \boldsymbol{u}_j(s)$. This allows the middle term on the left of the above equation to be written as:

$$\begin{aligned}\sum_{i=1}^{f_2(G)} \sum_{j=1}^{f_1(F_i)} \int_{E_j} \kappa_g \,\mathrm{d}s &= \sum_{k=1}^{f_1(G)} \left[\int_0^{s_k} \boldsymbol{c}''_k(s) \cdot \boldsymbol{u}_i(s) \,\mathrm{d}s - \int_0^{s_k} \boldsymbol{c}''_k(s) \cdot \boldsymbol{u}_j(s) \,\mathrm{d}s \right] \\ &= \sum_{k=1}^{f_1(G)} \int_0^{s_k} \kappa \boldsymbol{e}_2 \cdot (\boldsymbol{u}_i(s) - \boldsymbol{u}_j(s)) \,\mathrm{d}s\end{aligned}$$

where $s \in [0, s_k]$ and $\boldsymbol{c}_k''(s) = \kappa \boldsymbol{e}_2$, or the product of the curvature κ and the second Frenet vector $\boldsymbol{e}_2$ of $\boldsymbol{c}_k(s)$. Since $\boldsymbol{u}_i(s)$ and $\boldsymbol{u}_j(s)$ are unit vectors, $\boldsymbol{u}_i(s) - \boldsymbol{u}_j(s)$ can be written as $2\sin(\beta/2)\boldsymbol{v}(s)$ where $\boldsymbol{u}_i(s) \cdot \boldsymbol{u}_j(s) = \cos\beta$ and $\boldsymbol{v}(s)$ is the unit vector pointing along $\boldsymbol{u}_i(s) - \boldsymbol{u}_j(s)$ as in Fig. 7.12. Since $\beta = \pi/3$, this simplifies further to $\boldsymbol{u}_i(s) - \boldsymbol{u}_j(s) = \boldsymbol{v}(s)$ and Eq. 7.11 reduces to:

$$\int_{\partial G} K \,\mathrm{d}A + \sum_{i=1}^{f_1(G)} \int_{E_i} \kappa \boldsymbol{e}_2 \cdot \boldsymbol{v} \,\mathrm{d}s + \alpha f_0(G) = 4\pi. \tag{7.13}$$

This is perhaps the extent of simplification that is possible for a single grain.

Equations 1 and 2 in the main text are found by summing Eq. 7.13 over all the grains in Ω. As described in the main text, the contribution of the second term on the left vanishes since every triple junction line is integrated over three times, once for each adjoining grain. $\kappa \boldsymbol{e}_2$ is an inherent quantity of the triple junction line that is the same for all three integrals, but the three $\boldsymbol{v}$ are all unit vectors in a plane with mutual angles of $2\pi/3$. This causes the sum of the three $\boldsymbol{v}$ to vanish identically for each triple junction line, leaving:

$$\sum_{i=1}^{f_3(\Omega)} \int_{\partial G_i} K \,\mathrm{d}A + \sum_{i=1}^{f_3(\Omega)} \alpha f_0(G_i) = 4\pi f_3(\Omega)$$

where $f_3(\Omega)$ is the number of grains of Ω. Dividing through by $f_3(\Omega)$ gives Eq. 1 of the main text. Alternatively, observing that every grain boundary is included two times in the first sum, that every quadruple junction point is included four times in the second sum, and dividing through by two gives Eq. 2 of the main text.

7.4.2 Other boundary conditions

If the grain structure Ω exists in a domain without periodic boundary conditions, then the elements of Ω intersecting the external surfaces need to be handled differently. The derivation of the counterparts of Eqs. 1 and 2 of the main text proceeds as above, starting with Eq. 7.11 for a grain G:

$$\begin{aligned}\sum_{i=1}^{\bar{f}_2(G)} \int_{\bar{F}_i} K \,\mathrm{d}A + \sum_{i=1}^{\bar{f}_2(G)} \sum_{j=1}^{\bar{f}_1(\bar{F}_i)} \int_{\bar{E}_j} \kappa_g \,\mathrm{d}s + \sum_{i=1}^{f_2(G)} \sum_{j=1}^{\bar{f}_1(F_i)} \int_{\bar{E}_j} \kappa_g \,\mathrm{d}s + \sum_{i=1}^{\bar{f}_0(G)} \alpha_i \\ + \sum_{i=1}^{f_2(G)} \int_{F_i} K \,\mathrm{d}A + \sum_{i=1}^{f_2(G)} \sum_{j=1}^{f_1(F_i)} \int_{E_j} \kappa_g \,\mathrm{d}s + \sum_{i=1}^{f_0(G)} \alpha_i = 2\pi\chi(\partial G)\end{aligned}$$

where quantities relating to grain structure elements on external surfaces are indicated by overlines (e.g., the second term on the left concerns triple junction lines on the external surface that bound grain boundaries on the external surface, whereas the third term on the left concerns triple junction lines on the external surface that bound internal grain boundaries). Using that $\chi(\partial G) = 2$ and that the angular defect at every internal quadruple junction point is α, repeating the steps involving integrals over triple junction lines, and summing over all the grains of Ω gives:

$$\sum_{i=1}^{f_3(\Omega)} \sum_{j=1}^{\bar{f}_2(G_i)} \int_{\bar{F}_j} K \, \mathrm{d}A + \sum_{i=1}^{f_3(\Omega)} \sum_{j=1}^{\bar{f}_1(G_i)} \int_{\bar{E}_j} \kappa \boldsymbol{e}_2 \cdot [2\cos(\lambda/2)\boldsymbol{v}] \, \mathrm{d}s + \sum_{i=1}^{f_3(\Omega)} \sum_{j=1}^{\bar{f}_0(G_i)} \alpha_j$$
$$+ \sum_{i=1}^{f_3(\Omega)} \sum_{j=1}^{f_2(G_i)} \int_{F_j} K \, \mathrm{d}A + \alpha \sum_{i=1}^{f_3(\Omega)} f_0(G_i) = 4\pi f_3(\Omega) \tag{7.14}$$

where $\cos(\lambda/2) = \sin(\beta/2)$ and λ is the interior angle along the triple junction line.

Dividing through by $f_3(\Omega)$ gives:

$$\Big\langle \sum_{i=1}^{\bar{f}_2(G)} \int_{\bar{F}_i} K \, \mathrm{d}A + \sum_{i=1}^{\bar{f}_1(G)} \int_{\bar{E}_i} \kappa \boldsymbol{e}_2 \cdot [2\cos(\lambda/2)\boldsymbol{v}] \, \mathrm{d}s + \sum_{i=1}^{\bar{f}_0(G)} \alpha_i \Big\rangle$$
$$+ \Big\langle \int_{\partial G} K \, \mathrm{d}A \Big\rangle + \alpha \langle f_0(G) \rangle = 4\pi$$

which is the counterpart to Eq. 1 of the main text and emphasizes the properties of individual grains. This version groups the contributions of grain structure elements on external surfaces in the first average over all grains of Ω. Since $f_3(\Omega)$ should increase as the volume whereas the number of terms in the average should increase as the surface area, the influence of the first average should decrease with decreasing surface area to volume ratio of Ω. Alternatively, starting with Eq. 7.14 and observing that every internal grain boundary is included two times in the fourth term, that every quadruple junction point is included four times in the fifth term, and dividing through by two gives:

$$\frac{1}{2}\Big\{ \sum_{i=1}^{\bar{f}_2(\Omega)} \int_{\bar{F}_i} K \, \mathrm{d}A + \sum_{i=1}^{\bar{f}_1(\Omega)} \int_{\bar{E}_i} \kappa \boldsymbol{e}_2 \cdot \Big[2 \sum_{j=1}^{\bar{f}_{1,3}(\bar{E}_i)} \cos(\lambda_j/2)\boldsymbol{v}_j \Big] \, \mathrm{d}s + \sum_{i=1}^{\bar{f}_0(\Omega)} \sum_{j=1}^{\bar{f}_{0,3}(\bar{V}_i)} \alpha_j \Big\}$$
$$+ \sum_{i=1}^{f_2(\Omega)} \int_{F_i} K \, \mathrm{d}A + 2\alpha f_0(\Omega) = 2\pi f_3(\Omega)$$

where $\overline{f}_{1,3}(\overline{E}_i)$ is the number of grains adjacent to the ith triple junction line $\overline{E}_i$ on the external surface and $\overline{f}_{0,3}(\overline{V}_i)$ is the number of grains adjacent to the ith quadruple junction point $\overline{V}_i$ on the external surface. This is the counterpart to Eq. 2 of the main text. This is as far as the equations can be developed without specifying the shape of the domain of Ω and the nature of the boundary conditions.

REFERENCES

[1] Erdem Eren and Jeremy K. Mason. Topological transitions during grain growth on a finite element mesh. *Phys. Rev. Materials*, 5:103802, Oct 2021.

[2] National Research Council. *Integrated Computational Materials Engineering: A Transformational Discipline for Improved Competitiveness and National Security*. The National Academies Press, Washington, DC, 2008.

[3] Stefan Zaefferer. Application of orientation microscopy in SEM and TEM for the study of texture formation during recrystallisation processes. In *Textures of Materials - ICOTOM 14*, volume 495 of *Materials Science Forum*, page 312. Trans Tech Publications, 7 2005.

[4] S. F. Li and R. M. Suter. Adaptive reconstruction method for three-dimensional orientation imaging. *Journal of Applied Crystallography*, 46(2):512524, Apr 2013.

[5] S.F. Li, J.K. Mason, J. Lind, and M. Kumar. Quadruple nodes and grain boundary connectivity in three dimensions. *Acta Materialia*, 64(0):220 – 230, 2014.

[6] A. Morawiec. Method to calculate the grain boundary energy distribution over the space of macroscopic boundary parameters from the geometry of triple junctions. *Acta Materialia*, 48(13):35253532, 2000.

[7] David M. Saylor, Adam Morawiec, and Gregory S. Rohrer. Distribution of grain boundaries in magnesia as a function of five macroscopic parameters. *Acta Materialia*, 51(13):3663 – 3674, 2003.

[8] David M. Saylor, Adam Morawiec, and Gregory S. Rohrer. The relative free energies of grain boundaries in magnesia as a function of five macroscopic parameters. *Acta Materialia*, 51(13):3675 – 3686, 2003.

[9] V.V. Bulatov, B.W. Reed, and M. Kumar. Grain boundary energy function for FCC metals. *Acta Materialia*, 65:161175, 2014.

[10] Brandon Runnels, Irene J. Beyerlein, Sergio Conti, and Michael Ortiz. A relaxation method for the energy and morphology of grain boundaries and interfaces. *Journal of the Mechanics and Physics of Solids*, 94:388408, 2016.

[11] D.J. Srolovitz, G.S. Grest, and M.P. Anderson. Computer simulation of recrystallization - I. Homogeneous nucleation and growth. *Acta Metallurgica*, 34(9):1833 – 1845, 1986.

[12] Elizabeth A. Holm, James A. Glazier, David J. Srolovitz, and Gary S. Grest. Effects of lattice anisotropy and temperature on domain growth in the two-dimensional Potts model. *Phys. Rev. A*, 43:2662–2668, Mar 1991.

[13] Dierk Raabe. Cellular automata in materials science with particular reference to recrystallization simulation. *Annual Review of Materials Research*, 32(1):53–76, 2002.

[14] H.L. Ding, Y.Z. He, L.F. Liu, and W.J. Ding. Cellular automata simulation of grain growth in three dimensions based on the lowest-energy principle. *Journal of Crystal Growth*, 293(2):489 – 497, 2006.

[15] K.G.F. Janssens. An introductory review of cellular automata modeling of moving grain boundaries in polycrystalline materials. *Mathematics and Computers in Simulation*, 80(7):1361 – 1381, 2010. Multiscale modeling of moving interfaces in materials.

[16] J.K. Mason, J. Lind, S.F. Li, B.W. Reed, and M. Kumar. Kinetics and anisotropy of the monte carlo model of grain growth. *Acta Materialia*, 82:155 – 166, 2015.

[17] J.K. Mason. Grain boundary energy and curvature in monte carlo and cellular automata simulations of grain boundary motion. *Acta Materialia*, 94:162 – 171, 2015.

[18] Liangzhe Zhang, Anthony D. Rollett, Timothy Bartel, Di Wu, and Mark T. Lusk. A calibrated monte carlo approach to quantify the impacts of misorientation and

different driving forces on texture development. *Acta Materialia*, 60(3):1201 – 1210, 2012.

[19] I. Steinbach and F. Pezzolla. A generalized field method for multiphase transformations using interface fields. *Physica D: Nonlinear Phenomena*, 134(4):385 – 393, 1999.

[20] Nele Moelans, Bart Blanpain, and Patrick Wollants. An introduction to phase-field modeling of microstructure evolution. *Calphad*, 32(2):268 – 294, 2008.

[21] J Gruber, N Ma, Y Wang, A D Rollett, and G S Rohrer. Sparse data structure and algorithm for the phase field method. *Modelling and Simulation in Materials Science and Engineering*, 14(7):1189, 2006.

[22] Srikanth Vedantam and B. S. V. Patnaik. Efficient numerical algorithm for multiphase field simulations. *Phys. Rev. E*, 73:016703, Jan 2006.

[23] Eisuke Miyoshi, Tomohiro Takaki, Munekazu Ohno, Yasushi Shibuta, Shinji Sakane, Takashi Shimokawabe, and Takayuki Aoki. Ultra-large-scale phase-field simulation study of ideal grain growth. *npj Computational Materials*, 3(1):25, 2017.

[24] M.R. Dorr, J.-L. Fattebert, M.E. Wickett, J.F. Belak, and P.E.A. Turchi. A numerical algorithm for the solution of a phase-field model of polycrystalline materials. *Journal of Computational Physics*, 229(3):626 – 641, 2010.

[25] Y. Jin, N. Bozzolo, A.D. Rollett, and M. Bernacki. 2D finite element modeling of misorientation dependent anisotropic grain growth in polycrystalline materials: Level set versus multi-phase-field method. *Computational Materials Science*, 104: 108123, 2015.

[26] Josep Gras Ribot, Vinamra Agrawal, and Brandon Runnels. A new approach for phase field modeling of grain boundaries with strongly nonconvex energy. *Modelling and Simulation in Materials Science and Engineering*, 27(8):084007, Oct 2019.

[27] Kyozi Kawasaki, Tatsuzo Nagai, and Katsuya Nakashima. Vertex models for two-dimensional grain growth. *Philosophical Magazine B*, 60(3):399–421, 1989.

[28] T. Nagai, S. Ohta, K. Kawasaki, and T. Okuzono. Computer simulation of cellular pattern growth in two and three dimensions. *Phase Transitions*, 28(1-4):177–211, 1990.

[29] F. Roters, P. Eisenlohr, L. Hantcherli, D.D. Tjahjanto, T.R. Bieler, and D. Raabe. Overview of constitutive laws, kinematics, homogenization and multiscale methods in crystal plasticity finite-element modeling: Theory, experiments, applications. *Acta Materialia*, 58(4):1152 – 1211, 2010.

[30] R. Logé, M. Bernacki, H. Resk, L. Delannay, H. Digonnet, Y. Chastel, and T. Coupez. Linking plastic deformation to recrystallization in metals using digital microstructures. *Philosophical Magazine*, 88(30-32):3691–3712, 2008.

[31] Michael R. Tonks, Derek Gaston, Paul C. Millett, David Andrs, and Paul Talbot. An object-oriented finite element framework for multiphysics phase field simulations. *Computational Materials Science*, 51(1):20 – 29, 2012.

[32] Dierk Raabe and Richard C Becker. Coupling of a crystal plasticity finite-element model with a probabilistic cellular automaton for simulating primary static recrystallization in aluminium. *Modelling and Simulation in Materials Science and Engineering*, 8(4):445, 2000.

[33] Andrew Kuprat. Modeling microstructure evolution using gradient-weighted moving finite elements. *SIAM Journal on Scientific Computing*, 22(2):535560, 2000.

[34] Jason Gruber, Denise C. George, Andrew P. Kuprat, Gregory S. Rohrer, and Anthony D. Rollett. Effect of anisotropic grain boundary properties on grain boundary plane distributions during grain growth. *Scripta Materialia*, 53(3):351–355, 2005.

[35] M Syha and D Weygand. A generalized vertex dynamics model for grain growth in three dimensions. *Modelling and Simulation in Materials Science and Engineering*, 18(1):015010, 2010.

[36] Emanuel A. Lazar, Jeremy K. Mason, Robert D. MacPherson, and David J., Srolovitz. A more accurate three-dimensional grain growth algorithm. *Acta Materialia*, 59(17): 68376847, 2011.

[37] Robert D. MacPherson and David J., Srolovitz. The von Neumann relation generalized to coarsening of three-dimensional microstructures. *Nature*, 466:10531055, 2007.

[38] Joseph C Tucker, Albert R Cerrone III, Anthony R Ingraffea, and Anthony D Rollett. Crystal plasticity finite element analysis for rené88dt statistical volume element generation. *Modelling and Simulation in Materials Science and Engineering*, 23(3): 035003, 2015.

[39] Glen Hansen and Gopan Perumpilly. RPI SCOREC. `https://github.com/SCOREC`, Sep. 2021.

[40] R. Quey, P.R. Dawson, and F. Barbe. Large-scale 3d random polycrystals for the finite element method: Generation, meshing and remeshing. *Computer Methods in Applied Mechanics and Engineering*, 200(17-20):1729 – 1745, 2011.

[41] Michael A. Groeber and Michael A. Jackson. Dream.3d: A digital representation environment for the analysis of microstructure in 3d. *Integrating Materials and Manufacturing Innovation*, 3(1):56–72, 2014.

[42] Gene M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities, reprinted from the afips conference proceedings, vol. 30 (atlantic city, n.j., apr. 1820), afips press, reston, va., 1967, pp. 483485, when dr. amdahl was at international business machines corporation, sunnyvale, california. *IEEE Solid-State Circuits Society Newsletter*, 12(3):19–20, 2007.

[43] S. Seol, C. W. Smith, D. A. Ibanez, and M. S. Shephard. A parallel unstructured mesh infrastructure. In *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, pages 1124–1132, Los Alamitos, CA, USA, Nov 2012. IEEE Computer Society.

[44] Jeremy K. Mason. Stability and motion of arbitrary grain boundary junctions. *Acta Materialia*, 125:286295, 2017.

[45] Erdem Eren and Jeremy K. Mason. VDlib. https://github.com/erdemeren/VDlib, 2021.

[46] Erdem Eren and Jeremy K. Mason. VDtrials. https://github.com/erdemeren/VDtrials, 2021.

[47] Keith Paton. An algorithm for finding a fundamental set of cycles of a graph. *Commun. ACM*, 12(9):514–518, September 1969.

[48] Norman E. Gibbs. A cycle generation algorithm for finite undirected linear graphs. *J. ACM*, 16(4):564–568, October 1969.

[49] J von Neumann. Written discussion. In *Metal Interfaces*, pages 108–110, Cleveland, Ohio, 1952. American Society for Metals.

[50] W W Mullins. Two-dimensional motion of idealized grain boundaries. *Journal of Applied Physics*, 27:900–904, March 1956.

[51] Erdem Eren, Brandon Runnels, and Jeremy Mason. Evolution of interfaces, triple points, and quadruple points using discrete interface equations of motion. *ArXiv e-prints*, 2203.03167, 2022.

[52] Gnter Gottstein, Vera Sursaeva, and Lasar S. Shvindlerman. The effect of triple junctions on grain boundary motion and grain microstructure evolution. *Interface Science*, 7(3):273283, 1999.

[53] J.E. Burke and D. Turnbull. Recrystallization and grain growth. *Progress in Metal Physics*, 3:220–292, 1952.

[54] Daniel A. Ibanez, E. Seegyoung Seol, Cameron W. Smith, and Mark S. Shephard. Pumi: Parallel unstructured mesh infrastructure. *ACM Trans. Math. Softw.*, 42(3), May 2016.

[55] V.N. Parthasarathy, C.M. Graichen, and A.F. Hathaway. A comparison of tetrahedron quality measures. *Finite Elements in Analysis and Design*, 15(3):255261, 1994.

[56] David A. Field. Qualitative measures for initial meshes. *International Journal for Numerical Methods in Engineering*, 47(4):887906, 2000.

[57] Xiangrong Li, Mark S. Shephard, and Mark W. Beall. 3d anisotropic mesh adaptation by mesh modification. *Computer Methods in Applied Mechanics and Engineering*, 194(48):4915–4950, 2005.

[58] Nele Moelans, Bart Blanpain, and Patrick Wollants. Quantitative analysis of grain boundary properties in a generalized phase field model for grain growth in anisotropic systems. *Physical Review B*, 78(2):024113, 2008.

[59] Nele Moelans, Bart Blanpain, and Patrick Wollants. Quantitative phase-field approach for simulating grain growth in anisotropic systems with arbitrary inclination and misorientation dependence. *Physical review letters*, 101(2):025502, 2008.

[60] Brandon Runnels, Vinamra Agrawal, Weiqun Zhang, and Ann Almgren. Massively parallel finite difference elasticity using block-structured adaptive mesh refinement with a geometric multigrid solver. *Journal of Computational Physics*, 427:110065, 2021.

[61] Weiqun Zhang, Ann Almgren, Vince Beckner, John Bell, Johannes Blaschke, Cy Chan, Marcus Day, Brian Friesen, Kevin Gott, Daniel Graves, et al. AM-

ReX: a framework for block-structured adaptive mesh refinement. *Journal of Open Source Software*, 4(37), 2019.

[62] Kenneth A. Brakke. *The Motion of a Surface by Its Mean Curvature.* Princeton University Press and University of Tokyo Press, 1987.

[63] R.I. Saye and J.A. Sethian. Analysis and applications of the Voronoi implicit interface method. *Journal of Computational Physics*, 231(18):6051 – 6085, 2012.

[64] S. Florez, M. Shakoor, T. Toulorge, and M. Bernacki. A new finite element strategy to simulate microstructural evolutions. *Computational Materials Science*, 172:109335, 2020.

[65] Julien Fausty, Nathalie Bozzolo, Daniel Pino Muñoz, and Marc Bernacki. A novel level-set finite element formulation for grain growth with heterogeneous grain boundary energies. *Materials & Design*, 160:578–590, 2018.

[66] L.A. Barrales Mora, V. Mohles, L.S. Shvindlerman, and G. Gottstein. Effect of a finite quadruple junction mobility on grain microstructure evolution: Theory and simulation. *Acta Materialia*, 56(5):1151 – 1164, 2008.

[67] Conyers Herring. Surface tension as a motivation for sintering. In Walter E. Kingston, editor, *The Physics of Powder Metallurgy*, page 143179. McGraw-Hill, 1951.

[68] Conyers Herring. The use of classical macroscopic concepts in surface energy problems. In Robert Gomer and Cyril Stanley Smith, editors, *Structure and Properties of Solid Surfaces*, page 581. University of Chicago Press, 1953.

[69] David L. Olmsted, Stephen M. Foiles, and Elizabeth A. Holm. Survey of computed grain boundary properties in face-centered cubic metals: I. Grain boundary energy. *Acta Materialia*, 57(13):3694–3703, 2009.

[70] E. Eren and J. K. Mason. Constant of motion for ideal grain growth in three dimensions. *Phys. Rev. B*, 104:L140103, Oct 2021.

[71] David Turnbull. Theory of grain boundary migration rates. *Journal of Metals*, 3(8): 661–665, 1951.

[72] P S Laplace. *Supplment au Dixime Livre du Trait de Mcanique Cleste: Sur laction Capillaire*. De LImprimerie de Crapelet, Paris, France, 1805.

[73] Joseph Antoine Ferdinand Plateau. *Statique expérimentale et théorique des liquides soumis aux seules forces moléculaires*, volume 2. Gauthier-Villars, 1873.

[74] Jean E Taylor. The structure of singularities in soap-bubble-like and soap-film-like minimal surfaces. *Annals of Mathematics*, pages 489–539, 1976.

[75] Cyril Stanley Smith. Some elementary principles of polycrystalline microstructure. *Metallurgical Reviews*, 9(1):148, 1964.

[76] John W Cahn. Significance of average mean curvature and its determination by quantitative metallography. *Transactions of the Metallurgical Society of AIME*, 239 (5):610, 1967.

[77] W. W. Mullins. The statistical self-similarity hypothesis in grain growth and particle coarsening. *Journal of Applied Physics*, 59(4):1341–1349, 1986.

[78] HV Atkinson. Overview no. 65: Theories of normal grain growth in pure single phase systems. *Acta Metallurgica*, 36(3):469–491, 1988.

[79] Jeremy K. Mason, Emanuel A. Lazar, Robert D. MacPherson, and David J. Srolovitz. Geometric and topological properties of the canonical grain-growth microstructure. *Phys. Rev. E*, 92:063308, Dec 2015.

[80] M Hillert. On the theory of normal and abnormal grain growth. *Acta Metallurgica*, 13(3):227 – 238, 1965.

[81] W Fayad, CV Thompson, and HJ Frost. Steady-state grain-size distributions resulting from grain growth in two dimensions. *Scripta materialia*, 40(10):1199–1204, 1999.

[82] PR Rios and ME Glicksman. Polyhedral model for self-similar grain growth. *Acta Materialia*, 56(5):1165–1171, 2008.

[83] Michael D Uchic, Michael Groeber, Robert Wheeler IV, Frank Scheltens, and Dennis M Dimiduk. Augmenting the 3D characterization capability of the dual beam FIB-SEM. *Microscopy and Microanalysis*, 10(S02):1136–1137, 2004.

[84] J Konrad, S Zaefferer, and D Raabe. Investigation of orientation gradients around a hard laves particle in a warm-rolled Fe3Al-based alloy using a 3D EBSD-FIB technique. *Acta Materialia*, 54(5):1369–1380, 2006.

[85] Henning Friis Poulsen. *Three-dimensional X-ray diffraction microscopy: mapping polycrystals and their dynamics*, volume 205. Springer Science & Business Media, 2004.

[86] Rob Kusner. The number of faces in a minimal foam. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):683–686, 1992.

[87] ME Glicksman. Analysis of 3-D network structures. *Philosophical Magazine*, 85(1): 3–31, 2005.

[88] RT DeHoff. The spherical image concept applied to grain structures. *Acta metallurgica et materialia*, 42(8):2633–2643, 1994.

[89] Dov Levine. Stability of monodisperse dry foams. *Philosophical Magazine Letters*, 74(4):303–307, 1996.

[90] Douglas A. Reinelt and Andrew M. Kraynik. Large elastic deformations of three-dimensional foams and highly concentrated emulsions. *Journal of Colloid and Interface Science*, 159(2):460–470, 1993.

[91] Jin Zhang, Wolfgang Ludwig, Yubin Zhang, Hans Henrik B. Srensen, David J. Rowenhorst, Akinori Yamanaka, Peter W. Voorhees, and Henning F. Poulsen. Grain boundary mobilities in polycrystals. *Acta Materialia*, 191:211–220, 2020.

[92] F. Ulomek and V. Mohles. Molecular dynamics simulations of grain boundary mobility in al, cu and γ-fe using a symmetrical driving force. *Modelling and Simulation in Materials Science and Engineering*, 22(5):055011, may 2014.

[93] F. Ulomek, C. J. O'Brien, S. M. Foiles, and V. Mohles. Energy conserving orientational force for determining grain boundary mobility. *Modelling and Simulation in Materials Science and Engineering*, 23(2):025007, jan 2015.

[94] Sutatch Ratanaphan, David L. Olmsted, Vasily V. Bulatov, Elizabeth A. Holm, Anthony D. Rollett, and Gregory S. Rohrer. Grain boundary energies in body-centered cubic metals. *Acta Materialia*, 88:346–354, 2015.

[95] Robert A. Carson, Steven R. Wopschall, and Jamie A. Bramwell. ExaConstit. [Computer Software] `https://doi.org/10.11578/dc.20191024.2`, Aug 2019. URL `https://github.com/LLNL/ExaConstit`.

[96] F. Roters, M. Diehl, P. Shanthraj, P. Eisenlohr, C. Reuber, S. L. Wong, T. Maiti, A. Ebrahimi, T. Hochrainer, H-O Fabritius, S. Nikolov, M. Friak, N. Fujita, N. Grilli, K. G. F. Janssens, N. Jia, P. J. J. Kok, D. Ma, F. Meier, E. Werner, M. Stricker, D. Weygand, and D. Raabe. Damask – the düsseldorf advanced material simulation kit for modelling multi-physics crystal plasticity, damage, and thermal phenomena from the single crystal up to the component scale. *Computational Materials Science*, 158:420–478, 2019.

[97] Paul R. Dawson and Donald E. Boyce. Fepx – finite element polycrystals: Theory, finite element formulation, numerical implementation and illustrative examples. 1504.03296, 2015.

[98] C.F. Gu, L.S. Toth, and M. Hoffman. Twinning effects in a polycrystalline magnesium alloy under cyclic deformation. *Acta Materialia*, 62:212 – 224, 2014.

[99] C. Can Aydner and M. Aya Telemez. Multiscale deformation heterogeneity in twinning magnesium investigated with in situ image correlation. *International Journal of Plasticity*, 56:203–218, 2014.

[100] Qiongyao He, Tianlin Huang, Linfei Shuai, Yubin Zhang, Guilin Wu, Xiaoxu Huang, and Dorte Juul Jensen. In-situ investigation of the evolution of annealing twins in high purity aluminium. *Scripta Materialia*, 153:68–72, 2018.

[101] H. V. Atkinson and S. Davies. Fundamental aspects of hot isostatic pressing: An overview. *Metallurgical and Materials Transactions A*, 31(12):2981–3000, 2000.

[102] Alfred Gray, Elsa Abbena, and Simon Salamon. *Modern differential geometry of curves and surfaces with Mathematica®*. Chapman and Hall/CRC, 2017.