Journal of Computational Physics 228 (2009) 8015–8033

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Diffusion generated motion for grain growth in two and three dimensions

Matt Elsey *, Selim Esedoğlu, Peter Smereka

Department of Mathematics, University of Michigan, Ann Arbor, MI 48109, United States

ARTICLE INFO

Article history:
Received 24 April 2009
Received in revised form 17 July 2009
Accepted 21 July 2009
Available online 30 July 2009

Keywords:
Diffusion-generated motion
Mean curvature flow
Grain growth

ABSTRACT

An efficient algorithm for accurately simulating curvature flow for large networks of curves in two dimensions and surfaces in three dimensions on uniform grids is proposed. This motion arises in the technologically important problem of simulating grain boundary motion in polycrystalline materials. In this formulation grain boundaries are zero-level sets of signed distance functions. Curvature motion is achieved by first diffusing locally maintained signed distance functions followed by a reinitialization step. A technique is devised to allow a single signed distance function to represent a large subset of spatially separated grains. Hundreds of thousands of grains can be simulated using a small number of signed distance functions (in this work, 32 in two dimensions and 64 in three dimensions are more than sufficient) using modest computational hardware.

1. Introduction

A polycrystalline material contains many crystallites (often called grains), differentiated by varying orientation. These materials are very commonplace, including most metals and ceramics. The properties of the microscale polycrystalline structure affect macroscale properties of these materials, such as fracture strength and conductivity. As such, understanding the statistics of the microscale structure is of great interest to materials scientists. One important model consists of grains moving with a normal velocity equal to curvature with grain boundaries meeting at 120° at triple points (for example, see Mullins [21] or Hillert [11]). Much work has focused on obtaining various statistics of grains evolving in this manner since such quantities may hold the key to important macroscopic properties. In order to obtain robust results, large-scale simulations with hundreds of thousands to millions of grains would be ideal. This is a challenging computational problem, especially in three dimensions. We propose and demonstrate a new algorithm for large-scale simulations of this evolution, expanding on the work in [4].

According to the well-known model of Mullins [21], grain boundaries evolve with normal velocity given by

$$v_n = \mu\gamma\kappa,$$

where μ denotes the boundary mobility, γ is the surface tension, and κ is the mean curvature of the interface between grains. In many cases (e.g. the isothermal annealing of pure metals), the mobility and surface tension may be taken to be constant, so that the normal velocity of the interface (a curve in two dimensions and a surface in three dimensions) is proportional to mean curvature. We consider the simplest, yet still important, case: that of equal surface tensions for each grain. We set $\mu\gamma = 1$ for convenience. As shown in [22,34], this normal speed arises as gradient descent for the energy

$$E = \sum_{k<\ell} (\text{length of } \Gamma_{k\ell}),$$

* Corresponding author.
E-mail addresses: melsey@umich.edu (M. Elsey), esedoglu@umich.edu (S. Esedoğlu), psmereka@umich.edu (P. Smereka).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.07.020

where $\Gamma_{k\ell}$ is the boundary between grain k and grain ℓ (and length is replaced by area in three dimensions). The standard boundary condition for this problem is the Herring angle condition, which for equal surface tensions states that triple junctions must meet at angles of 120°. This angle condition arises naturally from the algorithms used in this paper, as shown in [4].

The algorithm we use in this paper is based on *distance function based diffusion generated motion* developed by Esedoglu et al. [4] in which signed distance functions to interfaces are convolved with Gaussian kernels to generate a variety of geometric motions, including multi-phase motion by mean curvature, in an unconditionally stable manner. These algorithms are variants of the original *threshold dynamics* scheme of Merriman et al. [20] in which characteristic functions are diffused to generate mean curvature motion. Replacing characteristic functions in [20] with signed distance functions allows algorithms in [4] to attain good accuracy on modest sized uniform grids with no need for adaptive refinement. In [20] the authors also consider multi-phase mean curvature motion and propose an algorithm based on representing each phase by a level set function and evolving them via the standard level set PDE for mean curvature motion; see also [32]. A detailed discussion of the differences between the algorithm in [4] (an extended version of which will be developed and used here) and those in [20] can be found in [4].

The problem of simulating networks of grains moving by curvature flow has attracted much attention over the past few decades and many different computational approaches have been proposed. We feel that the algorithm presented here has advantages over previous formulations. Indeed, we are able to perform well-resolved simulations of grain networks in both two and three dimensions on a scale significantly larger than previously reported in the literature. We can easily simulate more than 250,000 grains in two dimensions, and 100,000 grains in three dimensions.

In previous work, grain boundary networks moving by curvature flow have been simulated by front tracking techniques in both two (e.g. [14]) and three dimensions (e.g. [31]). Computational efficiency is the big advantage of this approach since all the computational resources are concentrated on the interface. A fundamental difficulty of this approach is managing the plethora of topological changes that can occur as grains disappear. These methods must explicitly detect and handle each topological change by some selection of rules and require that the triple point condition be maintained separately. Furthermore, it is difficult to check if edges (in two dimensions) or surfaces (in three dimensions) cross using explicit methods. In three dimensions, it is a particularly difficult task to enumerate the ways in which two explicitly represented surfaces might meet. Even more difficult, if not practically impossible, would be checking to see if any such collisions occur. Using such representations in practice requires making assumptions about the types of topological changes that can typically occur. These assumptions may leave out important transitions or allow for nonphysical artifacts such as the interpenetrating of phase boundaries. For examples of these topological transition rules, see the types of critical events permitted in simulations [14,31] and the discussion of the surface operations permitted in Brakke's Surface Evolver code in [2]. In two dimensions, it is expected (though not fully proven, see [18]) that boundary networks under pure curvature motion change topology only through junction collisions, greatly simplifying the class of interactions possible. However, no such expectation is held in three dimensions. Even in the two-phase case, one phase can pinch off and split into two pieces. In two dimensions, the addition of bulk energy terms to the energy will also generally result in more complicated topological changes.

The phase-field formulation (e.g. [6,10,16,13,29]) will ameliorate these difficulties, but introduce a problem of its own. In this approach, a phase function is evolved for each grain and the grain boundary is a level set of the phase function. These methods naturally handle the aforementioned difficulties associated with topological changes but require the phase function to have a sharp transition layer at the grain boundary. It is crucial that this layer be fully resolved in order to accurately approximate curvature flow. For example, Kim et al. [13] report that they need at least six grid points in the transition layer to achieve acceptable accuracy. This indicates that a typical grain size must have something like 25 grid points per dimension to be even marginally resolved – a serious limitation to the accurate simulation of a large number of grains. We show that, using the approach described in this paper, our grains are well-resolved with 10 grid points per dimension, and we can follow them as they shrink down to about 4 or 5 grid points per dimension with a few percent relative error. Similar difficulties are present using threshold dynamics [19,20,24,26]. Threshold dynamics methods can become "stuck" on uniform grids in regions where the interface moves slowly (although this situation can be remedied via adaptive refinement; see e.g. [25,24]).

Another approach is to use a Potts model via kinetic Monte Carlo techniques (e.g.[1]). This is essentially a different model and its connection with curvature flow is a difficult question. Finally we mention that there are level set techniques, different than those present here, that could also be employed to tackle this problem (e.g. [34,5]).

The algorithm used in this paper is able to capture many of the advantages of the above methods with few of the disadvantages. Our method represents the grain boundaries implicitly using a signed distance function thereby achieving subgrid accuracy on a uniform mesh. Our method naturally handles topological changes and naturally imposes the Herring angle conditions (i.e. 120°) at junctions. Therefore, we capture the advantages of the phase field method without the disadvantage of needing to resolve a transition layer. In addition, the algorithms proposed in this paper are unconditionally stable. We have developed a technique in which sufficiently separated grains are represented by the same distance function (something similar was done for a phase field method in [16]). In addition, some of the computational work can be confined to narrow band near the grain boundaries. In this way we keep some of the advantage of the front-tracking formulations.

In this work, we implement the algorithm introduced in [4] (more fully described in Section 2). We then present results from a large two-dimensional simulation (initially containing over 150,000 grains), and preliminary results from a three-dimensional simulation (beginning with over 15,000 grains) in Section 3 (detailed three-dimensional results with over 100,000 grains will be published separately). We conclude by presenting an extension to the algorithm in Section 4 which allows for the inclusion of bulk energy terms into the energy functional for which our evolution constitutes gradient descent dynamics.

2. Algorithm

The signed distance function-based diffusion generated algorithm for motion by mean curvature of multiple phases as proposed in [4] is reproduced below. The evolution generated by this algorithm is proven to generate motion by mean curvature and to satisfy the symmetric Herring angle condition at triple junctions [4] in 2D (and hence also along triple lines in 3D).

We shall first describe the algorithm presented in [4] using the following notation: The set of points contained in the kth grain at time $t = n\Delta t$ will be denoted as Σ_k^n where Δt is the time step. The signed distance function from the boundary of Σ_n^k is denoted as d_n^k. Our sign convention is such that $d_n^k > 0$ for points in Σ_n^k. Further we suppose that there are N grains. The algorithm devised in [4] is then given by:

1. Given the initial sets $\Sigma_1^0, \ldots, \Sigma_N^0 \subset \mathbb{R}^m$ construct the corresponding signed distance functions d_k^0 (i.e. $\Sigma_k^0 = \{x : d_k^0(x) > 0\}$). For $n = 0, \ldots, n_{\max}$, perform steps 2–4.
2. Form the convolutions: $A_k(x) := K_{\Delta t} * d_k^n$ for $k = 1, \ldots, N$, where $K_{\Delta t}$ is

   $$K_{\Delta t} = G_{\Delta t} \quad \text{or} \quad K_{\Delta t} = \frac{1}{4}\left(4G_{\frac{3}{2}\Delta t} - G_{3\Delta t}\right),$$

   and $G_{\Delta t}$ is the fundamental solution of the heat equation:

   $$G_{\Delta t}(x) = \frac{1}{(4\pi\Delta t)^{\frac{m}{2}}} e^{-\frac{|x|^2}{4\Delta t}}.$$

3. Construct $B_k(x)$ for $k = 1, \ldots, N$ to remove overlaps and vacuums from the convolution step:

   $$B_k(x) = \frac{1}{2}\left(A_k(x) - \max_{\ell}\{A_\ell(x) : \ell \neq k\}\right)$$

4. Construct the updated signed distance function $d_k^{n+1}(x)$ for $k = 1, \ldots, N$ according to

   $$d_k^{n+1}(x) = \textbf{Redist}(B_k(x)).$$

Remarks. The operation **Redist**$(B(x))$ means construct a signed distance function from the zero-level set of $B(x)$. Formally the algorithm has the same order of accuracy using either $G_{\Delta t}$ or $K_{\Delta t} = \frac{1}{4}\left(4G_{\frac{3}{2}\Delta t} - G_{3\Delta t}\right)$, however the spatial truncation error of the second kernel is devoid of terms involving derivatives of curvature; see [4] for details. The *redistribution* or *projection* step 3 of the algorithm above is the same as in the threshold dynamics case [20] and is motivated by the well-known phase-field formulation (e.g. [3,10]) of the problem.

As stated here, the algorithm uses one distance function per grain. The improvement to the algorithm proposed in this work stems from the observation that a single distance function can serve to represent many grains that are not immediate neighbors of each other. Further, if we demand that only grains which are sufficiently far apart share the same distance function, then potential interactions that could occur during the convolution step will be negligible.

2.1. Extension

Here we present an extension of the algorithm from the previous section that allows one to use the same distance function for multiple grains. The setting for this algorithm is slightly different than the previous one. We begin as before, namely with N grains Σ_k^0 with $k = 1, \ldots, N$, but initialize only M signed distance functions, d_ℓ^0 with $\ell = 1, \ldots, M$, where $M < N$. These d_ℓ^0 have the property that they are the signed distance function for a collection of disjoint grains, and the union of these collections consists of all the grains. As the algorithm proceeds, it must check to be sure that this disjointness property is maintained. If it appears that it is about to fail (i.e. two distinct grains in one of the collections become too close), various grains will need to be reassigned to different distance functions and if need be a new distance function will be introduced. We call this operation *swapping*. A crucial point is that $M \ll N$ unless one considers some pathological initial conditions. Even then, since the evolutions considered here are regularizing with a preference towards grains with small isoperimetric ratios, M is expected and observed to be fairly small at subsequent times during the evolution.

1. Given the initial sets $\Sigma_1^0, \ldots, \Sigma_N^0$ construct M functions, d_k^0 so that each is the signed distance function for a collection Ξ_k of disjoint grains so that $\cup_{k=1}^M \Xi_k = \cup_{k=1}^N \Sigma_k$.
   For $n = 0, \ldots, n_{\max}$, perform steps 2–5.
2. **Convolution**: Form the convolutions: $A_k(x) := K_{\Delta t} * d_k^n$ for $k = 1, \ldots, M$
3. **Comparison**: Construct $B_k(x)$ for $k = 1, \ldots, M$ to remove overlaps and vacuums from the convolution:

$$B_k(x) = \frac{1}{2}(A_k(x) - \max_{\ell}\{A_\ell(x) : \ell \neq k\})$$

4. **Redistancing**: Construct the signed distance function $d_k^{n+\frac{1}{2}}(x)$ in a tubular neighborhood of the zero-level set of $B_k(x)$. For $k = 1, \ldots, M$ according to

$$d_k^{n+\frac{1}{2}} = \textbf{Redist}(B_k(x))$$

5. **Swapping**: If necessary swap appropriate grains between signed distance functions to ensure that all the grains associated to given signed distance function remain well separated. Redistance around swapped grains and denote the resulting signed distance functions as d_k^{n+1}.

2.1.1. Details

We now describe the steps of the above algorithm in more detail in the fully discrete setting. For convenience, the formulas are written down in the 2D setting, but extend trivially to all dimensions.

2.1.1.1. Convolution. We define the convolution kernels $G_{\Delta t}$ and $K_{\Delta t}$ in terms of the space-discretized solution to the heat equation $u_t = u_{xx} + u_{yy}$. Suppose the grid discretizes $[0,1]^2$, with equal grid spacing ($\Delta x = \Delta y$). Let $u_{ij}(t)$ be the space-discretized approximation to $u(x, y, t)$ at $(i\Delta x, j\Delta y, t)$. Using centered differencing in space, we obtain:

$$\frac{d}{dt}u_{ij}(t) = \frac{1}{\Delta x^2}((u_{i+1j} - 2u_{ij} + u_{i-1j}) + (u_{ij+1} - 2u_{ij} + u_{ij-1})). \tag{1}$$

Apply the discrete Fourier transform in space to obtain,

$$\frac{d}{dt}\hat{u}_{r,s} = \frac{2}{\Delta x^2}(\cos(2\pi s\Delta x) + \cos(2\pi r\Delta x) - 2)\hat{u}_{r,s}.$$

Given initial data $\hat{u}_{r,s}(t)$, this ODE has solution $\hat{u}_{r,s}(t + \Delta t)$

$$\hat{u}_{r,s}(t + \Delta t) = \hat{u}_{r,s}(t)\exp\left(\frac{-2\Delta t}{\Delta x^2}(2 - \cos(2\pi r\Delta x) - \cos(2\pi s\Delta x))\right).$$

Therefore, the discrete heat equation (1) has solution $u_{ij}(t + \Delta t) = u_{ij} * (G_{\Delta t})_{ij}$ where $*$ denotes the discrete convolution and $(G_{\Delta t})_{ij}$ is defined via its discrete Fourier transform:

$$(\widehat{G}_{\Delta t})_{r,s} = \exp\left(\frac{-2\Delta t}{\Delta x^2}(2 - \cos(2\pi r\Delta x) - \cos(2\pi s\Delta x))\right).$$

Finally, we implement a Richardson extrapolation-like procedure to improve the accuracy of the kernel, (as described in [4]), and define:

$$K_{\Delta t} = \frac{1}{3}\left(4G_{\frac{3}{2}\Delta t} - G_{3\Delta t}\right).$$

2.1.1.2. Comparison. The convolution step gives diffusion generated motion along simple interfaces, but may create overlaps or vacuums at junctions where multiple interfaces meet. To enforce the desired no-overlap/no-vacuum condition, we apply a comparison step to obtain the updated level set functions

$$B_k(x) = \frac{1}{2}(A_k(x) - \max\{A_\ell : \ell \neq k\}),$$

This formulation guarantees that exactly one of the $B_k(x)$ (for $k = 1, \ldots, M$) is positive at any given location x. Furthermore, this procedure ensures that the symmetric Herring angle condition is maintained at all triple points [4].

2.1.1.3. Redistancing. At each time step, we need to compute the signed distance function $d_k^{n+\frac{1}{2}}$ to a union Ξ_k of disjoint grains; we need the distance function only in a tubular neighborhood of the boundary $\partial\Xi_k$ of Ξ_k. The width of the tubular neighborhood is proportional to the kernel width, which in turn is proportional to $\sqrt{\Delta t}$. We make use of a two-phase redistancing

algorithm that depends only on the input values $B_k(x) = \frac{1}{2}(A_k(x) - \max\{A_\ell : \ell \neq k\})$ at grid points within two grid points of the interface. For the remainder of the discussion of redistancing, we drop the subscript k for convenience, as each set is updated independently of the others.

Define the set of boundary points β to be

$$\beta = \{(i,j) : (|\mathrm{sgn}(B_{i+1\,j}) - \mathrm{sgn}(B_{i-1\,j})| + |\mathrm{sgn}(B_{i\,j+1}) - \mathrm{sgn}(B_{i\,j-1})|) > 0\},$$

where $B_{i,j} = B(i\Delta x, j\Delta y)$ and

$$\mathrm{sgn}(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x = 0, \\ -1 & \text{if } x < 0. \end{cases}$$

These boundary values are set initially to respect the condition that $|\nabla d| \equiv |\nabla d^{n+\frac{1}{2}}| = 1$, while moving the interface as little as possible. Specifically, we set

$$d_{ij} = \frac{B_{ij}}{|\nabla B_{ij}|}, \quad \forall (i,j) \in \beta.$$

Typically, the centered difference approximation is appropriate for $|\nabla B_{ij}|$, but does not work well on small grains. See Fig. 1 for an illustration in one dimension. The solid line is the exact signed distance function to the thick bar shown at the bottom of the plot. The dashed line shows the centered difference approximation to the gradient of the signed distance function at the indicated point. Upwind differencing is only first-order accurate in general, but gives a more accurate value for the gradient at this point. We define

$$|\nabla B_{ij}|_1 = \sqrt{\left(\frac{B_{i+1\,j} - B_{i-1\,j}}{2\Delta x}\right)^2 + \left(\frac{B_{i\,j+1} - B_{i\,j-1}}{2\Delta y}\right)^2},$$

$$|\nabla B_{ij}|_2 = \sqrt{\left(\max\left(\frac{|B_{i+1\,j} - B_{i\,j}|}{\Delta x}, \frac{|B_{i\,j} - B_{i-1\,j}|}{\Delta x}\right)\right)^2 + \left(\max\left(\frac{|B_{i\,j+1} - B_{i\,j}|}{\Delta y}, \frac{|B_{i\,j} - B_{i\,j-1}|}{\Delta y}\right)\right)^2},$$

and define

$$|\nabla B_{ij}| = \begin{cases} |\nabla B_{ij}|_1 & \text{if } \frac{1}{2}|\nabla B_{ij}|_2 \leqslant |\nabla B_{ij}|_1 \leqslant 2|\nabla B_{ij}|_2, \\ |\nabla B_{ij}|_2 & \text{otherwise.} \end{cases}$$

We fix the values d_{ij} for all $(i,j) \in \beta$, and first generate a first-order in space accurate approximation of the signed distance function using fast sweeping as described in [30,33]. Then we perform an iterative second-order accurate method (described in [23]) for a limited number of iterations on this output. The input $B(x)$ may be far from a distance function near junctions.

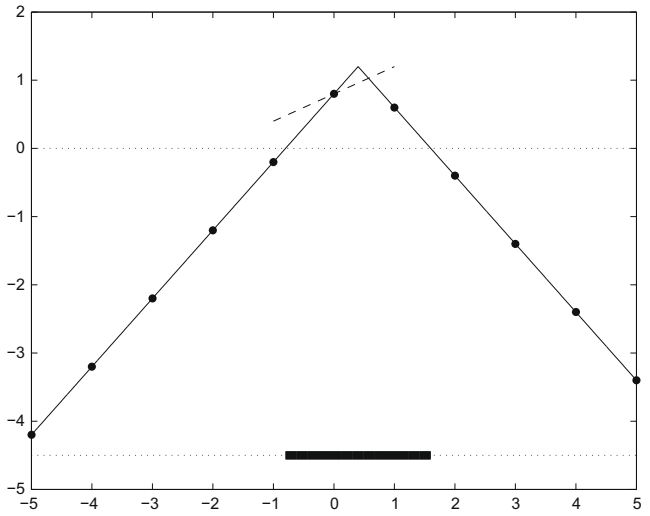


Fig. 1. Failure of centered differencing on small grains. The thick black line indicates the set $\Sigma = \{x : d(x) > 0\}$. The slope of the dashed line indicates the centered difference approximation to $|\nabla d|_1$ at the specified point. The upwind differencing finds $|\nabla d|_2$=1, the correct value for $|\nabla d|$.

Performing the fast sweeping initially allows us to perform only a limited number of iterations with the second-order method, which is the most time-intensive part of the algorithm.

2.1.1.4. Swapping. The swapping step allows each signed distance function to store many grains safely. Without this step, it would be necessary to maintain each individual grain in a separate set to guarantee that coalescence could not occur. For example, in a calculation performed on a 4096^2 grid, we begin with over 160,000 grains and use only $M = 32$ sets to track them all. The algorithm introduces new signed distance functions if needed to ensure that inter-grain spacing is maintained. In our experience, the algorithm typically does not require more than $M = 32$ sets in two dimensions, and not more than $M = 64$ sets in three dimensions. As the grain network evolves we find it will rarely, if ever, introduce new signed distance functions. Without the savings of both memory and computational time permitted by this additional step (allowing the number of sets, M, to satisfy $M \ll N$, the total number of grains), such a large-scale computation would be impossible.

Our approach, described below, is similar to that of Krill and Chen [16]. They reassign grains to prevent any particular grain from being maintained in the same set as any of its nearest or second-nearest neighbors. We, instead, make sure that any two grains described by the same signed distance function are not too close (we will be more precise shortly). This distinction is significant for our algorithm, as spatial separation is key to prevent distinct grains, described by the same signed distance function, from interacting during the convolution step (the width of the kernel is of course related to the time step size, which can be large thanks to the unconditional stability of the proposed algorithms).

To describe this algorithm we must first outline some notation. First recall that the set $\Xi_k = \{x : d_k(x) > 0\}$ corresponds to a collection of disjoint grains. These grains are the connected components of Ξ_k. We say that two grains, say, Σ_a and Σ_b in Ξ_k are τ-close if their union is completely contained in the same connected component of $\{x : d_k(x) > -\tau\}$ (which is trivially checked by comparing the $(-\tau)$-super level set membership of any two grid points belonging to Σ_a and Σ_b). We choose $\tau > 0$ to be proportional to $\sqrt{\Delta t}$ to prevent distinct grains in Ξ_k from interacting during the convolution step. In the simulations presented in Section 3, we take $\tau \approx 6\sqrt{\Delta t}$. See Fig. 2 for illustration of the selection process.

Here we describe the new step in the algorithm in greater detail.

1. For $k = 1, \ldots, M$, initialize $d_k^{n+1} = d_k^{n+\frac{1}{2}}$.
2. For $k = 1, \ldots, M$: While there are at least two grains in d_k^{n+1} that are τ-close, select any τ-close pair of grains from d_k^{n+1}, and perform steps 3–7.
3. Select the smaller grain from the pair (measured by the number of grid points contained in each grain) and denote the signed distance function they are associated with as k. Let $d_G(x)$ be the signed distance function to the boundary of the selected grain and define the set $X = \{x : d_G(x) > -\tau\}$.
4. Find a set, Ξ_ℓ^{n+1}, such that $\ell \neq k$ and $d_\ell^{n+1}(x) \leqslant -\tau \ \forall\ x \in X$. If such a set cannot be found, increment $M \leftarrow M + 1$, initialize $d_M^{n+1} = -\tau$, and select $\ell = M$.
5. Add the grain to Ξ_ℓ^{n+1} by setting $d_\ell^{n+1}(x) = d_G(x) \ \forall\ x \in X$.
6. Remove this same grain from Ξ_k^{n+1} by setting $d_k^{n+1}(x) = -\tau \quad \forall x \in X$.
7. Redistance d_k^{n+1} and d_ℓ^{n+1} on the set X.

3. Numerical results

In this section, two types of numerical results are presented. First, we display the convergence of our algorithm an exact solution known for two-phase motion, and demonstrate that we match a known solution well in three-phase motion. Next, we examine the spatial and temporal convergence of our algorithm in a multiphase case for which no exact solution is known. Finally, we demonstrate the quality of our algorithm on large data sets simulating normal grain growth using statistical measures such as average grain size and grain area distribution.

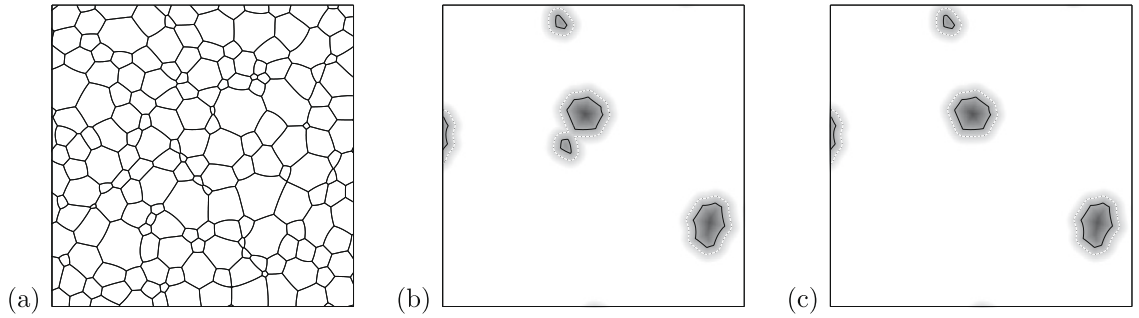


Fig. 2. (a) Part of a grain pattern. (b) Overlay of the boundary of $d^{n+\frac{1}{2}} > 0$ (solid line) and $d^{n+\frac{1}{2}} > -\tau$ (dotted line) on the signed distance function $d^{n+\frac{1}{2}}$. (c) Same for d^{n+1} after a grain is removed and d^{n+1} is recalculated.

3.1. Convergence to exact solutions in two phase motion

We begin by verifying that our algorithm accurately simulates two phase motion by mean curvature on the simplest examples in two and three dimensions: the circle and the sphere. In each case, the motion reduces to the simple ordinary differential equation,

$$\dot{r}(t) = \kappa = \frac{-C}{r}, \tag{2}$$

where $C = 1$ for the circle and $C = 2$ for the sphere. Eq. (2) has the solution

$$r(t) = \sqrt{r(0)^2 - 2Ct}.$$

In our tests, we chose $r(0) = 0.25$. For the circle, we took as our stopping time $t^\star = 3/128$, and for the sphere $t^\star = 3/256$, so that the exact solution has $r(t^\star) = 0.125$. While the evaluation of the curvature is second-order accurate in space and time, the method as a whole is expected to show linear convergence in both space and time. This is due to the time integration, which operates under the assumption that curvature remains constant through each iteration. See Tables 1 and 2 for numerical results. The results labeled "Exact Redist Result" were obtained by replacing the distance function at the redistancing step by the exact distance function for a circle or sphere with the same zero-level set at each step. We note that the linear convergence rate is strongly indicated by the exact redistancing results for resolutions $\geqslant 256 \times 256$ for the circle and $\geqslant 64 \times 64 \times 64$ for the sphere. Our redistancing technique causes some cancellations of error at low resolutions, but follows the linear convergence trend shown by the exact redistancing results well at higher resolutions.

3.2. Comparison to known profile in three phase motion

In this test of three phase motion, we choose homogenous Neumann boundary conditions and consider a T-junction initial condition as shown in Fig. 3. It was shown in [10] that there is an exact solution for this initial T-junction geometry consisting of a steady profile moving at constant speed. The profile is given for $0 \leqslant x \leqslant 0.5$ by:

$$y(x,t) = \frac{3}{\pi}\log\left(\cos\left(\frac{\pi x}{3}\right)\right) + (y_0 - vt),$$

where v is the velocity of the profile and y_0 is determined by the initial location of the T-junction. Fig. 3 shows the close agreement between the predicted profile where vt was chosen to match the computed profile at $x = 0$. These results were computed on a 128×128 grid. In (a), the predicted profile and computed profiles are indistinguishable. Parts (b), (c), and (d) zoom in successively on the final computed profile and the associated prediction. We see that the results differ by less than 10^{-3} for all x. The triangular "split" in the profile seen in the zoomed views is purely a visualization artifact.

3.3. Convergence of multiphase motion

There are no explicit solutions available for the evolution of a general grain pattern, especially through topological changes. Furthermore, there is no rigorous notion of a weak solution with uniqueness through such events. However, it is reasonable to expect that at least the statistical descriptors of the network (for example, the distribution of grain areas) to be eventually independent of further spatial or temporal resolution under refinement. To test this expectation, we choose an initial condition containing 11,217 grains and sample on grids of 1024×1024, 2048×2048 and 4096×4096, refining the discretization in *both space and time*. We keep the ratio between Δx and Δt constant to refine both spatial discretization and the effective sampling rate of the Gaussian kernel. Table 3 describes the simulations.

The simulation runs for total time $80/1024^2$. Approximately 6000 grains disappear during this evolution, corresponding to many thousands of topological transitions (i.e. elimination of edges). Less than half the original number of grains remain at the end of the simulation. The number of grains remaining after each simulation varies (shown in Table 3), but this is not a

Table 1
Convergence check: motion by curvature of a circle.

Resolution	Iterations	$r(t^\star)$	% Error	Exact Redist Result	% Error
8×8	7	0.143812	15.0493	0.068539	45.1687
16×16	15	0.124301	0.5595	0.112370	10.1040
32×32	30	0.123497	1.2022	0.123544	1.1649
64×64	60	0.123918	0.8653	0.124435	0.4520
128×128	120	0.124246	0.6110	0.124562	0.3501
256×256	240	0.124585	0.3323	0.124751	0.1996
512×512	480	0.124797	0.1627	0.124870	0.1043
1024×1024	960	0.124900	0.0800	0.124934	0.0532
2048×2048	1920	0.124952	0.0383	0.124966	0.0268

Table 2
Convergence check: motion by curvature of a sphere.

Resolution	Iterations	$r(t^\star)$	% Error	Exact Redist Result	% Error
$8 \times 8 \times 8$	7	0.153862	23.0896	0.113474	9.2212
$16 \times 16 \times 16$	15	0.128935	3.1484	0.120004	3.9964
$32 \times 32 \times 32$	30	0.124040	0.7682	0.123044	1.5648
$64 \times 64 \times 64$	60	0.123881	0.8951	0.124010	0.7923
$128 \times 128 \times 128$	120	0.124289	0.5688	0.124481	0.4150
$256 \times 256 \times 256$	240	0.124627	0.2988	0.124735	0.2121

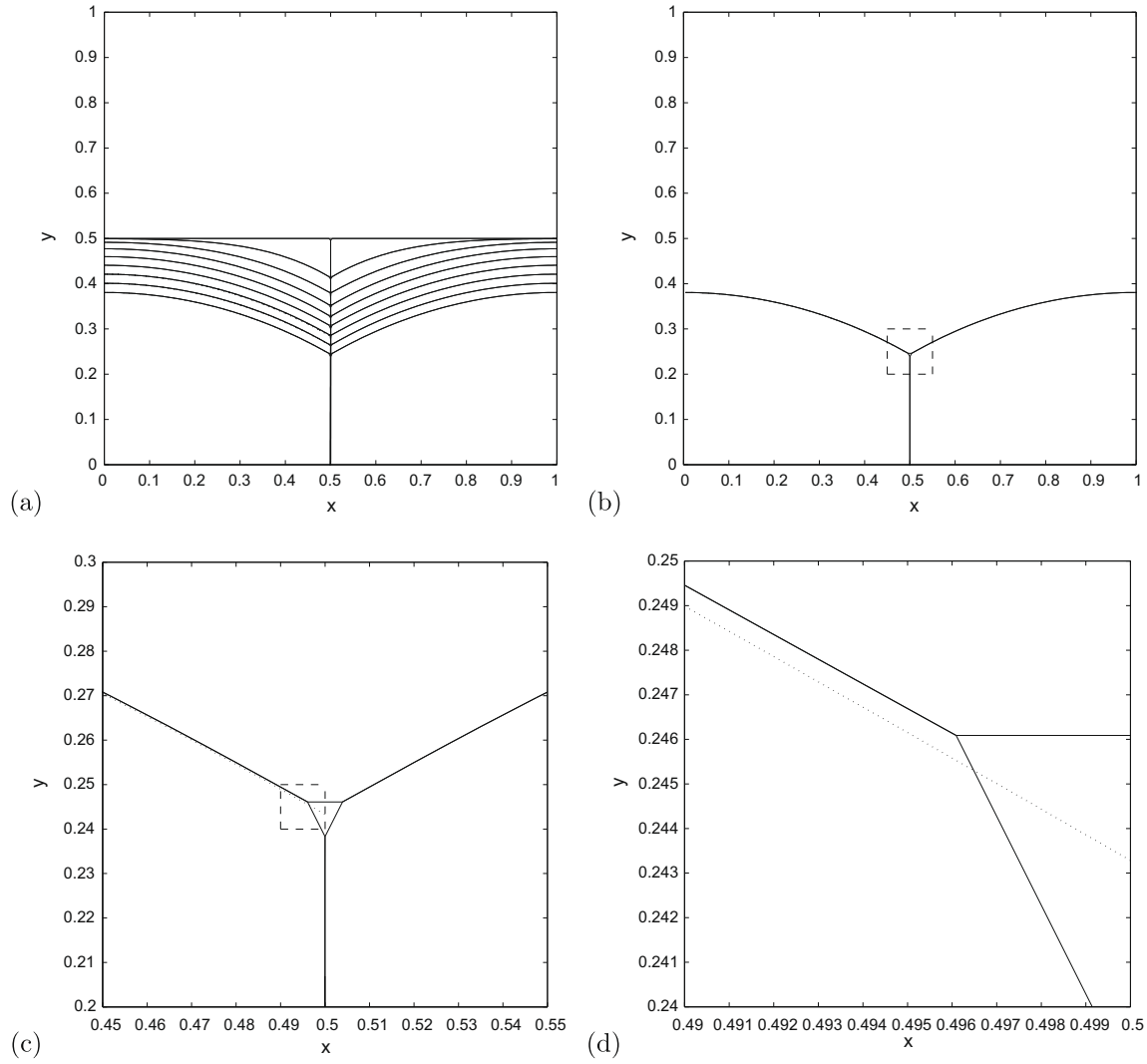


Fig. 3. (a) Interface shown at various times in evolution (solid). The exact profile is overlaid (dotted) once a constant profile is attained. (b)–d) Successive zoom in to the computed profile (solid) and exact profile (dotted). The triangular region visible in (c) and (d) is a visualization artifact.

Table 3
Input parameters and final number of grains for simulations testing convergence of multiphase motion. *nt* indicates the number of time steps used, and $n(nt \cdot \Delta t)$ indicates the number of grains remaining at the end of the simulation.

Simulation	Δx	Δt	nt	$n(nt \cdot \Delta t)$
1024×1024	$1/1024$	$0.8/1024^2$	100	5177
2048×2048	$1/2048$	$0.4/1024^2$	200	5286
4096×4096	$1/4096$	$0.2/1024^2$	400	5398

good measure of the convergence, because there is a lower limit on the size of grains that can be accurately represented by a given time step. Instead, it is more appropriate to look at statistical quantities (such as the distribution of grain areas) and the actual microstructure resulting from the simulations. These are shown in Fig. 4. The histogram of grain areas demonstrates that each simulation has a very similar distribution of grain areas for grains with areas larger than approximately 10^{-4}. The deviation in the total number of grains remaining in each of the simulations can be attributed almost fully to the differences in the first bin alone of the histogram.

A small section of the microstructure is shown in Fig. 4(c), one 25th of the entire computational domain for each simulation. There are very few differences between the evolution with $\Delta x = 1/4096$ and with $\Delta x = 1/2048$. There are considerably more differences between these simulations and the result with $\Delta x = 1/1024$, but even there, the majority of the microstructure exactly matches that computed at higher resolutions. The agreement between the results is remarkable con-

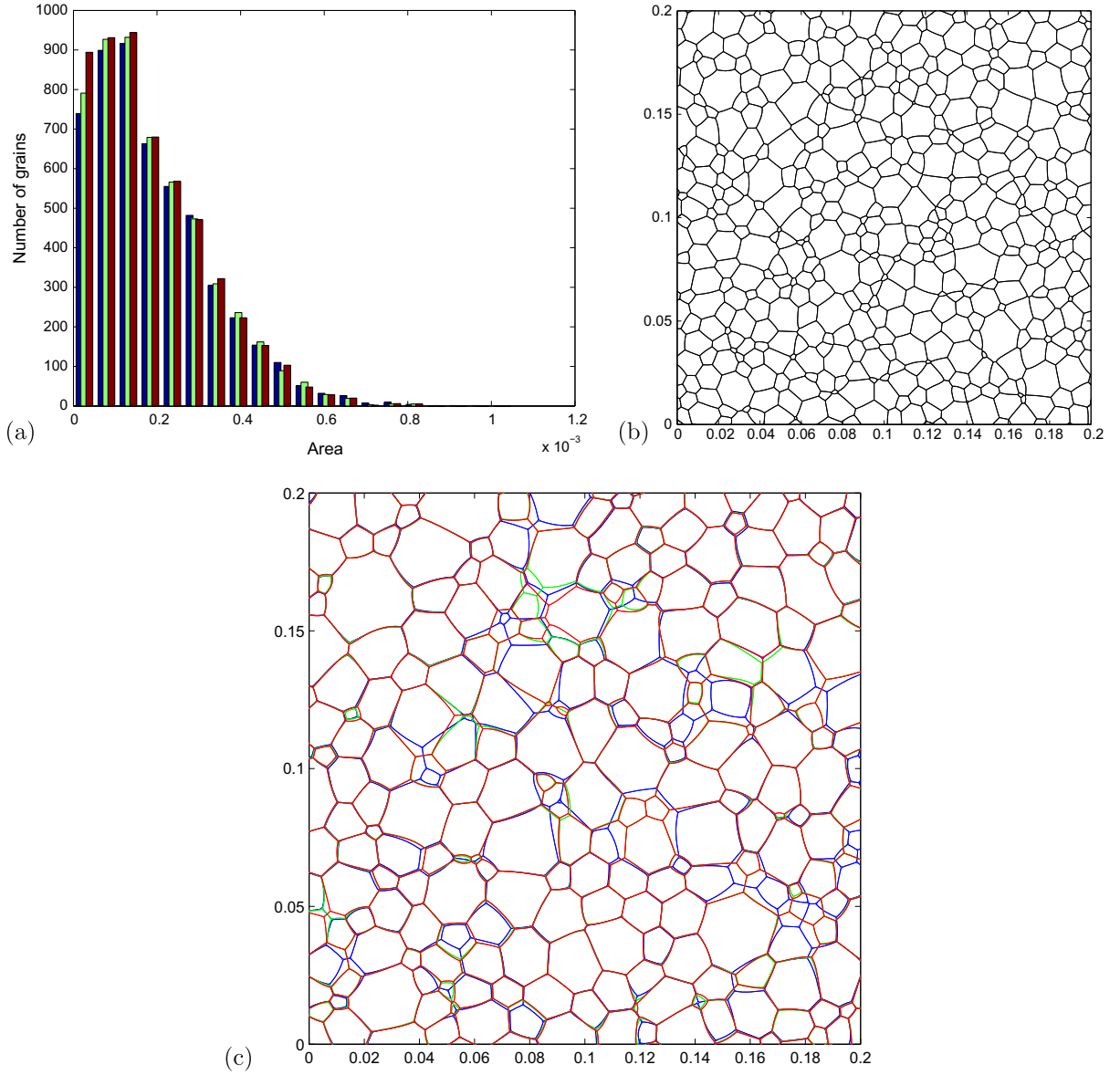


Fig. 4. Multiphase grain motion convergence test results. Each test was run for total time $80/1024^2$. $\Delta x = 1/1024, \Delta t = 0.8/1024^2$ corresponds to blue, $\Delta x = 1/2048, \Delta t = 0.4/1024^2$ corresponds to green, and $\Delta x = 1/4096, \Delta t = 0.2/1024^2$ corresponds to red in (a) and (c). (a) The histogram of grain areas for the various simulations. Note that significant differences in the histograms occur only for very small grains. (b) A subsection of the initial condition for these simulations containing 1/25 of the simulation domain. Approximately 6000 grains disappear between this state and the final states pictured in (c), corresponding to many thousand topological transitions. (c) The same subsection of the domains at time $80/1024^2$. Note that the agreement between the red and green microstructure is better than the agreement between blue and either red or green microstructure. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

sidering the great disparity between the initial condition (Fig. 4(b)) and the simulation results and the thousands of topological changes occurring in the evolution.

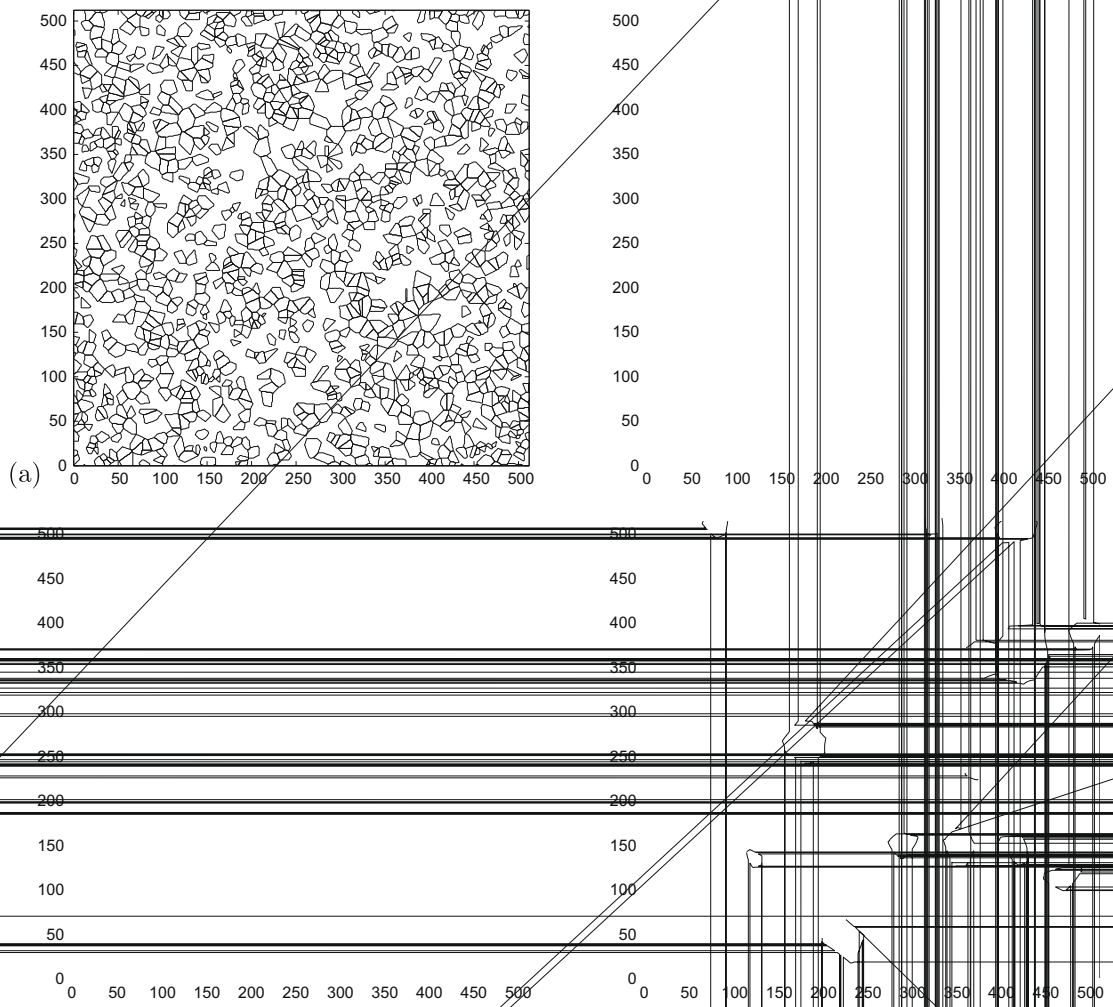
3.4. Large-scale simulation of grain growth in two dimensions

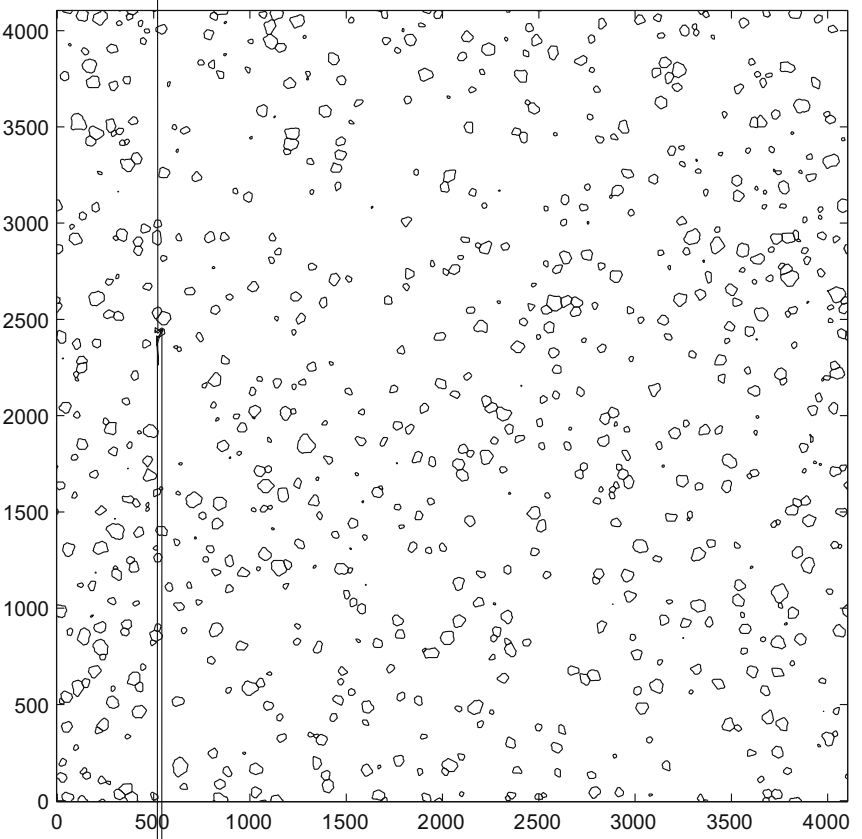
We concern ourselves with the standard case, in which all interfaces move with normal velocity equal to the curvature of the interface. In future work we will investigate varying surface tensions and bulk energies. As previously shown in [4], this algorithm naturally imposes the symmetric angle condition (all triple junctions meet at 120° angles).

We discretize the $[0,1] \times [0,1]$ domain with $\Delta x = \Delta y = 1/4096$. We use periodic boundary conditions, natural as the interactions between grains are short-ranged. Our initial condition contains 166,927 grains, and was obtained by constructing the Voronoi diagram for random points with a uniform distribution in the computational domain. We take 1500 time steps with time step $\Delta t = 0.8/4096^2$, at the end of which 11,217 grains remain and the evolution of the system has slowed considerably. (An adaptive time stepping strategy would therefore be prudent, and is entirely feasible given the unconditional stability of our algorithms, but this was not carried out: All experiments in this paper were generated using uniform time steps.) For the entire evolution, we maintain just 32 sets to track all the grains. For snapshots of the evolution, see Fig. 5. These all show 1/64 of the full grain pattern. Fig. 6 shows the final full grain pattern. Earlier time steps contain too many grains for the full pattern to be viewable on a single page.

Recall that the energy of the system is given by

$$E = \sum_{k<\ell} (\text{length of } \Gamma_{k\ell})$$





in two spatial dimensions (and by summing the interfacial area in three dimensions). In terms of the signed distance functions $d_k(x)$, this energy can be written in terms of the Dirac delta function, δ, as

$$E = \frac{1}{2}\sum_{k=1}^{M}\int_{\Omega}\delta(d_k(x))dx.$$

The factor $1/2$ arises since this formula counts interfaces two times. We can discretize E, in two space dimensions, as

$$E = \frac{\Delta x^2}{2}\sum_{k}\sum_{i,j}\tilde{\delta}(d_k(x_i, y_j)). \tag{3}$$

We use a first-order discretization of the delta function, $\tilde{\delta}$, as proposed in [28]. The discrete version of Eq. (3) in three dimensions is similar. The energy, E, is evaluated at each iteration and found to be strictly decreasing at every time step. See

Fig. 7(a). Note that our method handles topological changes naturally, and that the energy of the system decreases even as over 150,000 grains disappear throughout 1500 iterations.

Several analytical approaches predict the mean grain radius $\langle r\rangle$ to grow as $\langle r\rangle \approx Ct^{1/2}$ (for example, see [7,11,17]). In normal grain growth, characterized by self-similarity of the distribution of $r/\langle r\rangle$, it immediately follows that the average grain area $\langle a\rangle$ is predicted to grow linearly as a function of time. We compute the average grain area as

$$\langle a\rangle = \frac{1}{N(t)},$$

where $N(t)$ is the number of grains surviving at time t, and see linear growth following a short relaxation time. This relaxation time is due to the initial condition, which we chose to be the Voronoi diagram for points distributed uniformly at random throughout the domain. Such an initial condition contains very few small grains and does not respect the Herring angle condition. Initially the evolution corrects the angle conditions and some time must pass before the smallest grains are found in the correct proportion so as to allow the mean grain area to increase linearly. See Fig. 7 (b) and (c).

Two other measures of interest are the relative grain area distribution and the number of edges distribution, as defined in [15]. Let $G(n,t)$ be the proportion of grains with n grain boundaries at time t, and $F(\xi,t)d\xi$ be the number of grains with relative area in $[\xi,\xi+d\xi]$ at time t, with $\xi = a/\langle a\rangle$. Normal grain growth is characterized by the self-similarity of F as t varies. See Fig. 8, which suggests that F is approximately self-similar for $t \geqslant 300\Delta t$.

Fig. 9 shows the evolution of the number of edges distribution. Note that the maximum of this distribution is initially at six sides, then shifts to five sides and back to six sides as the evolution proceeds. This behavior was seen in multiple simulation runs. The number of edges distribution is of general interest and has been reported for both simulations [27,9,12,15] and experiments [8]. The maximum has been found at both five and six sides. It is not known if this distribution should be self-similar through time as the relative grain area distribution is predicted to be (see Fig. 8).

3.5. Three-dimensional simulation of grain growth

Our algorithm and its implementation extends with only minor modifications to three dimensions very naturally. We again consider the case in which all interfaces (surfaces) move with normal velocity equal to the mean curvature of the surface. We discretize our computational domain the unit cube with $\Delta x = \Delta y = \Delta z = \frac{1}{256}$, and again apply periodic boundary conditions. We maintain 64 sets of disjoint grains to track the grains through iterations. In 3D, each grain can have many more neighbors than in 2D; thus, in general a larger number of sets Ξ_k are necessary to keep grains within the same set well-separated.

Here, our initial condition contains 16,767 total three-dimensional grains. See Fig. 11 for a view of grains contained in five of the 64 sets. We take 500 iterations, at which time 853 grains remain. See Fig. 10 for the evolution of the number of grains and average grain volume. We note that the dependence of $\langle v\rangle$ on t is non-linear, as opposed to the relationship of $\langle a\rangle$ on t in two dimensions. In [1], the authors report that the growth kinetics exhibit power-law behavior following an initial transition phase, specifically, they report

$$\langle v\rangle^{\frac{1}{3n}} = ct + d,$$

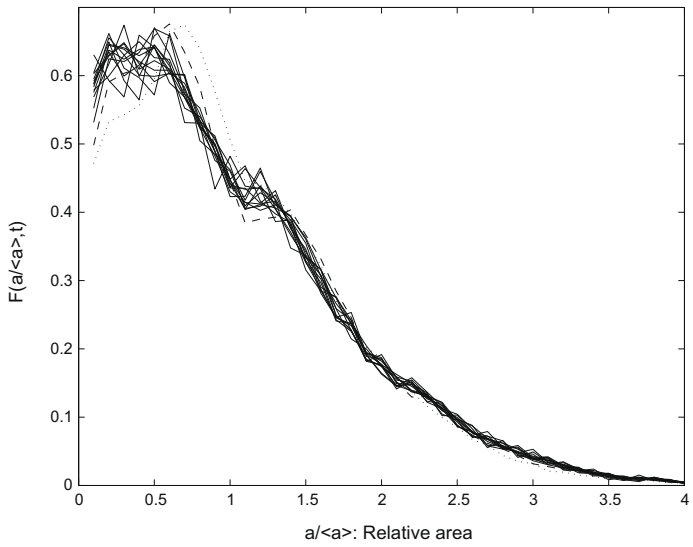
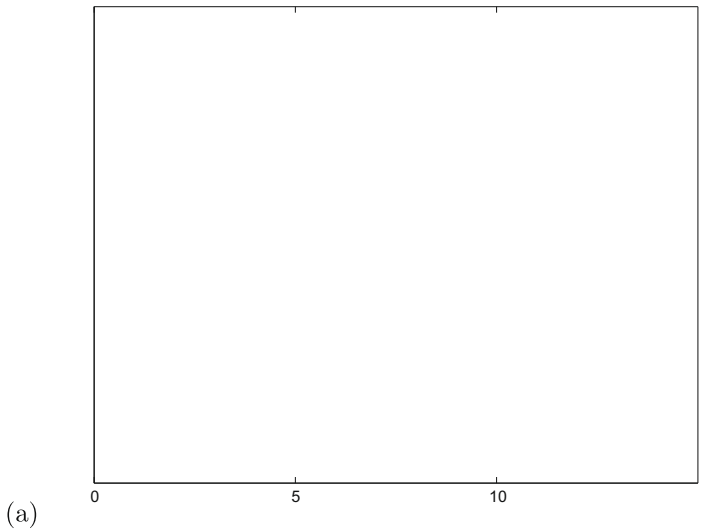
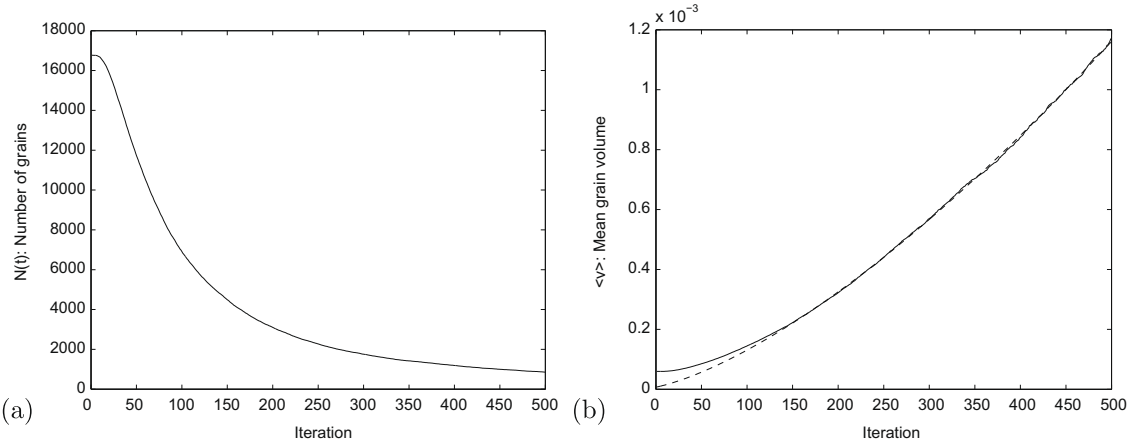


Fig. 8. Relative area probability densities at 100, 200, ..., 1500 iterations. $G(n,100\Delta t)$ is dotted, $G(n,200\Delta t)$ is dashed, and the rest are plotted as solid lines.



(a)



(a) (b)

Fig. 10. Three-dimensional simulation results: (a) $N(t)$, the number of grains, and (b) $\langle v \rangle$, the average grain volume (solid) plotted against $\langle v \rangle^{1/3n} = ct + d$, with $t_{min} = 150$ (dashed).

where n is identified as the kinetic exponent for grain growth, i.e.

$$\langle r \rangle \approx Ct^n$$

gives the long-term behavior of average grain size. The experimental results reported in [1] vary, reporting $1/4 \leqslant n \leqslant 1/2$ where they find $n = 0.48 \pm 0.04$ using Monte Carlo methods. Our simulation data is presented in Tables 4 and 5. Our results for $50 \leqslant t_{min} \leqslant 250$ give $0.48 \leqslant n \leqslant 0.55$ which is slightly larger than what was found in [1].

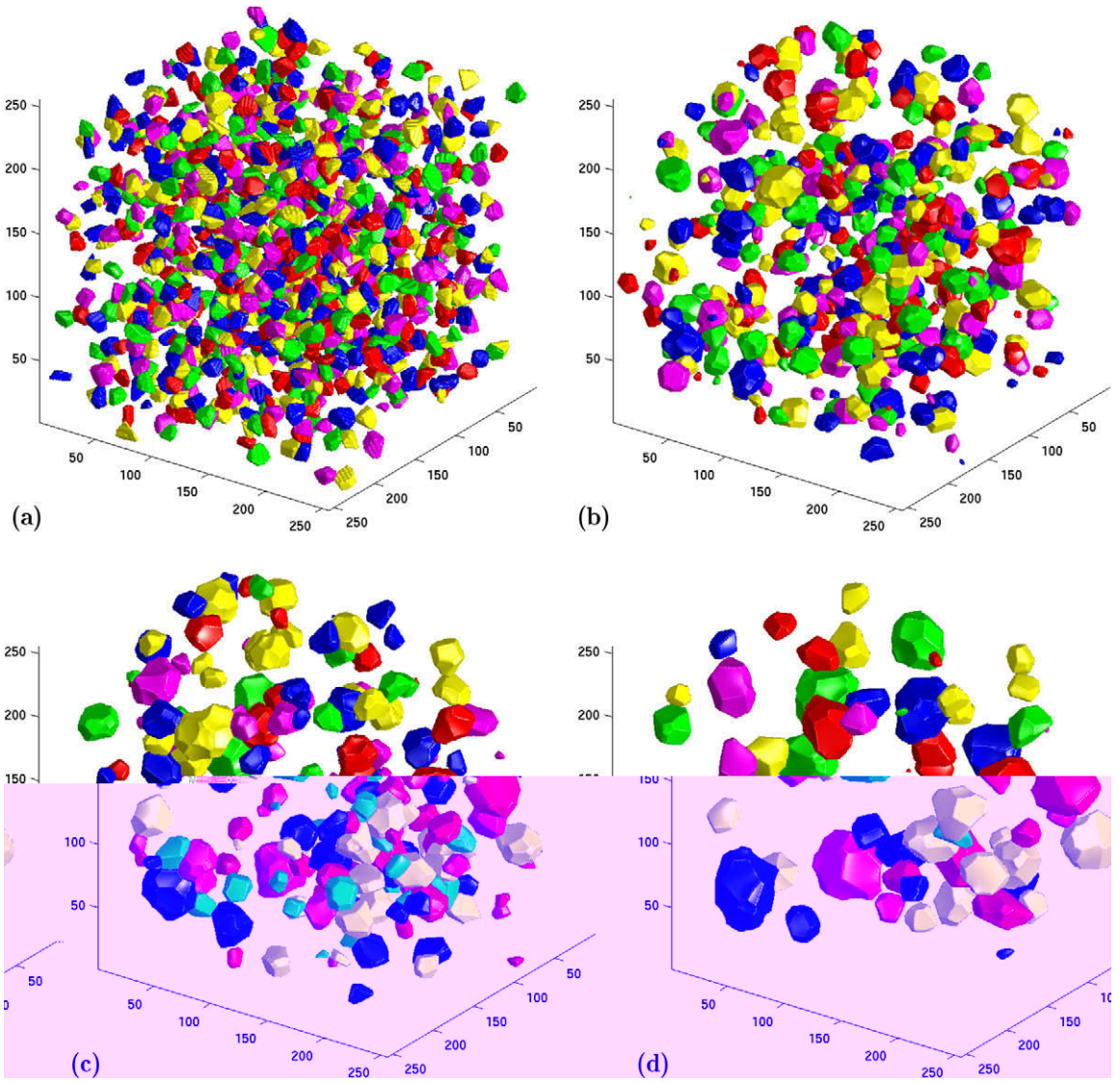


Fig. 11. Each subfigure shows the grains from five sets (of 64 total sets) at various times in the evolution. (a) The initial condition contains 16,767 total grains. (b) After 100 iterations, there are 6911 grains. (c) After 200 iterations, there are 3101 grains. (d) After 400 iterations, there are 1189 grains.

Table 4
Fit of data from iterations in range $t_{min} \leqslant t \leqslant 500$ to $\langle v \rangle^{1/3n} = ct + d$, with 95% confidence.

t_{min}	c	n
10	2.6×10^4	0.593 ± 0.007
50	3.4×10^4	0.548 ± 0.005
100	4.4×10^4	0.512 ± 0.005
150	5.3×10^4	0.487 ± 0.005
200	5.3×10^4	0.486 ± 0.009
250	3.5×10^4	0.544 ± 0.018

Figs. 11 and 12 show the same sets at later times in the evolution. Note that some grains may have swapped in or out of these sets during the evolution and thus may not be included in the visualization consistently, though large grains are swapped only rarely. Fig. 13 shows a single grain from two viewpoints after 500 iterations. At this time 853 grains remain in the simulation. This grain is approximately 40 grid points across in each dimension. The average effective grain radius as measured from grain volume at this time is approximately 17 grid cells. Thus the average grain is somewhat smaller than the featured grain, but not by a significant amount. The grain appears to be very well resolved on this grid. The individual facets are apparent and are separated by sharp edges.

Table 5
Circular grain growth with bulk energies: simulation and prediction of growth through final time $500\Delta t$, before circular grains begin to collide.

Initial radius	Final radius	Predicted radius	Relative error (%)
0.03331	0.04349	0.04379	0.68
0.03026	0.03092	0.03145	1.68
0.03014	0.03028	0.03080	1.68
0.03010	0.02999	0.03059	1.95
0.03004	0.02940	0.03024	2.89
0.02992	0.02866	0.02952	2.90
0.02475	0	0	–

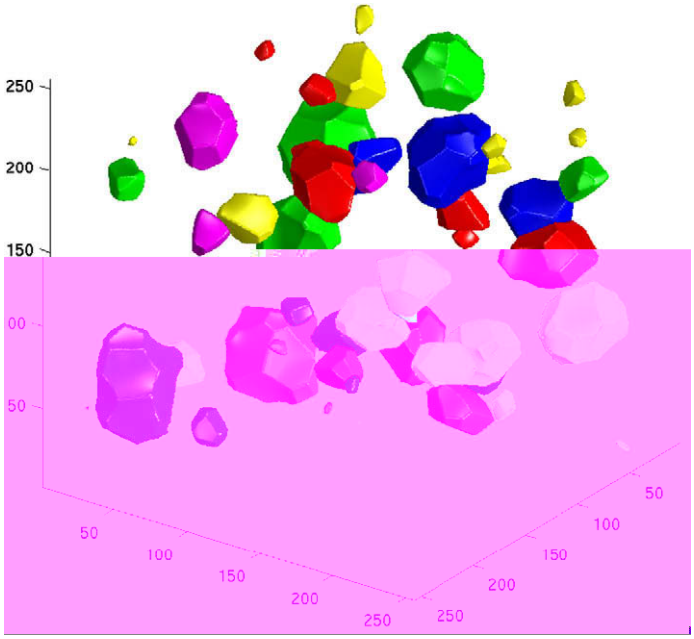


Fig. 12. Grains from five level sets after 500 iterations. 853 of the initial 16,767 grains remain.

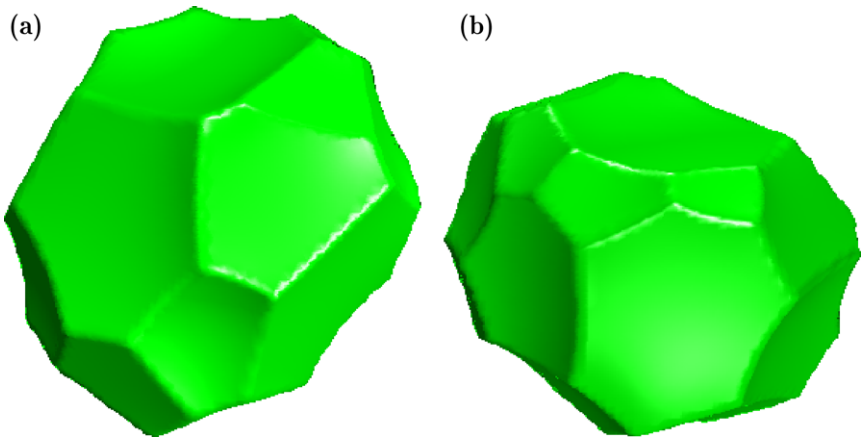


Fig. 13. Two views of a single grain (corresponding to a 180° rotation in the *xy*-plane) chosen from the evolution after 500 iterations. This grain is approximately 40 grid cells across in each direction, slightly larger than average. The grain is very well resolved, with facets, edges, and corners all easily distinguished.

Much of the statistical analysis remains to be done in three dimensions. Statistics on the fully three-dimensional runs are of great interest in materials science as materials are inherently three-dimensional. Furthermore, it is also important to reconcile results from two-dimensional simulations to results from cross-sections of three-dimensional computations.

To this end, we immediately see that the character of cross-sections of three-dimensional computations is markedly different from that of the two-dimensional computations. Fig. 14 shows one such cross-section. Obviously the symmetric (120°) angle condition is not expected to be preserved in cross-sections, and does not appear to be in the figure. More tellingly, Fig. 15 shows the relative area distribution and number of edges distribution for an ensemble of these cross-sections (containing 66,437 two-dimensional grains). In agreement with data from other three-dimensional simulations (e.g. [31,1]), small grains are present in cross-section in much higher frequencies than two-dimensional simulations predict. As mentioned in [1], the greater frequency of small grains in cross-section can be partially attributed to the fact that cross-sectional planes may cut across the ends of grains that are large in three dimensions. Furthermore, the number of edges distribution (shown in Fig. 15) is much flatter and wider than predicted by two-dimensional simulations.

4. Simulations in the presence of bulk energy terms

A simple extension to the model considered in previous sections is the inclusion of bulk energy terms:

$$E = \sum_{k<\ell} (\text{length of } \Gamma_{k\ell}) + \sum_{k} (\text{area of phase } k) e_k.$$

where e_k denotes the bulk energy density for phase k. This gives rise to the following normal velocity

$$v_n(\Gamma_{k\ell}) = \kappa_{k\ell} + e_\ell - e_k, \tag{4}$$

see, for example, [22,34,5]. Note that adjacent phases with equal bulk energy density terms will evolve solely by curvature, as the bulk energy contributions from each phase will cancel.

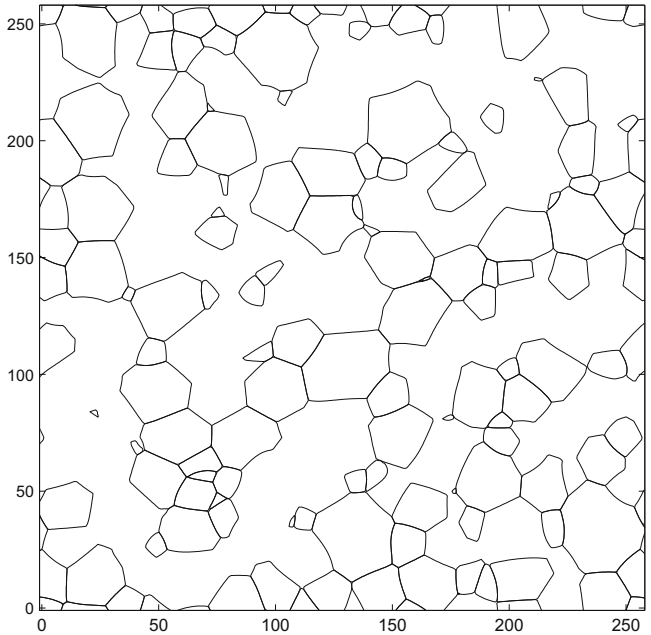
In [4], the authors present a simple modification to the two-phase algorithm to generate motion with normal speeds of the form

$$v_n(\gamma) = \kappa + e.$$

In analogy, our multiphase algorithm changes in only one step. We add an additional term to the convolution step:

$$A_k(x) := K_{\Delta t} * d_k - 2(\Delta t) e_k.$$

This additional set of parameters allows for the simulation of a wider class of motions. For example, an energy and normal speed of this form can be used to simulate the growth of grains on a background medium, by giving the background medium a bulk energy greater than that of the grains. We present two examples of this growth in two dimensions.



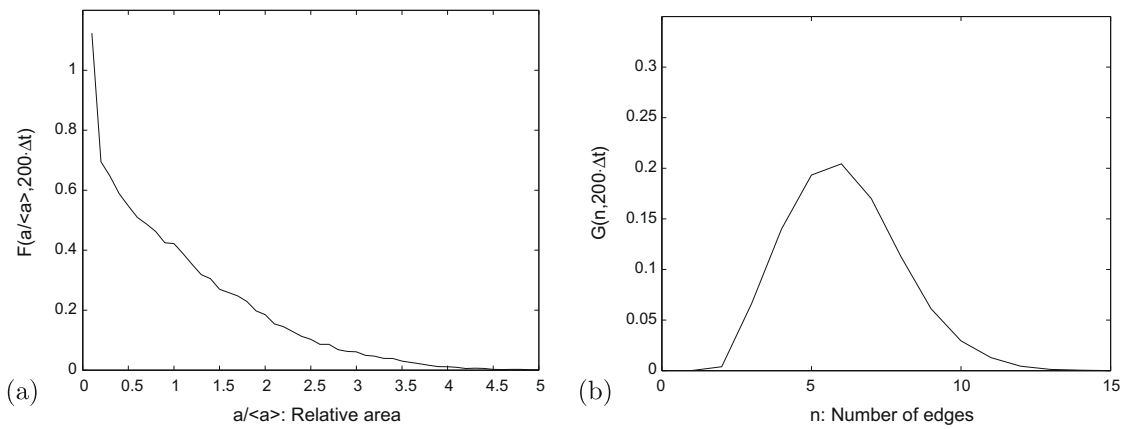


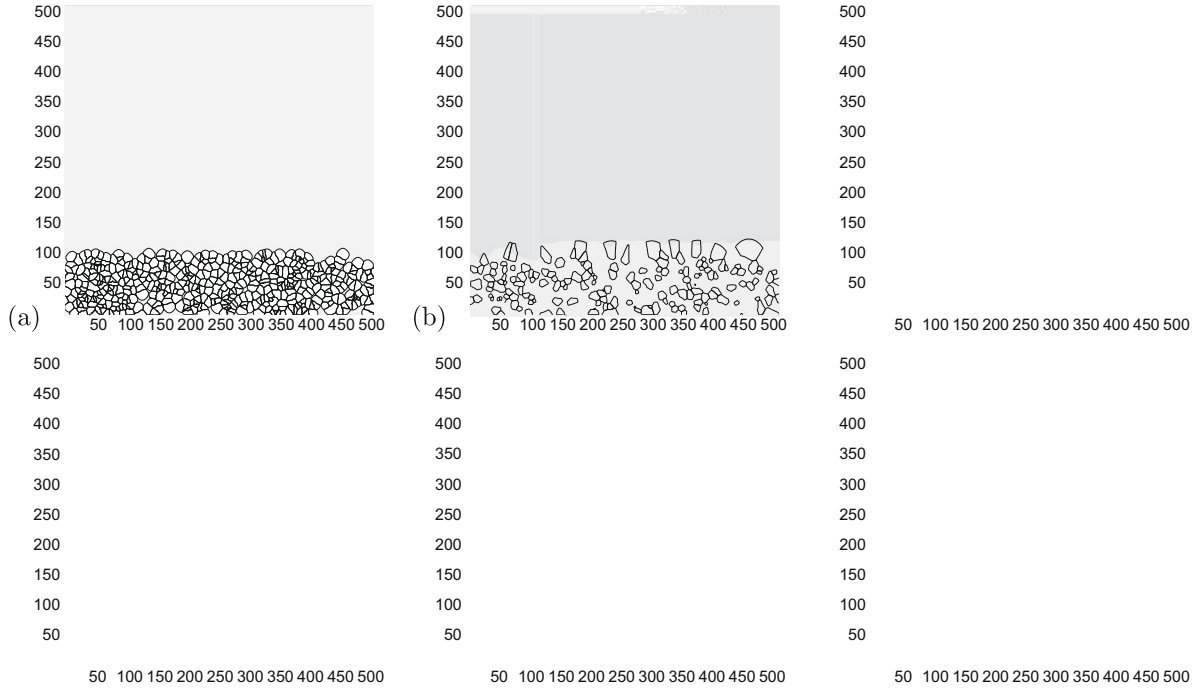
Fig. 15. (a) Relative area distribution and (b) number of edges distribution from two-dimensional cross-sections of three-dimensional evolution. The distribution reflects the increased number of small grains in cross-section of three-dimensional simulations as compared to two-dimensional simulations.

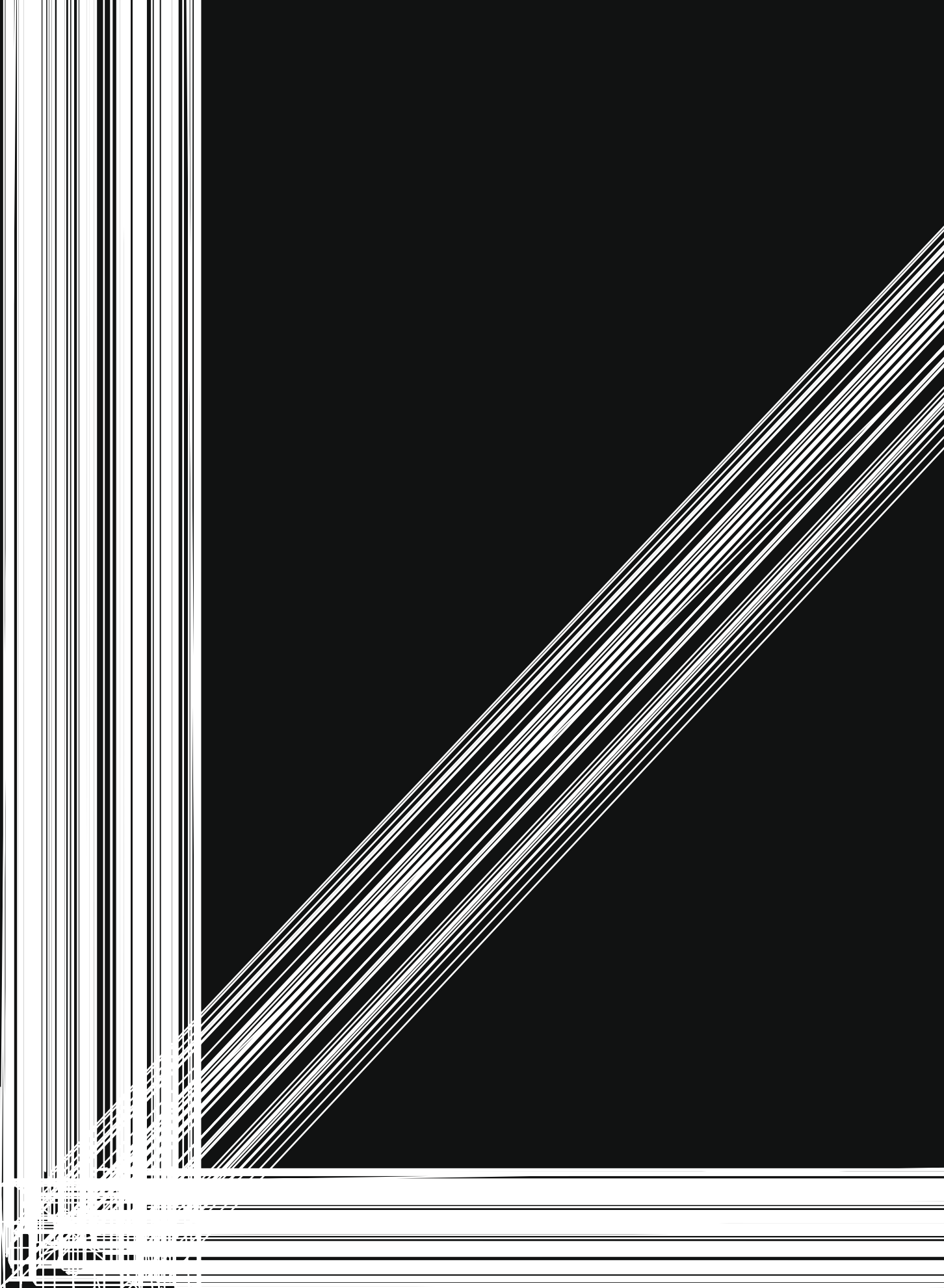
4.1. Example 1

We place many seeds, all with no bulk energy, near one boundary of the unit square and allow them to grow into a background medium with positive bulk energy. A simulation of this growth in two dimensions is shown in Fig. 16. Where two phases meet, the interface moves solely by curvature. When a single grain meets the background phase, the normal velocity also receives a contribution from the bulk energy term. Note that the grains in the lower part of the simulation evolve as in normal grain growth, with approximately equiaxed grains and the mean grain size increasing. In contrast, the grains on the top layer primarily grow vertically, into the background medium. These grains tend to be very large and elongated in comparison to the grains in the lower part of the simulation.

4.2. Example 2

In the second example, we take the initial condition to be 100 circular grains surrounded by another material (e.g. a liquid). We take the bulk energy of the grains to be 0 and the "liquid" to have bulk energy 100/3. The coefficient of surface





In future work, we intend to extend the model to allow for varying surface tensions, so that the normal velocity of an interface is given by

$$v_n(\Gamma_{ij}) = f_{ij}\kappa_{ij} + e_j - e_i,$$

allowing for the simulation of a very general class of motions that are of interest in applications. Further refinement of the numerical techniques used should allow for even larger simulations of three-dimensional grain growth, a phenomenon which is not as well studied as the two-dimensional case. The statistics arising in this case are of great interest. They should be carefully compared to statistics arising from other simulation techniques and from available experimental results.

Acknowledgments

The authors thank David Kinderlehrer for helpful and encouraging conversations. Matt Elsey was supported by NSF DMS-0748333. Selim Esedoglu was supported by NSF DMS-0748333 and an Alfred P. Sloan Foundation fellowship. Peter Smereka was supported by NSF DMS-0810113.

References

[1] M.P. Anderson, G.S. Grest, D.J. Srolovitz, Computer simulation of normal grain growth in three dimensions, Philos. Mag. B 59 (3) (1989) 293–329.
[2] K.A. Brakke, The surface evolver, Exp. Math. 1 (1992) 141–165.
[3] L. Bronsard, F. Reitich, On three-phase boundary motion and the singular limit of vector-valued Ginzburg–Landau equation, Arch. Ration. Mech. Anal. 124 (1993) 355–379.
[4] S. Esedoglu, S. Ruuth, R. Tsai, Diffusion generated motion using the distance function, Tech. Rep., UCLA, 2009. CAM Report 09-29.
[5] S. Esedoglu, P. Smereka, A variational formulation for a level set representation of multiphase flow and area preserving curvature flow, Commun. Math. Sci. 6 (1) (2008) 125–148.
[6] D. Fan, Computer simulation of grain growth using a continuum field model, Acta Mater. 45 (2) (1997) 611–622.
[7] P. Feltham, Grain growth in metals, Acta Metall. 5 (1957) 97–105.
[8] V.E. Fradkov, A.S. Kravchenko, L.S. Shvindlerman, Experimental investigation of normal grain growth in terms of area and topological class, Scripta Metall. 19 (11) (1985) 1291–1296.
[9] V.E. Fradkov, L.S. Shvindlerman, D.G. Udler, Computer simulation of grain growth in two dimensions, Scripta Metall. 19 (11) (1985) 1285–1290.
[10] H. Garcke, B. Nestler, B. Stoth, A multiphase field concept: numerical simulations of moving phase boundaries and multiple junctions, SIAM J. Appl. Math. 60 (1) (1999) 295–315.
[11] M. Hillert, On the theory of normal and abnormal grain growth, Acta Metall. 13 (1965) 227–238.
[12] K. Kawasaki, T. Nagai, K. Nakashima, Vertex models for two-dimensional grain growth, Phil. Mag. B 60 (3) (1989) 399–421.
[13] S.G. Kim, D.I. Kim, W.T. Kim, Y.B. Park, Computer simulations of two-dimension and three-dimensional ideal grain growth, Phys. Rev. E 74 (2006) 061605.
[14] D. Kinderlehrer, J. Lee, I. Livshits, A. Rollett, S. Ta'asan, Mesoscale simulation of grain growth, Mater. Sci. Forum 67–70 (2004) 1057–1062.
[15] D. Kinderlehrer, I. Livshits, S. Ta'asan, A variational approach to modeling and simulation of grain growth, SIAM J. Sci. Comput. 28 (5) (2006) 1694–1715.
[16] C.E. Krill III, L.-Q. Chen, Computer simulation of 3-d grain growth using a phase-field model, Acta Mater. 50 (2002) 3057–3073.
[17] N.P. Louat, On the theory of normal grain growth, Acta Metall. 22 (1974) 721–724.
[18] C. Mantegazza, M. Novaga, V.M. Tortorelli, Motion by curvature of planar networks, Ann. Scuola Norm. Super. Pisa–Cl. Sci. 3 (2) (2004) 235–324.
[19] P. Mascarenhas, Diffusion generated motion by mean curvature, Tech. Rep., UCLA 1992. CAM Report 92-33.
[20] B. Merriman, J.K. Bence, S. Osher, Motion of multiple junctions: a level set approach, J. Comput. Phys. 112 (2) (1994) 334–363.
[21] W. Mullins, Two-dimensional motion of idealized grain boundaries, J. Appl. Phys. 27 (6) (1956) 900–904.
[22] F. Reitich, H.M. Soner, Three-phase boundary motion under constant velocities. I: the vanishing surface tension limit, Proc. R. Soc. Ed. 126A (1996) 837–865.
[23] G. Russo, P. Smereka, A remark on computing distance functions, J. Comput. Phys. 163 (2000) 51–67.
[24] S.J. Ruuth, A diffusion-generated approach to multiphase motion, J. Comput. Phys. 145 (1998) 166–192.
[25] S.J. Ruuth, Efficient algorithms for diffusion-generated motion by mean curvature, J. Comput. Phys. 144 (1998) 603–625.
[26] S.J. Ruuth, B. Merriman, Covolution-thresholding methods for interface motion, J. Comput. Phys. 169 (2001) 678–707.
[27] P.S. Sahni, D.J. Srolovitz, G.S. Grest, M.P. Anderson, S.A. Safran, Kinetics of ordering in two dimensions. II. Quenched systems, Phys. Rev. B 28 (5) (1983) 2705–2717.
[28] P. Smereka, The numerical approximation of a delta function with application to level set methods, J. Comput. Phys 211 (1) (2006) 77–90.
[29] Y. Suwa, Y. Saito, H. Onodera, Parallel computer simulation of three-dimensional grain growth using the multi-phase-field model, Mater. Trans. 49 (4) (2008) 704–709.
[30] Y.H.R. Tsai, L.-T. Cheng, S. Osher, H.-K. Zhao, Fast sweeping algorithms for a class of Hamilton–Jacobi equations, SIAM J. Numer. Anal. 41 (2) (2003) 673–694.
[31] F. Wakai, N. Enomoto, H. Ogawa, Three-dimensional microstructural evolution in ideal grain growth – general statistics, Acta Mater. 48 (2000) 1297–1311.
[32] X. Zhang, J.-S. Chen, S. Osher, A multiple level set method for modeling grain boundary evolution of polycrystalline materials, Tech. Rep., UCLA, 2006. CAM Report 06-69.
[33] H.K. Zhao, A fast sweeping method for Eikonal equations, Math. Comput. 74 250 (2004) 603–627.
[34] H.-K. Zhao, T. Chan, B. Merriman, S. Osher, A variational level set approach to multiphase motion, J. Comput. Phys. 127 (1996) 179–195.