Topological transitions during grain growth on a finite element mesh

Erdem Eren [*] and Jeremy K. Mason [†]

Department of Materials Science and Engineering, University of California at Davis, Davis, California 95616, USA

The topological transitions that occur to the grain boundary network during grain growth in a material with uniform grain boundary energies are believed to be known. The same is not true for more realistic materials, since more general grain boundary energies in principle allow many more viable 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. As a specific example, all possible transitions for a typical configuration with five grains around a junction point are enumerated, and some exceptional transitions are found to be energetically similar to the conventional ones even for a uniform boundary energy. A general discrete formulation to calculate grain boundary velocities is used to simulate grain growth for an example microstructure. The method is implemented as a C++ library based on SCOREC, an open source massively parallelizable library for finite element simulations with adaptive meshing.

I. INTRODUCTION

One of the overarching goals of integrated computational materials engineering (ICME) [1] 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 [2], and specifically three-dimensional X-ray diffraction microscopy (3DXRD) that enables nondestructive three-dimensional imaging of millimeter-sized samples [3,4]. 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 [5]. Saylor *et al.* subsequently used a related technique to estimate the GBE from EBSD analysis of the surface of aluminum samples [6,7]. While explicit functions for the grain boundary energy are not yet widely available (with a few exceptions [8,9]), this will likely change in the near future. When 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. The MC Potts [10,11] and CA [12–14] 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 [15,16], 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 [17]. 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 [16].

The phase field method is an implicit boundary approach that was initially developed to study phase transitions [18], and can be modified to include small deformations and mildly anisotropic interface energies [19]. 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 [20–22] and adaptive meshing [23] 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 [24]. 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

---
[*]ereren@ucdavis.edu
[†]jkylemason@ucdavis.edu

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 [25], but the modeling of finite mechanical deformation has still not been addressed.

Early front tracking methods had the advantage of concentrating computational resources just on the boundaries, and were often used to study mean curvature flow [26,27]. 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 [28]. 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 [29–31]. This does not resolve accuracy concerns though, since transferring the solution potentially introduces information loss and increases computational complexity.

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 [32]. 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.* [33]. 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 [32]. 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.* [4].

Using a surface mesh representation, Syha and Weygand studied the effects of an anisotropic grain boundary energy [34]. 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 [35,36]. 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 does not 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 [37]. 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 for 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 [32,34,35]. Second, topological transitions are achieved by merging mesh entities with one of the neighboring grains [32], by sequentially splitting points [34], or selecting from a restricted set of operations [35], 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.

There are four practical contributions described in this paper. 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 [38]. 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 [39,40] implements all these operations and interfaces with SCOREC [41,42], a massively parallel mesh management library with local adaptive remeshing.

The process of code development and the capabilities of VDLIB have already resulted in three scientific contributions. First, the process of finding all possible topological transitions for a canonical configuration of five grains around a junction
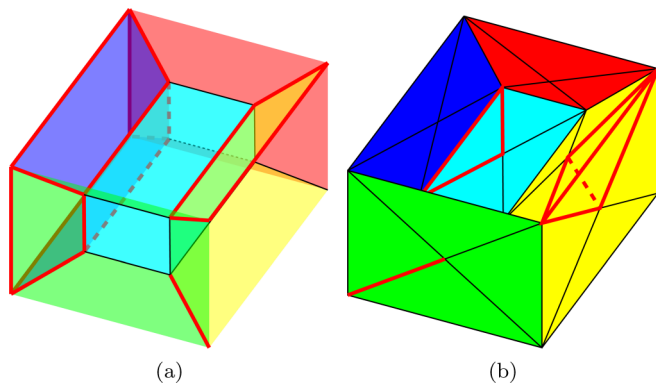
FIG. 1. (a) An example grain structure composed of seven grains. A central rectangular grain is surrounded by six grains, with examples of a volume, surface, and line outlined in red. (b) A finite element representation of this grain structure where tetrahedra, triangles, edges, and vertices are used to discretize volumes, surfaces, lines, and points. Examples of a tetrahedron, triangle, and edge are outlined in red.
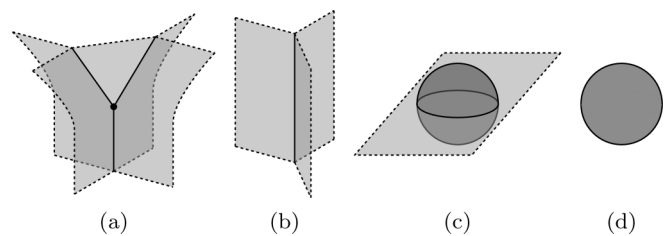

FIG. 2. Examples indicating adjacency rules. (a) A point should bound at least three lines. This point bounds three lines and four volumes, two on the left and right and two in front of and behind the page. (b) A line should bound at least three surfaces. (c) A surface separating a top and a bottom volume 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.

point revealed two possibilities that are likely not mentioned anywhere in the literature, and that are named mixed removal and digon insertion type-II in Sec. VII. Second, the spurious stratum insertion discussed in Sec. IV C is established as one of the essential topological transitions for the progress of a grain growth simulation. Third, Sec. VI generalizes the MacPherson-Srolovitz relation [36] 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.

II. MICROSTRUCTURE REPRESENTATION

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 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. 1(a). 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(a)], a line at least three surfaces [Fig. 2(b)], 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(c)], or a ball to be embedded in the interior of a volume [Fig. 2(d)].

III. 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. 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, with several specifics of the algorithm given in Sec. II of the Supplemental Material (SM) [43].
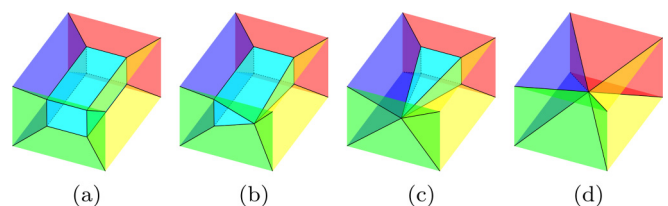

FIG. 3. The cases of collapse shown on the rectangular prism example. (a) The initial microstructure. (b) Line collapse. (c) Surface collapse. (d) Volume collapse.
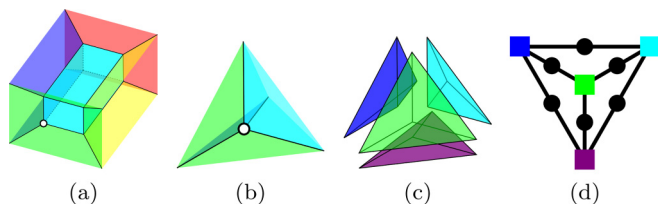
FIG. 4. (a) Consider the point at the bottom left corner of the central volume. (b) The neighborhood of the point shows the relationships with the surrounding surfaces and volumes. (c) The volumes in an exploded view. (d) The adjacency graph showing the volumes as squares and the surfaces as disks. In this figure, volumes and squares are the same color.

B. 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. 4 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 specifics given in Sec. VIII of the SM.

1. Line insertions

Every possible line insertion corresponds to a circuit on the associated adjacency graph, with an example shown in Fig. 5. 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. 5(a)
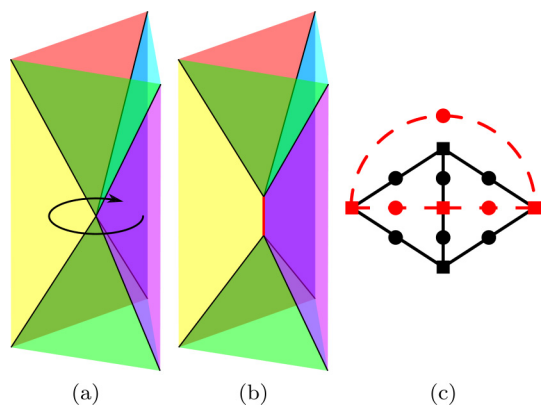

FIG. 5. A line insertion corresponds to a circuit on the adjacency graph. (a) A five grain configuration and a circuit going around the point. (b) Every surface punctured by the circuit is extended by adding the inserted line to their adjacency lists. (c) The adjacency graph around the point. Edges along the circuit are dashed.
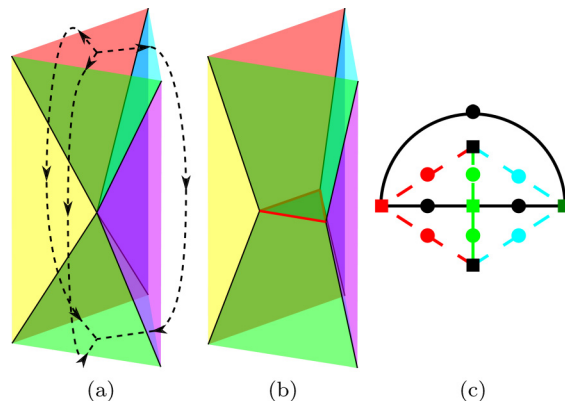

FIG. 6. A surface insertion corresponds to a set of paths on the adjacency graph. (a) A five grain configuration, showing a set of three nonintersecting paths connecting the disconnected (top and bottom) volumes. (b) A surface is inserted between the disconnected volumes with one bounding line for each 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).

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. 5(a) precisely corresponds to the circuit in Fig. 5(c), 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 [44,45]. 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. III of the SM.

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 volumes and ends on the other, as in Fig. 6(a). 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. 6(b). The set of all possible surface insertions can be found by constructing all possible sets of nonintersecting 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. IV of the SM.
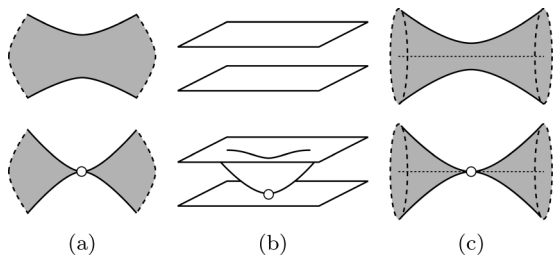
FIG. 7. Topological transitions not considered here. (a) Two lines bounding the same surface meet to form a new point. (b) Two bounding surfaces of a volume meet to form a new point. (c) The cross section of a cylindrical volume is reduced to a point.

C. 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. 7 does not occur. That is, the only allowed topological transitions are those for which the length of a line, the area 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. 7 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. 8. 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. II) 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. V of the SM. The construction of a small neighborhood necessarily involves the mesh, and will be considered further in Sec. IV C.

IV. 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 engineer-
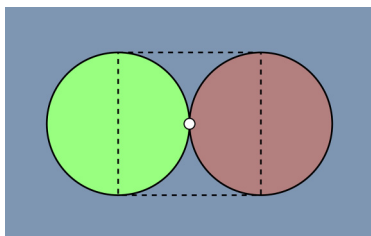

FIG. 8. A point connected to two spherical grains, and two grains in front of and behind the page. The neighborhood of the point is outlined by a dashed line. The surface in the page is represented twice in the neighborhood of the point.
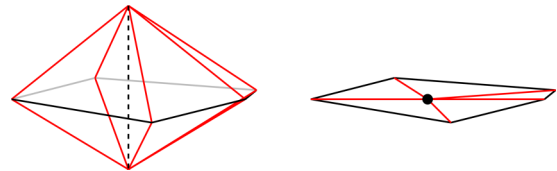

FIG. 9. Lens collapse operation. Left: The lens composed of tetrahedra and triangles bounded by the collapsing dashed edge. Right: The disk obtained by collapsing the lens.

ing [41,42]. 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. 1.

Applying the stratum collapse and insertion operations described in Sec. III 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_i^d more specifically the ith d-stratum.

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

On a simplicial mesh, an edge bounds a collection of tetrahedra and triangles forming a lens around that edge. As shown in Fig. 9, 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
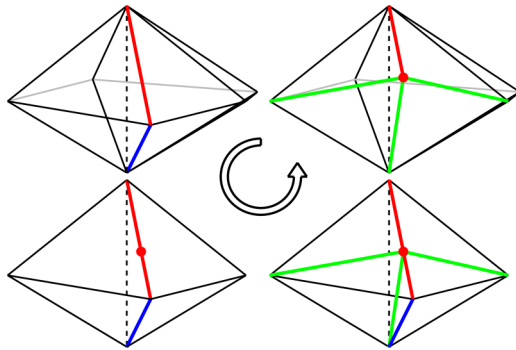
FIG. 10. Edge split operation during preconditioning. The thicker edges in red and blue belong to strata S_i^d and S_j^e, respectively. If S_i^d and S_j^e are not the same and one does not bound the other, collapse of the dashed vertical edge is not allowed. Splitting the red edge and all entities that 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.

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. 10 to one of the edges of the problematic couple resolves this situation. Second, it is possible that two d-dimensional entities could unintentionally merge. This could occur even if they do not belong to 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. 11(a). The edge split procedure can also resolve this by isolating the collapsing entity, as shown in Fig. 11(b). 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.
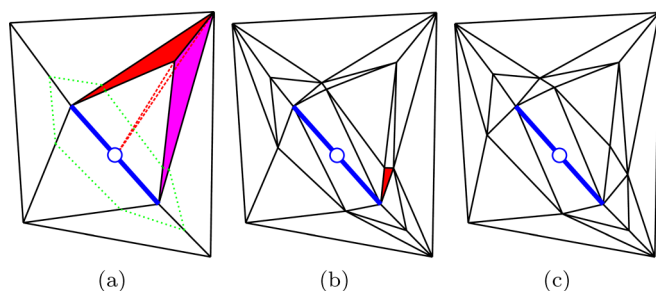

(a)   (b)   (c)

FIG. 11. 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.

The three-dimensional equivalent of the preconditioning operation in Fig. 11 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. 11(c) and described in Sec. VI of the SM 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.

B. Stratum insertion

As described in Sec. III B, 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 disk that can be expanded. The expansion forms a new vertex, a new edge, and a set of new triangles and tetrahedra corresponding to the disk 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 disk- or finlike 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 the associated bias in the numerical scheme for vertex velocities.

Stratum insertion requires expansion of a disk/fin, creation of triangles and tetrahedra with the same stratum memberships as the edges and triangles on the disk/fin, and creation
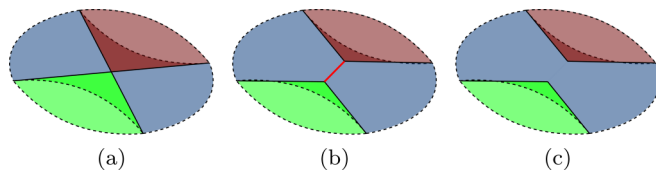
FIG. 12. Steps of a spurious line insertion. (a) A point, connected to the red and green volumes above and below and two volumes in front of and behind the page. (b) Insertion of the spurious line, adjacent to two surfaces both separating the same volumes. (c) The spurious line is removed and the two surfaces are merged.

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 disk associated with the circuit is used to create three disks, 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 disk entities belong to the same strata as in the initial disk. Two new S^1 edges are created to connect the S^1 vertex to the bounding S^0 vertices. The volume between the disks and around the new S^1 edges is filled by triangles and tetrahedra corresponding to edges and triangles on the disks. 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 triangle. 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. V.

C. 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. 12. 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 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. X of the SM.

V. 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. 13. 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
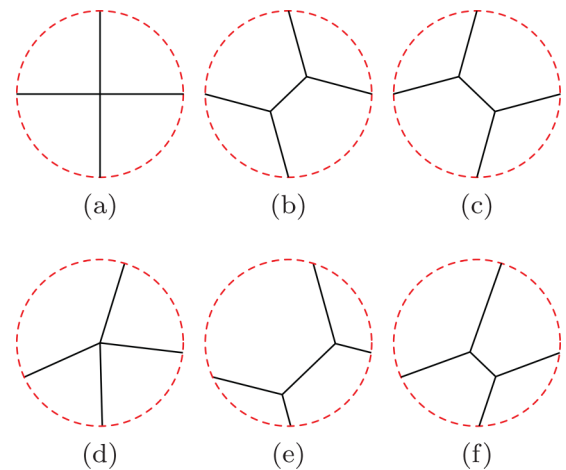

FIG. 13. The choice of insertion can change the overall trajectory of the system. (a) A two-dimensional degenerate configuration with four grains could transition to either (b) or (c) since they are energetically equivalent. For (d), (e) and (f) both lower the energy, but (e) more so.

relaxation algorithm, as shown in Fig. 14 for a digon insertion. The bold black lines in Fig. 14(a) 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. 14(b). 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. 14(c)] 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 Figs. 14(c) and 14(d) 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 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

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

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 [34]. The difference is that our approach instead compares all possible single stratum insertions at once using the energy dissipation rate criterion, presumably
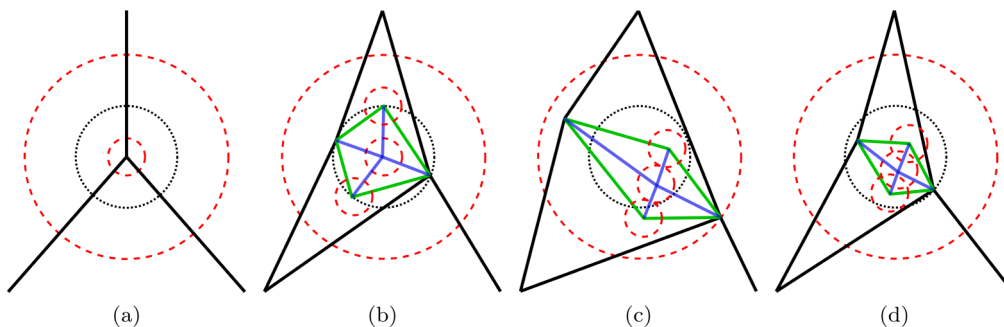
FIG. 14. The steps of mesh level insertion and reorientation for a digon insertion. (a) Fins of triangles along paths, shown in bold black. (b) Insertion of the new digon, where S^1 edges are shown as green lines and S^2 edges are shown as blue lines. (c) The vertices are allowed to move until one of the ending criteria is reached. (d) The digon is scaled to be within the projection sphere, and relaxation continues until the energies converge.

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.

VI. 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 [36], the three-dimensional extension of the von Neumann–Mullins relation [46,47]. 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 [36] governing the rates of change of volumes is given by

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

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 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 [35]. 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. (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. VII 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,

$$\frac{dV(D)}{dt} = -\mu\gamma[2\pi\mathcal{L}(D) - \mathcal{N}(D)], \tag{2}$$

$$\mathcal{N}(D) = \sum_j \beta_j l_j, \tag{3}$$

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. (2) reduces to Eq. (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.

VII. 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 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. 5(a). All possible insertions can be found by applying the circuit and path detection algorithms, and these are shown in Fig. 15 (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

$$S^1 \text{ insertions} \qquad\qquad S^2 \text{ insertions}$$

Volume removal   Petal removal type-I   Petal removal type-II   Mixed removal   Digon insertion type-I   Digon insertion type-II   Trigon insertion
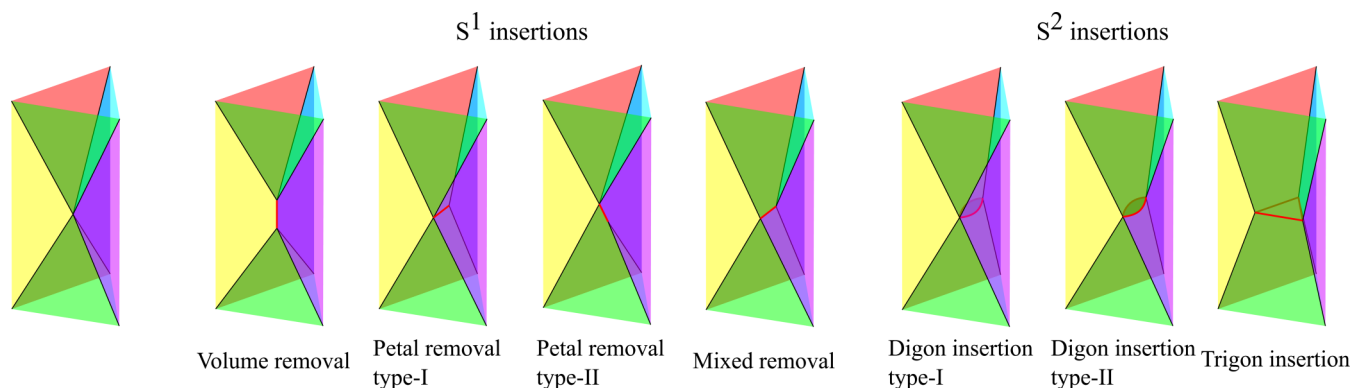
FIG. 15. All possible insertions for the canonical configuration, classified by symmetry groups. Observe that digon insertions can be obtained by decomposing circuits containing disconnected 3-stratum couples into two paths connecting the couples and using these to insert a 2-stratum. Digon insertion type-I is related to petal removal type-I and digon insertion type-II is related to mixed removal.

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. 15.

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 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. 15. 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. 16. The energy changes in Fig. 16(a) 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. 16(b), and correctly identify 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. 16. The bias depends on the selection of the ratio of the radii of the inner and outer spheres in Fig. 14. 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
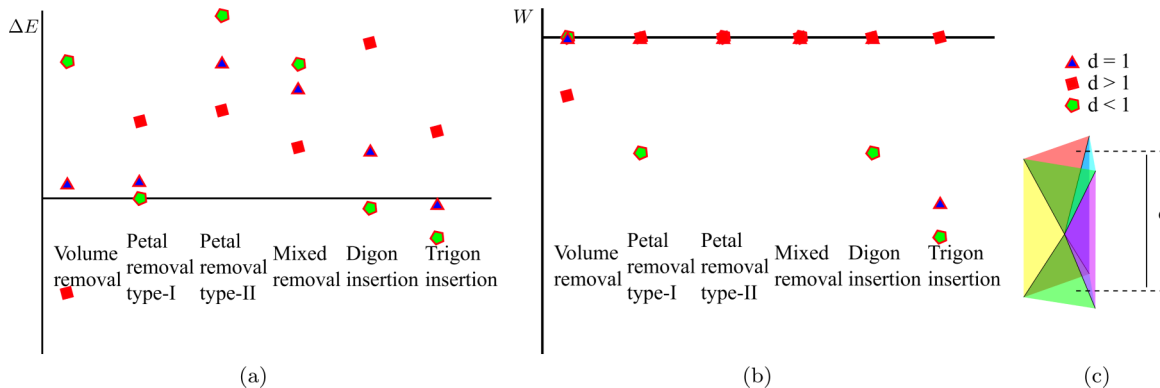

(a)   (b)   (c)

FIG. 16. (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.
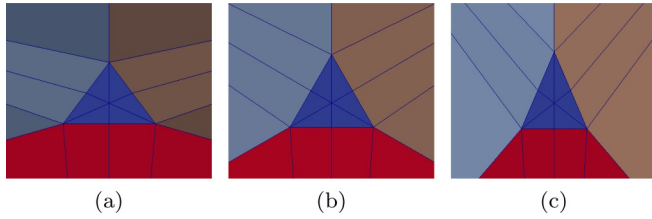
FIG. 17. The effect of orthogonal stretching on the trigon shape. (b) Starting configuration, where dihedral angles between surfaces separating the surrounding S^3 are equal. (a)–(c) After stretching (compressing) the configuration in the lateral direction, running the relaxation yields a laterally stretched (compressed) S^2.

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. 17, 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 [48]. 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. VII of the SM. 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 additionally required to contain at least two edges to provide sufficient degrees of freedom. The microstructure is evolved using equations of motion by Mason [38] 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 subiterations 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. 18, with the discretization of several grains visible in Fig. S8 of the SM. 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-
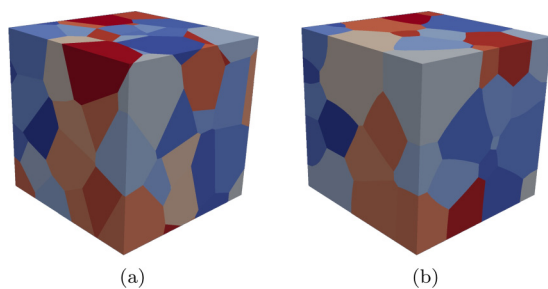

FIG. 18. Simulation of a microstructure composed of 100 grains under isotropic grain boundary energy. (a) Initial configuration. (b) The number of grains is about one half of the starting number.
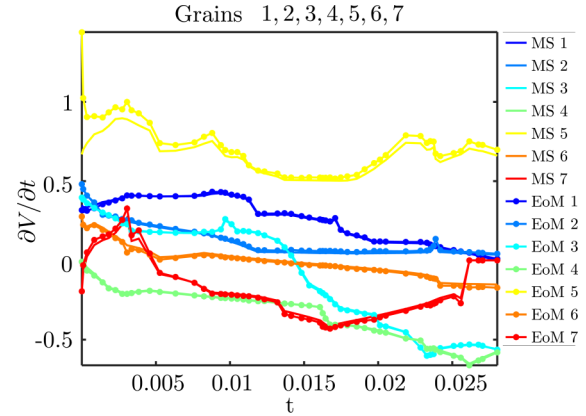

FIG. 19. The rates of volume change for example grains as calculated by the modified MacPherson-Srolovitz (MS) relation, and first-order approximation using the equations of motion (EoM).

7700HQ CPU at 2.80 GHz. More details of the computational resource use and scaling are provided in Sec. XII of the SM.

The modified MS relation in Sec. VI 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. 19. The initial discrepancy is mainly due to the deviation from the 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. 19) which decrease with time. Despite using linear elements and an explicit time integration scheme, there is overall good agreement with the MS relation.

VIII. 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 [38] 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.

[1] N. R. Council, *Integrated Computational Materials Engineering: A Transformational Discipline for Improved Competitiveness and National Security* (The National Academies, Washington, DC, 2008).

[2] S. Zaefferer, in *Textures of Materials—ICOTOM 14* (Trans Tech Publications, Zurich, 2005), Vol. 495 of *Materials Science Forum*, pp. 3–12.

[3] S. F. Li and R. M. Suter, J. Appl. Crystallogr. **46**, 512 (2013).

[4] S. Li, J. Mason, J. Lind, and M. Kumar, Acta Mater. **64**, 220 (2014).

[5] A. Morawiec, Acta Mater. **48**, 3525 (2000).

[6] D. M. Saylor, A. Morawiec, and G. S. Rohrer, Acta Mater. **51**, 3663 (2003).

[7] D. M. Saylor, A. Morawiec, and G. S. Rohrer, Acta Mater. **51**, 3675 (2003).

[8] V. Bulatov, B. Reed, and M. Kumar, Acta Mater. **65**, 161 (2014).

[9] B. Runnels, I. J. Beyerlein, S. Conti, and M. Ortiz, J. Mech. Phys. Solids **94**, 388 (2016).

[10] D. J. Srolovitz, G. S. Grest, and M. P. Anderson, Acta Metal. **34**, 1833 (1986).

[11] E. A. Holm, J. A. Glazier, D. J. Srolovitz, and G. S. Grest, Phys. Rev. A **43**, 2662 (1991).

[12] D. Raabe, Annu. Rev. Mater. Res. **32**, 53 (2002).

[13] H. L. Ding, Y. Z. He, L. F. Liu, and W. J. Ding, J. Cryst. Growth **293**, 489 (2006).

[14] K. G. F. Janssens, Math. Comput. Simul. **80**, 1361 (2010).

[15] J. K. Mason, J. Lind, S. F. Li, B. W. Reed, and M. Kumar, Acta Mater. **82**, 155 (2015).

[16] J. K. Mason, Acta Mater. **94**, 162 (2015).

[17] L. Zhang, A. D. Rollett, T. Bartel, D. Wu, and M. T. Lusk, Acta Mater **60**, 1201 (2012).

[18] I. Steinbach and F. Pezzolla, Physica D **134**, 385 (1999).

[19] N. Moelans, B. Blanpain, and P. Wollants, Calphad **32**, 268 (2008).

[20] J. Gruber, N. Ma, Y. Wang, A. D. Rollett, and G. S. Rohrer, Model. Simul. Mater. Sci. Eng. **14**, 1189 (2006).

[21] S. Vedantam and B. S. V. Patnaik, Phys. Rev. E **73**, 016703 (2006).

[22] E. Miyoshi, T. Takaki, M. Ohno, Y. Shibuta, S. Sakane, T. Shimokawabe, and T. Aoki, npj Comput. Mater. **3**, 25 (2017).

[23] M. R. Dorr, J.-L. Fattebert, M. E. Wickett, J. F. Belak, and P. E. A. Turchi, J. Comput. Phys. **229**, 626 (2010).

[24] Y. Jin, N. Bozzolo, A. D. Rollett, and M. Bernacki, Comput. Mater. Sci. **104**, 108 (2015).

[25] J. G. Ribot, V. Agrawal, and B. Runnels, Model. Simul. Mater. Sci. Eng. **27**, 084007 (2019).

[26] K. Kawasaki, T. Nagai, and K. Nakashima, Philos. Mag. B **60**, 399 (1989).

[27] T. Nagai, S. Ohta, K. Kawasaki, and T. Okuzono, Phase Trans. **28**, 177 (1990).

[28] F. Roters, P. Eisenlohr, L. Hantcherli, D. D. Tjahjanto, T. R. Bieler, and D. Raabe, Acta Mater. **58**, 1152 (2010).

[29] R. Logé, M. Bernacki, H. Resk, L. Delannay, H. Digonnet, Y. Chastel, and T. Coupez, Philos. Mag. **88**, 3691 (2008).

[30] M. R. Tonks, D. Gaston, P. C. Millett, D. Andrs, and P. Talbot, Comput. Mater. Sci. **51**, 20 (2012).

[31] D. Raabe and R. C. Becker, Model. Simul. Mater. Sci. Eng. **8**, 445 (2000).

[32] A. Kuprat, SIAM J. Sci. Comput. **22**, 535 (2000).

[33] J. Gruber, D. C. George, A. P. Kuprat, G. S. Rohrer, and A. D. Rollett, Scripta Mater. **53**, 351 (2005).

[34] M. Syha and D. Weygand, Model. Simul. Mater. Sci. Eng. **18**, 015010 (2010).

[35] E. A. Lazar, J. K. Mason, R. D. MacPherson, and D. J. Srolovitz, Acta Mater. **59**, 6837 (2011).

[36] R. D. MacPherson and D. J. Srolovitz, Nature (London) **466**, 1053 (2007).

[37] J. C. Tucker, A. R. C. III, A. R. Ingraffea, and A. D. Rollett, Model. Simul. Mater. Sci. Eng. **23**, 035003 (2015).

[38] J. K. Mason, Acta Mater. **125**, 286 (2017).

[39] E. Eren and J. Mason, VDlib, https://github.com/erdemeren/VDlib (2021).

[40] E. Eren and J. Mason, VDtrials, https://github.com/erdemeren/VDtrials (2021).

[41] S. Seol, C. W. Smith, D. A. Ibanez, and M. S. Shephard, in *2012 SC Companion: High Performance Computing, Networking Storage and Analysis* (IEEE Computer Society, Los Alamitos, CA, USA, 2012), pp. 1124–1132.

[42] G. Hansen and G. Perumpilly, RPI SCOREC, https://github.com/SCOREC (2021).

[43] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevMaterials.5.103802 for the topological transition algorithms, elaborations on the generalized collapse, use of the exterior shell for volume preservation, examples from simulations, and scaling of the computational resource usage.

[44] K. Paton, Commun. ACM **12**, 514 (1969).

[45] N. E. Gibbs, J. ACM **16**, 564 (1969).

[46] J. von Neumann, in *Metal Interfaces* (American Society for Metals, Cleveland, Ohio, 1952), pp. 108–110.

[47] W. W. Mullins, J. Appl. Phys. **27**, 900 (1956).

[48] R. Quey, P. Dawson, and F. Barbe, Comput. Methods Appl. Mech. Eng. **200**, 1729 (2011).