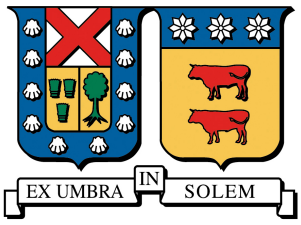
Universidad Técnica Federico Santa María
Departamento de Informática
Valparaíso, Chile



Computational Analysis of a 3D Vertex Model for Grain Growth in Polycrystalline Material

Alejandro Herminio José Sazo Gómez

Tesis para optar al Grado de
Magíster en Ciencias de la Ingeniería Informática

Profesor Guía: Claudio Torres López
Profesor Correferente Interno:
Profesor Correferente Externo:
Presidente Comisión: Marcelo Mendoza

June 3, 2018

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Chapter 2

Two Dimensional Grain Growth

GRAIN growth in polycristalline materials is studied over thin film that is a simplification of real three dimensional grain structures. This structure is composed by grains and boundaries. Each boundary is shared by two grains and the point where three junctions (and also three grains) meet is called a vertex or triple junction. General topology also admits more than three neighbor vertices, however we will only take into account in the models the existence of triple junctions, naming them indistinctly as vertices too. Formally, the mathematical description of a grain structure is defined over the unit squared domain $[0,1]^2 \subset \mathbb{R}^2$ with periodic boundary condition, thus the domain is a torus. The grain structure is composed by a disjoint set of N regions over the whole domain. We denote this set as:

$$\mathbf{\Sigma} = \mathbf{\Sigma}(t) = \left\{\Sigma^{(1)}, \Sigma^{(2)}, \ldots, \Sigma^{(N)}\right\}, \quad N = N(t). \tag{2.1}$$

Note that the number of grains varies over time, which is indicated in the temporal dependence notation. With each grain $\Sigma^{(l)},\ l = 1, \ldots, N$ we associate an orientation $\alpha_l \in [0, 2\pi)$. The grains are limited by their boundaries. The boundary set is defined as:

$$\mathbf{\Gamma} = \mathbf{\Gamma}(t) = \{\Gamma^{(1)}, \Gamma^{(2)}, \ldots, \Gamma^{(K)}\}, \quad K = K(t). \tag{2.2}$$

Again, the number of boundaries in the system also depends on time. The grain misorientation parameter $\Delta\alpha^{(k)}$ is defined as $\Delta\alpha^{(k)} = \alpha_{l_2} - \alpha_{l_1},\ k = 1, \ldots, K$. The grain boundary energy $\gamma^{(k)}$ is assumed to depend only on the grain misorientation parameter, that is $\gamma^{(k)} = \gamma(\Delta\alpha^{(k)}),\ k = 1, \ldots, K$ and some even periodic function $\gamma : \mathbb{R} \to \mathbb{R}$.

Each boundary is parametrized as a curve in the plane defined as:

$$\Gamma^{(k)}(t) = \left\{\boldsymbol{\xi}^{(k)}(s,t), \quad s \in [0,1]\right\}, \quad k = 1, \ldots, K. \tag{2.3}$$

As stated previously, boundaries meet at triple junctions, and these correspond to the start or the end point of exactly three boundaries. We denote the set of triple junctions as:

$$\mathcal{X} = \mathcal{X}(t) = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}, \quad M = M(t). \tag{2.4}$$

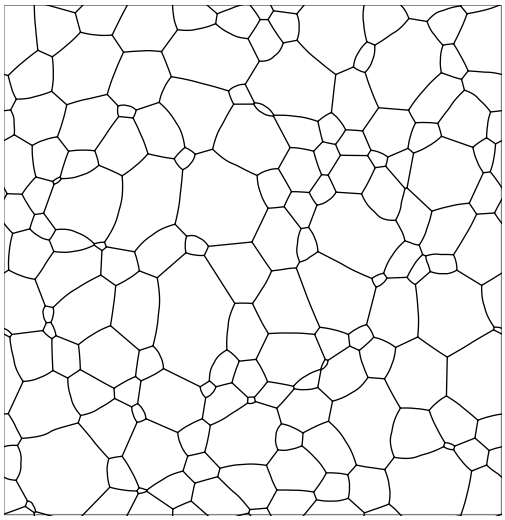


Figure 2.1: Grain structure under periodic boundary conditions.

Let k_1, k_2, k_3 three different boundaries with a common triple junction $\mathbf{x}_m$. For the curve parameters $s_{k_i} = \{0, 1\}$ the triple junctions are defined in function of the boundaries as:

$$\mathbf{x}_m(t) = \boldsymbol{\xi}^{(k_1)}(s_{k_1}, t) = \boldsymbol{\xi}^{(k_2)}(s_{k_2}, t) = \boldsymbol{\xi}^{(k_3)}(s_{k_3}, t), \quad m = 1, \ldots, M.$$

Figure 2.1 shows a simulated grain structure under the given definition. As discussed before, the grain system evolves over time, vertices and boundaries moves, some grains shrink and other grow. When a boundary decreases its length to zero, or when a grain area becomes zero, a component of the system disappears. These changes are denominated topological transitions and are described in the next section.

2.1 Topological Transitions

Topological transitions are the result of coarsening during grain growth and are well documented by Ferro et al. [3] and are dependent on the grain boundary energy, thus some configurations are more probable than others. The most common handled topological transitions [5, 6, 17, 19] in grain growth models are neighbor switching (also known as flipping) and grain removal, as shown in Figures 2.2 and 2.3.

Neighbor switching (Figure 2.2) consist in a grain boundary switching neighboring boundaries. This occurs when a boundary collapses, the vertices approach each other, decreasing the boundary length to zero. This configuration, called a quadruple junction since the four involved boundaries are together, is unstable and breaks in two new vertices and the new boundary grows in a new direction.

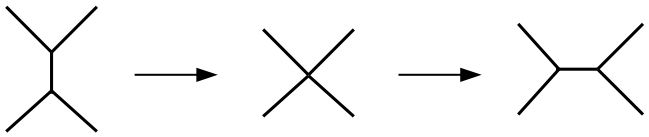


Figure 2.2: Neighbor switching

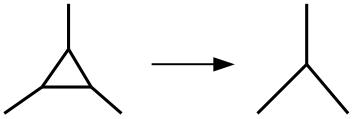


Figure 2.3: Grain removal

Grain removal (Figure 2.3) consists in the removal of a grain which decreased its size to zero. The grain then is replaced by a group of boundaries or a single vertex. There are many results from grain removal that arise from different configurations, but the most important is the disappearing of a three sided grain. A three sided grain is removed when one or more of its boundaries switch neighbors. This will generate a degenerated grain with two sides. Instead of that, a new vertex is created and the boundaries adjacent to the former grain are now joined.

2.2 Topology of Grain Structures

As stated before the grain structure is projected on a torus where each vertex is connected to three other vertices, and thus belongs to three boundaries. The structure has the following Euler characteristic considering the number of vertices M, the number of boundaries K and the number of grains N [12]:

$$M - K + N = 0 \tag{2.5}$$

If we count the number of vertices in a grain structure we count three boundaries per vertex, but as each boundary is shared by two vertices, the boundaries are counted

twice. Using these facts and considering (2.5) we obtain the following relations:

$$3\,M = 2\,K$$
$$N = \frac{K}{3} = \frac{M}{2}$$

Consider an initial condition of a grain structure with N_0 number of grains, M_0 vertices and K_0 boundaries where $N_0 = M_0/2 = K_0/3$. The topological transitions during grain structure evolution will change the number of these components. The only topological transition considered that alter the number of components in a grain structure is grain removal. When a grain is removed, three boundaries and two vertices are removed, leaving a single *survivor* vertex. Now assume that at a given time t we started with N_t grains and then we removed n_t grains, leaving at the next time step N_{t+1} grains. Topological transitions described induces that we removed exactly $3n_t$ boundaries and $2n_t$ vertices and the following definitions yields between vertices, boundaries and grains between time steps t and $t+1$:

$$dN_t = N_{t+1} - N_t = -n_t$$
$$dK_t = K_{t+1} - K_t - 3n_t$$
$$dM_t = M_{t+1} - M_t - 2n_t.$$

We can build the following relation in terms of the variation of grains:

$$3\,dM_t = 2\,dK_t. \tag{2.6}$$

If we add the variations over all the time steps, from time $t = 0$ to $t = T$ we obtain:

$$\begin{aligned}\sum_{t=0}^{T} 3\,dM_t &= 3(M_1 - M_0 + M_2 - M_1 + \cdots + M_{T+1} - M_T) \\ &= 3(M_{T+1} - M_0).\end{aligned}$$

$$\begin{aligned}\sum_{t=0}^{T} 2\,dK_t &= 2(K_1 - K_0 + K_2 - K_1 + \cdots + K_{T+1} - K_T) \\ &= 2(K_{T+1} - K_0).\end{aligned}$$

This total variations for vertices and boundaries are equal according to (2.6). By the fact that the initial condition is true, i.e., $3M_0 = 2K_0$, for a time $t = T$ the number of vertices and boundaries are related as:

$$N_{T+1} = \frac{M_{T+1}}{2}$$
$$N_{T+1} = \frac{K_{T+1}}{3}$$
$$3M_{T+1} = 2K_{T+1}.$$

This implies that grain structure evolution holds the same relation between grains, boundaries and vertices along the simulation.

2.3 Curvature and Vertex Models

The total energy of the grain structure depends on the individual energies along each grain boundary. The total energy thus takes the form:

$$E(t) = \sum_{k=1}^{K} \int_0^1 \gamma^{(k)} \left\| \mathbf{l}^{(k)}(s,t) \right\| \, ds, \tag{2.7}$$

where $\mathbf{l}^{(k)}(s,t) = \dfrac{\partial \boldsymbol{\xi}^{(k)}}{\partial s}(s,t)$ is a tangent vector to $\boldsymbol{\xi}^{(k)}$ and $\|\cdot\|$ is the l^2-norm. Grain system motion equations can be found from this equation by derivating it with respect to time. Computing the derivative of (2.7) we obtain:

$$\begin{aligned} \frac{dE}{dt}(t) &= \sum_{k=1}^{K} \int_0^1 \gamma^{(k)} \frac{\mathbf{l}^{(k)}(s,t)}{\|\mathbf{l}^{(k)}(s,t)\|} \cdot \frac{\partial \mathbf{l}^{(k)}}{\partial t}(s,t) \, ds \\ &= \sum_{k=1}^{K} \int_0^1 \gamma^{(k)} \, \mathbf{T}^{(k)}(s,t) \cdot \frac{\partial \mathbf{v}^{(k)}}{\partial s}(s,t) \, ds, \end{aligned} \tag{2.8}$$

where $\mathbf{T}^{(k)}(s,t)$ is a unit tangent vector of the boundary and $\gamma^{(k)}\mathbf{T}^{(k)}$ it is called capillar force. Note that (2.8) is valid as long as $K'(t) = 0$ within $[t_1, t_2]$, i.e., there are not topological transitions. Integrating by parts (2.8) we obtain:

$$\frac{dE}{dt}(t) = -\sum_{k=1}^{K} \int_0^1 \gamma^{(k)} \frac{\partial \mathbf{T}^{(k)}}{\partial s} \cdot \mathbf{v}^{(k)} \, ds + \sum_{m=1}^{M} \mathbf{v}_m \cdot \sum_{l=1}^{3} \gamma^{(m,l)} \mathbf{T}^{(m,l)}, \tag{2.9}$$

In order to enforce the decreasing energy of the system we must ensure that (2.9) be negative. If we only consider the triple junction drag the boundaries becomes flat, therefore $\dfrac{\partial \mathbf{T}^{(k)}}{\partial s} = 0$ for each boundary. The triple junction velocity is set to decrease energy as:

$$\mathbf{v}_m = -\lambda \gamma^{(m,l)} \mathbf{T}^{(m,l)},$$

where λ is called triple junction mobility. This model is called Vertex model. On the other hand if we only consider the grain boundary influence, we can set the velocity to be proportional to the boundary normal vector, which yields the following expression for boundary velocity:

$$\mathbf{v}^{(k)} = \mu \gamma^{(k)} \frac{\partial \mathbf{T}^{(k)}}{\partial s},$$

where μ is called boundary mobility. Since the motion is curvature-driven,i.e., there is no triple junction influence, this model is called *Curvature model*. A geometrical consequence under this model is the *Herring condition* where the angle between

boundaries, namely dihedral angle, is always $2\pi/3$ [4]. This comes from forcing the geometrical condition over tangent vectors:

$$\sum_{l=1}^{3} \gamma^{(m,l)} \mathbf{T}^{(m,l)} = 0. \tag{2.10}$$

Moreover the rate of growth for a grain becomes independent of its area, being proportional to the number of sides of the grain or grain class $\mathrm{ns}(g)$. This relation stablishes that a grain with $\mathrm{ns}(g) < 6$ will shrink and with $\mathrm{ns}(g) > 6$ will grow, in fact the case of 6-sided grain the rate of growth is zero. This relation is known as *Von Neumann-Mullins relation* [8] and can be expressed as:

$$\frac{dA}{dt}(g) = \frac{\nu}{3}(\mathrm{ns}(g) - 6) \tag{2.11}$$

where ν is a term proportional to grain boundary energies $\gamma^{(k)}$ and mobility μ. Under constant $\gamma^{(k)}$, $\nu \propto \gamma\mu$.

Chapter 3

Closed Boundary Motion

An interesting case of study for understanding mathematical and numerical aspects of grain growth simulation is to understand the growth of a closed boundary which tries to minimize its energy. Let $\boldsymbol{\xi}(s,t) = \langle x(s,t), y(s,t)\rangle$, $s \in [0, 2\pi]$ a closed curve in $\mathbb{R}^2$ such that $\boldsymbol{\xi}(0,t) = \boldsymbol{\xi}(2\pi,t)$, where s is the parametrization variable and t is the time variable. This will be the base definition of a closed boundary. Let $\mathbf{l}(s,t) = \dfrac{\partial \boldsymbol{\xi}}{\partial s}(s,t)$ a tangent vector to the boundary $\boldsymbol{\xi}$ and $\mathbf{T}(s,t) = \dfrac{\mathbf{l}(s,t)}{\|\mathbf{l}(s,t)\|}$ the unit tangent vector in the same direction. Also let $\mathbf{N}(s,t) = \dfrac{\partial \mathbf{T}}{\partial s}(s,t)$ a unit normal vector to $\boldsymbol{\xi}$. From the total energy equation in (2.7), the energy of this boundary becomes a single integral term:

$$E(t) = \int_0^{2\pi} \|\mathbf{l}(s,t)\| \, ds, \tag{3.1}$$

where for simplicity we assumed an isotropic regime, i.e., $\gamma = 1$. Taking the derivative of (3.1) with respect to the time yields:

$$\frac{dE}{dt}(t) = \int_0^{2\pi} \frac{\mathbf{l}(s,t)}{\|\mathbf{l}(s,t)\|} \cdot \frac{\partial \mathbf{l}}{\partial t}(s,t) \, ds. \tag{3.2}$$

Let $\mathbf{v}(s,t) = \dfrac{\partial \boldsymbol{\xi}}{\partial t}(s,t)$ the velocity of the boundary. Equation (3.2) becomes:

$$\frac{dE}{dt}(t) = \int_0^{2\pi} \mathbf{T}(s,t) \cdot \frac{\partial \mathbf{v}}{\partial s}(s,t) \, ds. \tag{3.3}$$

Integrating by parts (3.3) we obtain:

$$\begin{aligned} \frac{dE}{dt}(t) &= \mathbf{T}(s,t) \cdot \mathbf{v}(s,t)\Big|_0^{2\pi} - \int_0^{2\pi} \frac{\partial \mathbf{T}}{\partial s}(s,t) \cdot \mathbf{v}(s,t) \, ds \\ &= -\int_0^{2\pi} \frac{\partial \mathbf{T}}{\partial s}(s,t) \cdot \mathbf{v}(s,t) \, ds. \end{aligned} \tag{3.4}$$

Thus, we will use (3.4) to understand the motion of a closed boundary.

3.1 Curvature based motion

Curvature based grain growth models assumes that the velocity of the boundaries is in the direction of its normal vector and proportional to its curvature $\kappa(s,t)$ [6]. Thus the velocity of the boundary can be defined as [16]:

$$\mathbf{v}(s,t) = \kappa(s,t)\mathbf{N}(s,t).$$

This term can be obtained also from the derivative of the vector $\mathbf{T}$ with respect to the arc length $\mathcal{L}$:

$$\begin{aligned} \frac{\partial \mathbf{T}}{\partial \mathcal{L}}(s,t) &= \kappa(s,t)\mathbf{N}(s,t) \\ \frac{\partial \mathbf{T}}{\partial s}(s,t)\frac{ds}{d\mathcal{L}} &= \kappa(s,t)\mathbf{N}(s,t). \end{aligned} \tag{3.5}$$

Let $\mathcal{L}(s)$ the arc length of the boundary up to s, given by:

$$\mathcal{L}(s) = \int_0^s \|\mathbf{l}(s,t)\|\, ds, \tag{3.6}$$

then we can obtain $\dfrac{ds}{d\mathcal{L}}$ from (3.6) by derivating with respect to s as:

$$\begin{aligned} \frac{d\mathcal{L}}{ds} &= \|\mathbf{l}(s,t)\| \\ \frac{ds}{d\mathcal{L}} &= \frac{1}{\|\mathbf{l}(s,t)\|}. \end{aligned}$$

Replacing in (3.5) we obtain:

$$\frac{1}{\|\mathbf{l}(s,t)\|}\frac{\partial \mathbf{T}}{\partial s}(s,t) = \kappa(s,t)\mathbf{N}(s,t). \tag{3.7}$$

Therefore to obtain curvature based motion we need to compute the grain boundary velocity as:

$$\mathbf{v}(s,t) = \frac{1}{\|\mathbf{l}(s,t)\|}\frac{\partial \mathbf{T}}{\partial s}(s,t). \tag{3.8}$$

3.2 Coupled model based motion

Let's consider the same boundary parametrization as the coupled model from (4.1). We do not need to handle a special treatment of vertices and interior points. Since we are dealing with a periodic condition for this grain boundary, instead of classic Lagrange interpolator functions ϕ_i, we should use another interpolation basis, for

example the periodic sinc interpolator [18]. The velocity of the boundary $\mathbf{v}$ is expressed in terms of the parametrization as:

$$\mathbf{v}(s,t) = \sum_{i=1}^{n} \dot{\mathbf{x}}_i(t)\, \phi_i(s)\,. \tag{3.9}$$

We can replace the velocity term in (3.4) to obtain the derivative of the energy in terms of the boundary.

$$\begin{aligned}\frac{dE}{dt}(t) &= -\int_0^{2\pi} \frac{\partial \mathbf{T}}{\partial s}(s,t) \cdot \left(\sum_{i=1}^{n} \dot{\mathbf{x}}_i(t)\, \phi_i(s) \right) ds \\ &= -\sum_{i=1}^{n} \dot{\mathbf{x}}_i(t) \cdot \int_0^{2\pi} \frac{\partial \mathbf{T}}{\partial s}(s,t)\phi_i(s)\; ds. \end{aligned} \tag{3.10}$$

The velocity at the boundary points $\mathbf{x}_i$ can be defined from (3.10) such that decreases the energy of the grain system as:

$$\dot{\mathbf{x}}_i(t) = \frac{\zeta}{\|\mathbf{l}(s_i,t)\|} \int_0^{2\pi} \frac{\partial \mathbf{T}}{\partial s}(s,t)\phi_i(s)\; ds, \tag{3.11}$$

where we introduced the curvature term from (3.7) for each discrete boundary point and a constant ζ which depends on the specific parametrization and interpolation and will be defined in the next section. This evolution equation contrasts with the curvature based evolution equation found before. Here we have instead of a normal vector at a given point an integral along the boundary. In order to find more agreement with the curvature based equation, we can study a circular boundary which holds simple formulas.

3.2.1 Circular Boundary

This simple case is very illustrative, since we can obtain an explicit formula for $\dfrac{dE}{dt}$ and also the area rate of change $\dfrac{dA}{dt}$. First we describe analytic results for boundary evolution and then we apply the coupled model idea of interpolate the boundary and obtain new equations.

The equation that describes this boundary is:

$$\boldsymbol{\xi}(s,t) = R(t)\langle \cos(s), \sin(s) \rangle, \quad R(t) > 0 \;\; \forall\, t \geq 0, \tag{3.12}$$

where we have decoupled the spatial parametrization in polar coordinates with the radial time-dependence. Notice that $\|\boldsymbol{\xi}(s,t)\| = R(t)$, that is, for a fixed time $t = \tau$,

the boundary preserves a constant radius and thus constant curvature along the boundary. The tangent vector $\mathbf{l}(s,t)$ is nothing but:

$$\mathbf{l}(s,t) = R(t)\langle -\sin(s), \cos(s)\rangle,$$

and again its norm $\|\mathbf{l}(s,t)\| = R(t)$ just like $\boldsymbol{\xi}$. This implies that the unit tangent vector is:

$$\mathbf{T}(s,t) = \langle -\sin(s), \cos(s)\rangle,$$

and the normal vector is:

$$\frac{\partial \mathbf{T}}{\partial s}(s,t) = \langle -\cos(s), -\sin(s)\rangle,$$

which is a vector always pointing to the circle center. If we took these results and plug them in (3.4) yields:

$$\frac{dE}{dt}(t) = \int_0^{2\pi} \langle \cos(s), \sin(s)\rangle \cdot \mathbf{v}(s,t)\, ds.$$

Considering that the boundary moves proportional to the curvature and in a normal direction, we replace the velocity term with the curvature result in (3.7):

$$\begin{aligned}
\frac{dE}{dt}(t) &= \int_0^{2\pi} \langle \cos(s), \sin(s)\rangle \cdot \frac{1}{\|\mathbf{l}(s,t)\|}\langle -\cos(s), -\sin(s)\rangle\, ds \\
&= \int_0^{2\pi} -\frac{1}{R(t)}\, ds \\
&= -2\pi\kappa(t)
\end{aligned} \tag{3.13}$$

Area rate of change can also be obtained from the definition of area given the boundary parametrization in (3.12):

$$\begin{aligned}
A(t) &= \frac{1}{2}\int_0^{2\pi} \|\boldsymbol{\xi}(s,t)\|^2\, ds \\
\frac{dA}{dt}(t) &= \int_0^{2\pi} \boldsymbol{\xi}(s,t) \cdot \mathbf{v}(s,t)\, ds \\
&= \int_0^{2\pi} R(t)\langle \cos(s), \sin(s)\rangle \cdot \frac{1}{R(t)}\langle -\cos(s), -\sin(s)\rangle\, ds \\
&= -2\pi
\end{aligned} \tag{3.14}$$

This means that the rate of change for the area of a circle is constant. The evolution of the radius $R(t)$ over time can be explicitly found by comparing the velocity of the curvature in (3.7) with the circle velocity expression:

$$\frac{dR}{dt}(t) = -\frac{1}{R(t)} \tag{3.15}$$

$$R(t) = \sqrt{R_0^2 - 2t}, \tag{3.16}$$

where $R(0) = R_0$ is the initial radius of the circle at time $t = 0$. The equation is valid as long as $R_0^2 > 2t$. In the limit, the circle should become a single point, and related vectors becomes indeterminate. Finally, the complete description of a circular boundary is given by:

$$\boldsymbol{\xi}(s,t) = \sqrt{R_0^2 - 2t}\,\langle \cos(s), \sin(s)\rangle. \tag{3.17}$$

The coupled model based formula induces a different result from the velocity found in (3.15). Assuming equispaced points in $[0, 2\pi]$, we replace $\mathbf{l}$ and $\dfrac{\partial \mathbf{T}}{\partial s}$ for the circle in (3.11) to obtain:

$$\dot{\mathbf{x}}_i(t) = \frac{1}{R(t)} \int_0^{2\pi} \langle -\cos(s), -\sin(s)\rangle\, \phi_i(s)\, ds, \tag{3.18}$$

where ϕ the periodic sinc interpolation function. If we replace the definition of the periodic sinc we obtain:

$$\dot{\mathbf{x}}_i(t) = -\frac{1}{R(t)} \int_0^{2\pi} \left\langle \cos(s) \frac{\sin(\pi(s-s_i)/h)}{\frac{2\pi}{h}\tan((s-s_i)/2)}, \sin(s) \frac{\sin(\pi(s-s_i)/h)}{\frac{2\pi}{h}\tan((s-s_i)/2)} \right\rangle ds \tag{3.19}$$

$$= -\frac{1}{R(t)} \frac{2\pi}{n} \left\langle \cos\left(\frac{2\pi i}{n}\right), \sin\left(\frac{2\pi i}{n}\right) \right\rangle. \tag{3.20}$$

Figure 3.1 shows the evolution of the boundary with 14 points using the velocity equation from (3.20). The discrete data is extracted from a circle and then is interpolated using the sinc periodic interpolator. For this example we set $R_0 = 2$, which implies that the limit time of the simulation is $t_{lim} = 2$. The theoretical description of the circle matches qualitatively with the interpolated curve, and for simplicity of presentation the theoretical circle is not shown.

Figure 3.2 and 3.3 shows relevant statistics of the experiment. Boundary area is in good agreement with the theoretical area. Area rate of change shows that whilst we should expect exactly $\dfrac{dA}{dt} = -2\pi$, we get some oscillations around this value, increasing in magnitude as simulation reaches t_{lim}. Actually the simulation is stopped at some time before reach t_{lim}, because of the mentioned acceleration of the circular boundary. The grain boundary energy under isotropic regime becomes just the perimeter of the circle $P(t) = 2\pi R(t)$. The measured energy of the simulated circle is a monotonic decreasing function in good agreement with the theoretical perimeter obtained from the radii in (3.16), as well as the rate of change of the energy.

3.2.2 Non-regular Boundary

Most grains and their boundaries are not restricted to hold circular shapes. The following example shows a boundary described by a polar rose with an initial condition:

$$\boldsymbol{\xi}(s,0) = 3 + \cos(3s)$$

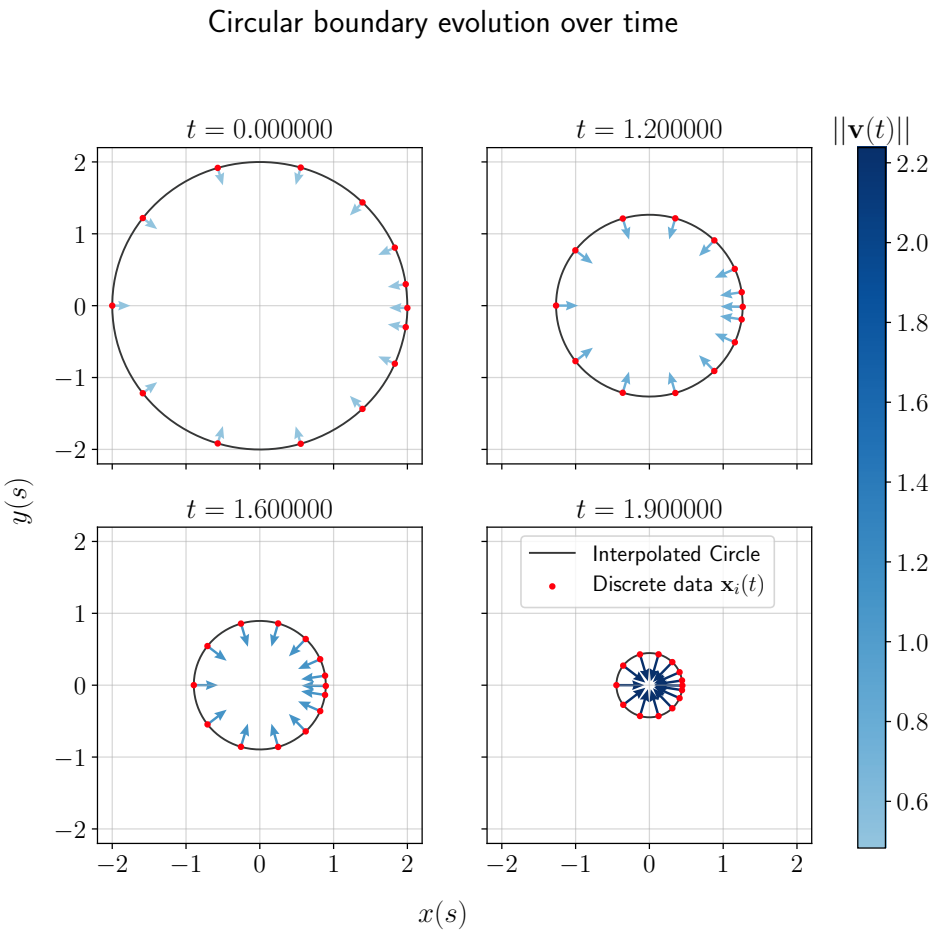
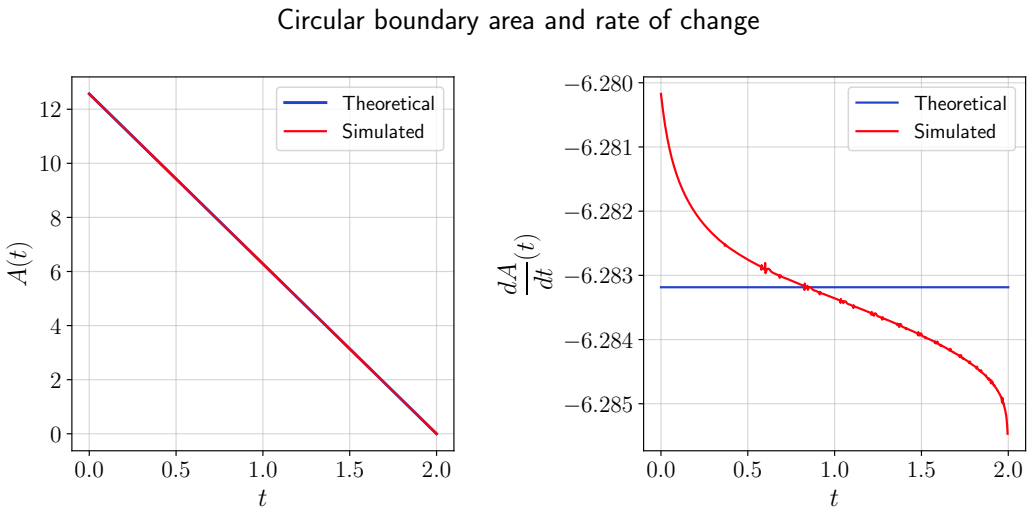


Figure 3.1: (Top left) The initial condition at $t = 0$ of the circular boundary with $R_0 = 2$. (Top right, bottom left) After some time the circle start to shrink and the velocity of the boundary increases. (Bottom right) Near $t = 2$ the circle becomes smaller and the velocity vectors increased their magnitude.

using 14 interpolation points. The number of points is chosen merely to resemble the interpolated function. The simulation is performed using the curvature equations from (3.20).

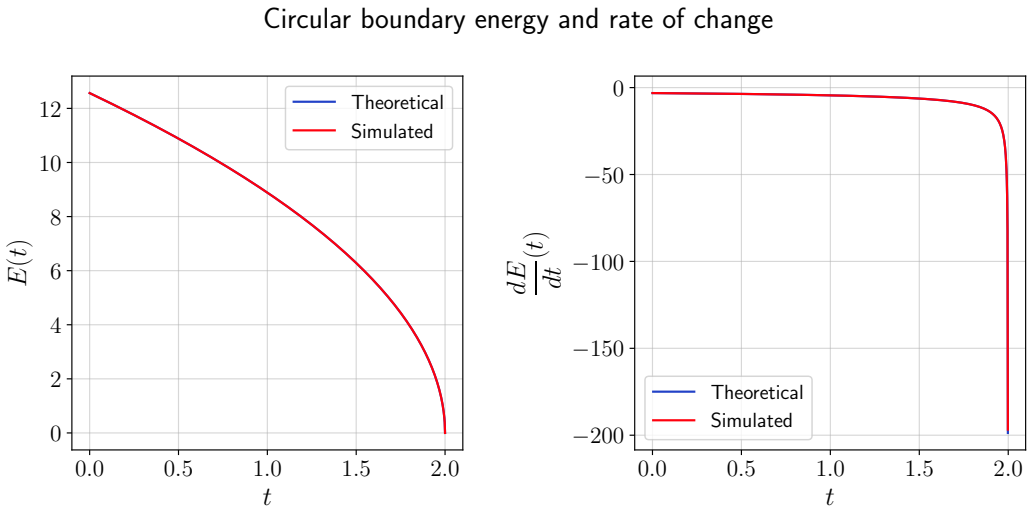
Figure 3.4 shows the evolution of the polar rose. The equations used ensures energy minimization, which induces that the velocity of the discrete points lying in the concave sections points outside the the rose, trying to achieve some iterations later a fully convex form. In the convex form now the points try to approximate a circle shape, which leads to the known evolution presented in Section 3.2.1. Figures 3.5 and 3.6 shows relevant statistics of the experiment. The polar rose area shows qualitatively linear behavior, but a closer look to the derivative shows that there is some perturbation during the minimization, which is soften towards the end of the simulation. The grain boundary energy shows a monotonic decreasing behavior, and



(a) (b)

Figure 3.2: (a) Circle area is in good agreement with the theoretical description. (b) Rate of change is not constant in the numerical simulation, but it is near the theoretical value.

its derivative shows a transient where the derivative becomes less negative due to the polar rose correcting its concave parts. After the rose becomes a convex figure, the derivative becomes more and more negative, until numerical collapse at the end of the evolution.



(a) (b)

Figure 3.3: (a) Grain energy function is in good agreement with the evolution of the perimeter of the circle. (b) Rate of change of the energy, in agreement with the precipitation of the boundary motion as the radio goes to zero.

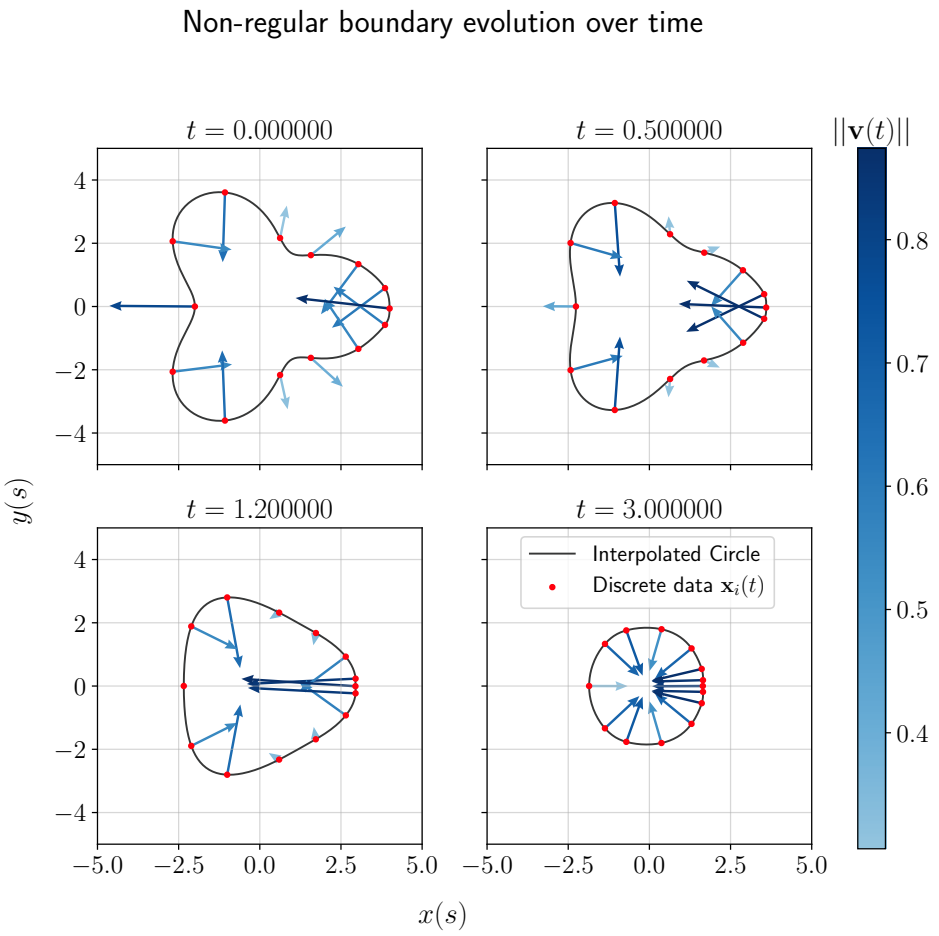
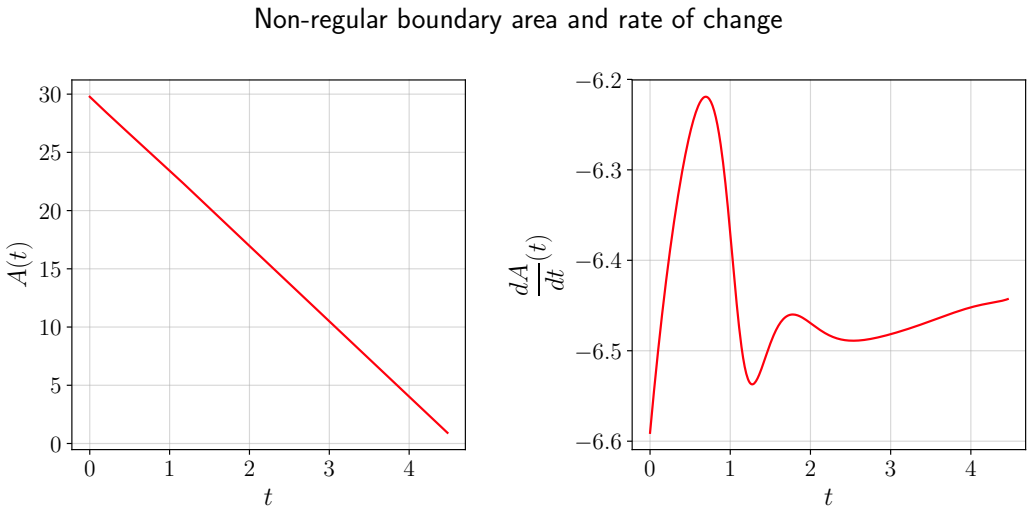
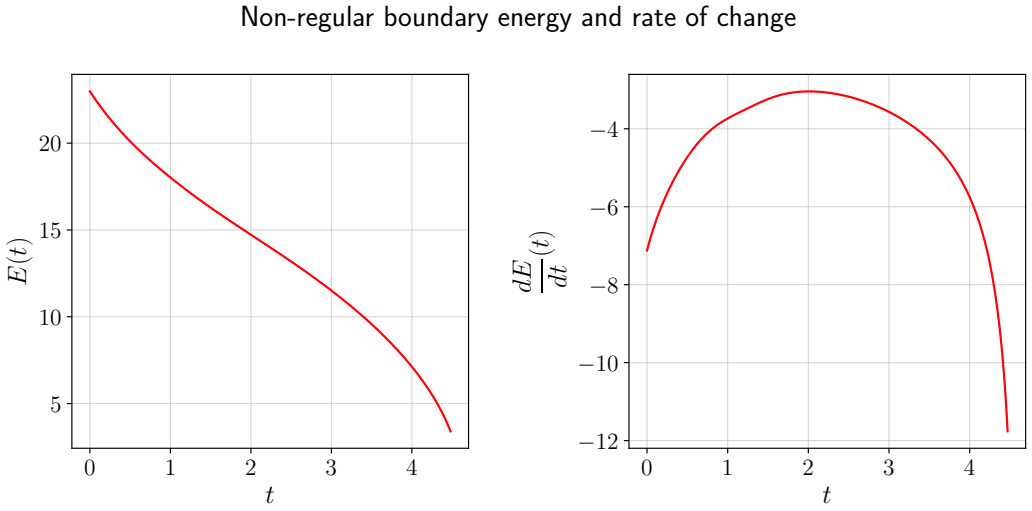


Figure 3.4: (Top left) The initial condition at $t = 0$ for the polar rose. (Top right) After some time the boundary is driven to correct the curvature while trying to minimize energy. (Bottom left) The boundary is completely convex and all the velocity vectors points to the interior. (Bottom right) The polar rose now has a circle-like shape and the velocity vectors tends to point to the center.



(a) (b)

Figure 3.5: (a) Polar rose area shows qualitatively linear behavior, as expected. (b) Rate of change is not constant in the numerical simulation but tries to stabilize.



(a) (b)

Figure 3.6: (a) Grain energy function shows a monotonic decreasing behavior. (b) Rate of change of the energy collapses at the end of the simulation, meaning that the polar rose is becoming a single point.

Chapter 4

The Coupled Model

NUMERICAL simulations studied by vertex-driven motion and curvature-driven motion as exposed in Chapter 2 can be also studied in a coupled way with the purpose of capturing the dynamics of both models and being able to reproduce each one of them individually.

$$\sum_{i=1}^{n} \mathbf{x}_i^{(k)}(t)\, \phi_i(s) \tag{4.1}$$

4.1 Numerical Implementation

An important result of the study of extinction time is that topological transitions are delayed because the extinction time estimation has an error proportional to the simulation Δt. This study suggest that the continuous motion of triple junctions and boundaries must be accurate enough to estimate extinction times of higher order.

The first idea to obtain a better approximation of extinction time is to perform r steps of size $\Delta\tau$ such that $\Delta\tau = \dfrac{\Delta t}{r}$. During the execution of this steps, the basic assumption is that no topological transition occurs and thus the system just evolves. Using a smaller time step helps to ensure a better approximation. On the other hand, straightforward higher order methods can be used. For example second order Runge-Kutta ensures that the grain system evolves with precision $\mathcal{O}(\Delta t^2)$, and therefore the extinction time has the same precision. Runge-Kutta can also be improved by the introduction of many steps of smaller size. In both methods, topological transitions that might occur in $[t, t+\Delta t]$ are detected and managed after the evolution is performed. Both methods are discussed below.

4.1.1 Multistep Euler

This method assumes that Euler method can be improved if steps of smaller size are performed to evolve the grain structure previous to handle topological transitions.

A number r of smaller steps of size $\Delta\tau$ is chosen, where $\Delta\tau = \Delta t/r$. The execution of this r steps is called the *multistep stage* and the system evolves assuming that there are not topological transitions. Algorithm 4.1 shows this method.

Algorithm 4.1 Multistep Euler for Coupled Model

```
1: procedure ME
2:     Δτ ← Δt/r  Time step of multistep phase.
3:     for k : 1, ... , r do
4:         V_t ← Compute velocities
5:         X_{t+Δτ} ← X_t + Δτ V_t
6:         t ← t + Δτ
7:     end for
8: end procedure
```

Using a smaller step-size helps to ensure a better approximation of the extinction time proportional to $\mathcal{O}(\Delta\tau)$. Inside the multistep stage, topological transitions are detected by computing the extinction time of the boundaries and comparing it with Δt but they are not performed. When the multistep stage is over all the topological transitions found are filtered so there are no inconsistencies and then are performed safely. No extra memory is required for this method, but the cost if purely computational since the method is performing r steps per main step and thus is almost r times slower. Notice that when $r = 1$, we recover the original Forward Euler method.

4.1.2 Multistep Second Order Runge-Kutta

If the goal is to obtain a good approximation of the extinction time, a higher order method can be implemented straightforward. For example extinction times with precision $\mathcal{O}(\Delta t^2)$ can be estimated using second order Runge-Kutta (RK2). We can also improve this method by means of introducing the multistep idea to perform several RK2 steps within $[t, t + \Delta t]$ as shown in Algorithm 4.2.

The cost of this method lies in the memory needed to store the extra data for performing the two estimations at time $\Delta\tau/2$ and $\Delta\tau$ and the number of steps r. Notice that when $r = 1$ we recover the original RK2. In this implementation it is only necessary to backup the positions of the vertices and interior points to be used in the last step of the method and not the whole data structure i.e., arc lengths, curvatures, grain areas, etc.

Algorithm 4.2 Multistep Second Order Runge-Kutta for Coupled Model

procedure MRK2
  $\Delta\tau \leftarrow \dfrac{\Delta t}{r}$ Time step of multistep phase.
  for $k : 1, \ldots, r$ **do**
    $\overline{\mathbf{X}}_t \leftarrow$ Backup positions $\mathbf{X}_t$ of triple junctions and interior points
    $\mathbf{V}_t \leftarrow$ Compute velocities
    $\mathbf{X}_{t+\Delta\tau/2} \leftarrow \mathbf{X}_t + \dfrac{\Delta\tau}{2}\mathbf{V}_t$. Evolve structure for first RK estimation
    $\mathbf{V}_{t+\Delta\tau/2} \leftarrow$ Compute velocities
    $\mathbf{X}_{t+\Delta\tau} \leftarrow \overline{\mathbf{X}}_t + \Delta\tau\mathbf{V}_{t+\Delta\tau/2}$. Evolve structure for second RK estimation
    $t \leftarrow t + \Delta\tau$
  end for
end procedure

Chapter 5

The Continuous Stored Energy Vertex Model

EXTENSION of the classic vertex model considers the introduction of an intragranular stored energy $\mathcal{E}$ which plays a key role in primary recrystallization [9, 10]. The local energy of a vertex i considering the stored energy term can be defined as:

$$E_i(t) = \sum_{j \in \mathcal{N}_i} \left(\gamma_{i,j} \mathcal{L}_{i,j}(t) + \mathcal{E}_{i,j} A_{i,j}(t) \right), \tag{5.1}$$

where $\mathcal{N}_i$ is the set of neighbor vertices to the vertex i, $\mathcal{L}_{i,j}$ is the arc length of the boundary formed by vertices i and j, $\mathcal{E}_{i,j}$ and $A_{i,j}$ is the stored energy and area of a grain adjacent to the boundary i, j using the right-hand rule. We split the local energy between the grain boundary contribution and the grain area contribution, so:

$$E_i(t) = \sum_{j \in \mathcal{N}_i} \gamma_{i,j} \mathcal{L}_{i,j}(t) + \sum_{g \in \mathcal{G}_i} \mathcal{E}_g A_g(t), \tag{5.2}$$

where $\mathcal{G}_i$ is the set of grains associated to vertex i. Now, we could define the total energy of the system as:

$$E(t) = \sum_{k \in \mathbf{\Gamma}} \gamma_k \mathcal{L}_k(t) + \sum_{g \in \mathcal{G}} \mathcal{E}_g A_g(t), \tag{5.3}$$

where $\mathbf{\Gamma}$ is the set of all the boundaries in the grain structure and $\mathcal{G} = \bigcup_i \mathcal{G}_i$ is the set of grains. So, now we could decompose the contribution from each vertex to the total energy as follows,

$$\widehat{E}_i(t) = \sum_{j \in \mathcal{N}_i} \gamma_{i,j} \frac{\mathcal{L}_{i,j}(t)}{2} + \sum_{g \in \mathcal{G}_i} \mathcal{E}_g \frac{A_g(t)}{\mathrm{ns}(g)}, \tag{5.4}$$

where $\mathrm{ns}(g)$ is the number of sides (or class) of grain g. This allows us to build the total energy as the sum of the contribution of each vertex such that we ensure we

only consider once each grain boundary contribution and also consider once each grain contribution, thus,

$$\begin{aligned}\sum_{i\in\mathcal{N}}\widehat{E}_i(t) &= \sum_{i\in\mathcal{N}}\left(\sum_{j\in\mathcal{N}_i}\gamma_{i,j}\frac{\mathcal{L}_{i,j}(t)}{2}+\sum_{g\in\mathcal{G}_i}\mathcal{E}_g\frac{A_g(t)}{\mathrm{ns}(g)}\right)\\ &= \underbrace{\sum_{i\in\mathcal{N}}\left(\sum_{j\in\mathcal{N}_i}\gamma_{i,j}\frac{\mathcal{L}_{i,j}(t)}{2}\right)}_{(a)}+\underbrace{\sum_{i\in\mathcal{N}}\left(\sum_{g\in\mathcal{G}_i}\mathcal{E}_g\frac{A_g(t)}{\mathrm{ns}(g)}\right)}_{(b)}\end{aligned}$$

In order to solve (a) we need to understand which elements are being counted. The outer sum takes all the vertices and the inner sum takes the neighbor vertices of each vertex. Consider the expanded terms for some vertex k:

$$\gamma_{k,j_1}\frac{\mathcal{L}_{k,j_1}(t)}{2}+\gamma_{k,j_2}\frac{\mathcal{L}_{k,j_2}(t)}{2}+\gamma_{k,j_3}\frac{\mathcal{L}_{k,j_3}(t)}{2},$$

where j_1, j_2, j_3 are the three neighbor vertices of vertex k. Let's see now the terms associated to one of these vertices, for example j_1:

$$\gamma_{j_1,k}\frac{\mathcal{L}_{j_1,k}(t)}{2}+\gamma_{j_1,j_{12}}\frac{\mathcal{L}_{j_1,j_{12}}(t)}{2}+\gamma_{j_1,j_{13}}\frac{\mathcal{L}_{j_1,j_{13}}(t)}{2}.$$

It is clear that for each vertex we are counting the energy associated to each boundary twice, each per vertex in a boundary. This justifies the introduction of the term $1/2$ in $\widehat{E}_i(t)$. The analysis to solve (b) is similar. The inner sum of here runs over the grains related to vertex i. We can build a similar example starting from a vertex k and counting the three stored energy terms related as:

$$\mathcal{E}_{g_1}\frac{A_{g_1}(t)}{\mathrm{ns}(g_1)}+\mathcal{E}_{g_2}\frac{A_{g_2}(t)}{\mathrm{ns}(g_2)}+\mathcal{E}_{g_3}\frac{A_{g_3}(t)}{\mathrm{ns}(g_3)}$$

Consider now that g_1 has $\mathrm{ns}(g_1)$ vertices. We already know that k is a vertex of g_1, the terms related to the $\mathrm{ns}(g_1)-1$ remaining vertices must have a term $\mathcal{E}_{g_1}\dfrac{A_{g_1}(t)}{\mathrm{ns}(g_1)}$ and thus we are summing in (b) the same term $\mathrm{ns}(g_1)$ times. Of course, for any grain g the times the related stored energy term is being added $\mathrm{ns}(g)$ times. This justifies the introduction of the term $1/\mathrm{ns}(g)$. Finally we can map the sum in order to count boundaries and grains. Each term related to a boundary is counted twice and each term related to a grain is counted $\mathrm{ns}(g)$ times and we can recover an expression for the total energy of the system as:

$$\begin{aligned}\sum_{i\in\mathcal{N}} \widehat{E}_i(t) &= \sum_{k\in\mathbf{\Gamma}} 2\,\gamma_k \frac{\mathcal{L}_k(t)}{2} + \sum_{g\in\mathcal{G}} \mathrm{ns}(g)\,\mathcal{E}_g \frac{A_g(t)}{\mathrm{ns}(g)} \\ &= \sum_{k\in\mathbf{\Gamma}} \gamma_k \mathcal{L}_k(t) + \sum_{g\in\mathcal{G}} \mathcal{E}_g A_g(t) \\ &= E(t).\end{aligned}$$

Notice that $\widehat{E}_i(t)$ is different from $E_i(t)$.

The boundary arc length $\mathcal{L}$ and grain area A can be seen as function of the vertices positions $\{\mathbf{x}_i\}$. Rewriting (5.4) marking this dependence as:

$$\widehat{E}_i(\mathbf{x}(t)) = \sum_{j\in\mathcal{N}_i} \gamma_{i,j} \frac{\mathcal{L}_{i,j}(\mathbf{x})}{2} + \sum_{g\in\mathcal{G}_i} \mathcal{E}_g \frac{A_g(\mathbf{x})}{\mathrm{ns}(g)}, \tag{5.5}$$

The evolution of the system, considering the stored energy term, such that it decreases energy is obtain by a gradient descent method, so $\dot{\mathbf{x}}_i(t) = -\dfrac{\partial E}{\partial \mathbf{x}_i}$ for each vertex.

5.1 Implementation

Computing the velocity of each vertex is actually the computation of the gradient $\nabla_{\mathbf{x}} E$. Here we propose a matrix-free approach to approximate the gradient. The main advantage of this approach is that it only needs the implementation of computation of the energy of the system, this simplifies enormously the bookkeeping we would need to handle this task.

A possible approximation to compute $\dfrac{\partial E(\mathbf{X})}{\partial \mathbf{x}_i}$ is to perform a matrix free derivation. Consider the vector $\mathbf{X} \in \mathbb{R}^{2n}$ of stacked components of $\mathbf{x}_i = (x_i, y_i)$, this is $\mathbf{X} = (x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_n)^T$ where n is the total number of triple junctions. Each k-th component of the gradient for this vector, say $\dfrac{\partial E(\mathbf{X})}{\partial \mathbf{X}_k}$ is approximated by:

$$\frac{\partial E(\mathbf{X})}{\partial \mathbf{X}_k} \approx \frac{E(\mathbf{X} + \varepsilon\,\mathbf{e}_k) - E(\mathbf{X})}{\varepsilon}, \tag{5.6}$$

where $\mathbf{e}_k$ is the k-th canonical vector in $\mathbb{R}^{2n}$. The evaluation of this approximation implies that the energy of the system must be recomputed twice for each vertex, and according to (5.3) this means to recompute each individual vertex energy each time. Fortunately, this can improved.

This quadratic cost required for a naive implementation is not real in practice. A more detailed analysis shows that the perturbation only affects a x or y component

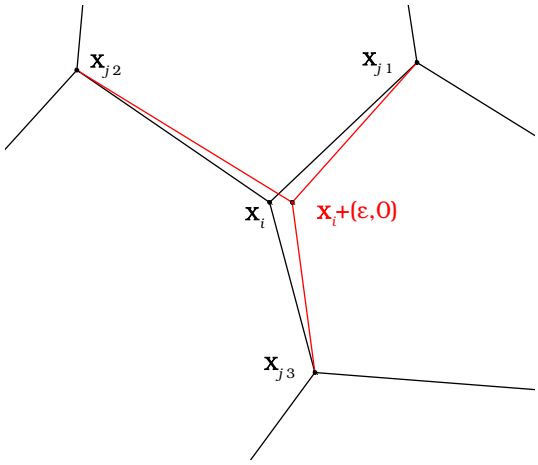


Figure 5.1: Scheme of vertex modifying surrounding grains and boundaries. Arc lengths and grain areas are modified.

of a vertex $\mathbf{x}_i$, which implies that only the three related grain areas and the three boundary arc-lengths changes and thus the energy changes only for the vertex k and its three neighbors as shown in Figure 5.1.

So, instead of recompute the n vertex energies for each one of the $2n$ components of the gradient, we can check directly the difference of energies given by the perturbation. Expanding the difference of energies from (5.6) and taking in account that the perturbation is local, we obtain:

$$E(\mathbf{X}+\varepsilon\,\mathbf{e}_k)-E(\mathbf{X})=\widehat{E}_i(\mathbf{X}+\varepsilon\,\mathbf{e}_k)-\widehat{E}_i(\mathbf{X})+\sum_{j\in\mathcal{N}_i}\widehat{E}_j(\mathbf{X}+\varepsilon\,\mathbf{e}_k)-\widehat{E}_j(\mathbf{X}) \quad (5.7)$$

The energy difference holds some common terms. For example $\widehat{E}_i(\mathbf{X}+\varepsilon\,\mathbf{e}_k)$ has the modified arc-lengths and areas considering that $\mathbf{x}_i$ moved but the stored energies and boundary energies remains constant in this time-step, thus we can use (5.5) and factorize $\widehat{E}_i(\mathbf{X}+\varepsilon\,\mathbf{e}_k)-\widehat{E}_i(\mathbf{X})$ by these constants.

$$\widehat{E}_i(\mathbf{X}+\varepsilon\,\mathbf{e}_k)-\widehat{E}_i(\mathbf{X})=\sum_{j\in\mathcal{N}_i}\gamma_{i,j}\frac{(\tilde{\mathcal{L}}_{i,j}-\mathcal{L}_{i,j})}{2}+\sum_{g\in\mathcal{G}_i}\mathcal{E}_g\frac{(\tilde{A}_g-A_g)}{\mathrm{ns}(g)},$$

where $\tilde{\mathcal{L}}_{i,j}$ is the perturbed arc length for vertices i and j and $\tilde{A}_g$ is the perturbed area of the grain g. The other energies terms related to neighbor vertices can be analyzed as follows. The difference $\widehat{E}_{j_1}(\mathbf{X}+\varepsilon\,\mathbf{e}_k)-\widehat{E}_{j_1}(\mathbf{X})$ resides in the contribution of the modified areas $\tilde{A}_{g_1},\tilde{A}_{g_2}$ and the arc length $\tilde{\mathcal{L}}_{i,j_1}$. The rest of the terms remains the same in each energy term and thus are canceled, therefore:

$$\widehat{E}_{j_1}(\mathbf{X}+\varepsilon\,\mathbf{e}_k)-\widehat{E}_{j_1}(\mathbf{X})=\gamma_{i,j_1}\frac{(\tilde{\mathcal{L}}_{i,j_1}-\mathcal{L}_{i,j_1})}{2}+\mathcal{E}_{g_1}\frac{(\tilde{A}_{g_1}-A_{g_1})}{\mathrm{ns}(g_1)}+\mathcal{E}_{g_2}\frac{(\tilde{A}_{g_2}-A_{g_2})}{\mathrm{ns}(g_2)} \quad (5.8)$$

The remaining energy differences for each neighboring vertex can be obtained analogously:

$$\widehat{E}_{j_2}(\mathbf{X}+\varepsilon\,\mathbf{e}_k)-\widehat{E}_{j_2}(\mathbf{X})=\gamma_{i,j_2}\frac{(\tilde{\mathcal{L}}_{i,j_2}-\mathcal{L}_{i,j_2})}{2}+\mathcal{E}_{g_2}\frac{(\tilde{A}_{g_2}-A_{g_2})}{\mathrm{ns}(g_2)}+\mathcal{E}_{g_3}\frac{(\tilde{A}_{g_3}-A_{g_3})}{\mathrm{ns}(g_3)} \tag{5.9}$$

$$\widehat{E}_{j_3}(\mathbf{X}+\varepsilon\,\mathbf{e}_k)-\widehat{E}_{j_3}(\mathbf{X})=\gamma_{i,j_3}\frac{(\tilde{\mathcal{L}}_{i,j_3}-\mathcal{L}_{i,j_3})}{2}+\mathcal{E}_{g_3}\frac{(\tilde{A}_{g_3}-A_{g_3})}{\mathrm{ns}(g_3)}+\mathcal{E}_{g_1}\frac{(\tilde{A}_{g_1}-A_{g_1})}{\mathrm{ns}(g_1)} \tag{5.10}$$

Notice the common terms in all the energy differences. Adding them up we obtain that:

$$\begin{aligned}\widehat{E}(\mathbf{X}+\varepsilon\,\mathbf{e}_k)-\widehat{E}(\mathbf{X})=&\,\gamma_{i,j_1}\Delta\mathcal{L}_{i,j_1}+\gamma_{i,j_2}\Delta\mathcal{L}_{i,j_2}+\gamma_{i,j_3}\Delta\mathcal{L}_{i,j_3}\\&+3\left(\mathcal{E}_{g_1}\frac{\Delta A_{g_1}}{\mathrm{ns}(g_1)}+\mathcal{E}_{g_2}\frac{\Delta A_{g_2}}{\mathrm{ns}(g_2)}+\mathcal{E}_{g_3}\frac{\Delta A_{g_3}}{\mathrm{ns}(g_3)}\right).\end{aligned} \tag{5.11}$$

Therefore we just need to compute the new arc lengths and areas and their differences.

Moreover, each estimation of the gradient can be computed in parallel maintaining the local information of temporal areas and arc lengths without overwriting data.

Chapter 6

Parallel Management of Topological Transitions

As discussed in Chapter 2, topological transitions modify the grain structure and data structure must reflect this relations at every time step. The update of this structure is critical, due to the consistency during numerical simulation and implementation performance. We discuss here the sequential management algorithm of topological transitions in a grain growth numerical simulation and then we propose a parallel management algorithm that aims to handle a great number of grains.

6.1 Sequential Management

The Coupled Model and Stored Energy Model developed in Chapter 4 and 5 both relies on the linked data structure of vertices linked to three grain boundaries, and grain boundaries linked to two vertices. In a sequential implementation of these models the transitions can be detected by estimating the time that a boundary will collapse, named extinction time or t_{ext}. If the extinction time lies in the current time step $[t, t + \Delta t]$, the boundary is considered to flip and we say this boundary is a candidate. A simple algorithm to solve the transitions within the time step is to sort the candidate boundaries by their extinction time in decreasing order. We can iteratively solve the conflicts by labeling those candidates involved in transitions of other candidates, the labeled boundaries are inhibited to flip this time step. Once all the conflicts has been solved the transitions are performed. Algorithm 6.1 summarizes the described management.

Due to the need of more significant statistics, hundreds of thousands of grains must be simulated. A sequential approach might be inefficient, and parallel implementation of the presented models is proposed. The parallel implementation is straightforward when we speak of evolving the data structure, but the problem of solve the topological transitions arise since the mechanism to solve inconsistencies is sequential and parallelism over data structure will generate race conditions.

Algorithm 6.1 Sequential Management of Topological Transitions

```
1: procedure SEQUENTIAL
2:     Γ ← Clear flip and inhibited state for each boundary
3:     Γ ← Update t_ext for each boundary
4:     Γ_flip ← Boundaries to flip with t_ext ∈ [0, Δt]
5:     Γ_flip ← Sort boundaries by t_ext in decreasing order
6:     Γ_tmp ← ∅,    Empty list of boundaries visited
7:     X_tmp ← ∅,    Empty list of vertices visited
8:     for each Γ ∈ Γ_flip do
9:         {x_i, x_j} ← Vertices of Γ
10:        if X_tmp ∩ {x_i, x_j} = ∅ then
11:            Γ_tmp ← Γ_tmp ∪ Γ
12:            X_tmp ← X_tmp ∪ {x_i, x_j}
13:        else
14:            return Γ_tmp
15:        end if
16:    end for
17: end procedure
```

6.2 Parallel Polling System

Since the concurrent access to data structure in a parallel implementation makes the sequential approach of managing topological transitions a bottleneck, a management system is proposed based on local information given by vertices and boundaries. First we need to compute the extinction time for each boundary and label the candidates for flipping. Instead of sorting the boundaries as the sequential algorithm, each vertex will store local information on which boundary has the lowest extinction time. We say that each vertex *voted* for a boundary. Next, each boundary counts the obtained votes by referring to its vertices. The vote counting is meaningful, two votes means that a boundary can flip since all their neighbor boundaries have not collided yet, and earns the right to label neighbor boundaries to not flip this time step. Other boundaries will obtain one or zero votes. This polling is repeated until no further labeling is possible, that is, when the previous set of candidates is the same as the next iteration.

Figure 6.1 shows a brief example of the polling system. Consider a neighborhood of boundaries with their extinction times already computed. Figure 6.1a shows the vertex poll. Each vertex in this neighborhood will vote for their boundary with lowest t_{ext}. We can see that some boundaries obtained zero, one or two votes. The case of a boundary with two votes is special because implies that two vertices (pink and green) decided that their shared boundary is a good candidate and therefore no other boundary in the immediate neighborhood is a flipping candidate where they

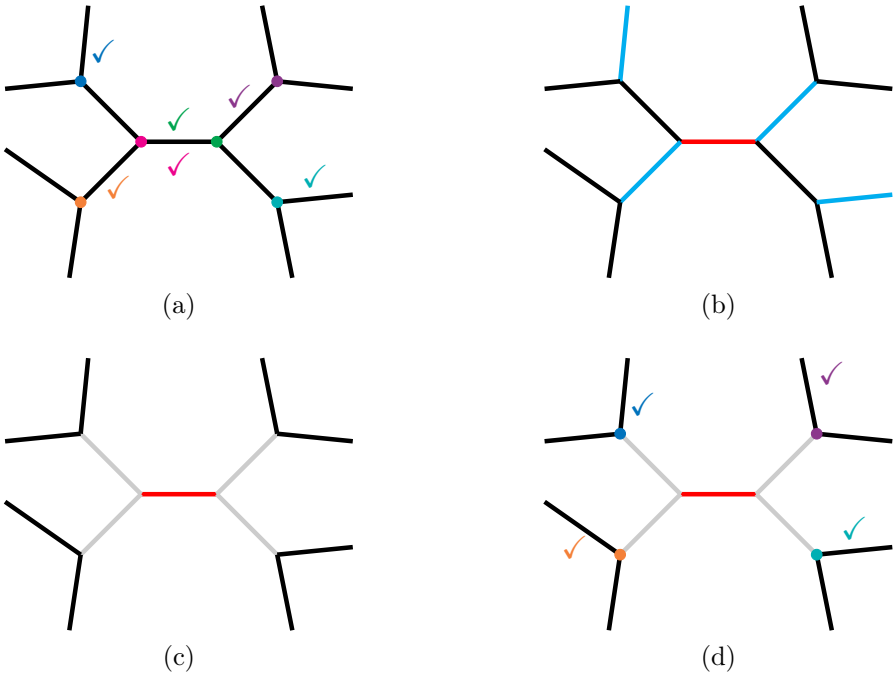


Figure 6.1: Polling system scheme.

obtained at most one vote. Figure 6.1b shows the final counting for each boundary. Boundaries with zero votes remain in black, boundaries with one votes are cyan and boundaries with two votes are red. Notice that although cyan boundaries might be able to flip, the priority is assigned to boundaries with two votes and they label their neighbor boundaries as inhibited as shown in Figure 6.1c. Inhibited boundaries, in gray, can't be voted at next iteration. Related vertices must vote between the remaining non-inhibited boundaries, as shown in Figure 6.1d.

This procedure can be seen as a fixed point iteration over a polling function $\text{Poll}(\mathcal{S})$, where the input of the system $\mathcal{S}$ is the inhibited state of each boundary. When the polling function, given an inhibited state, returns the same inhibited state, we have found a fixed point of the function and thus we converged to a feasible solution. Algorithm 6.3 summarizes this idea.

The advantage of this system is that the vertices votes and boundaries counting can be performed in parallel since data is accessed as read-only. No race condition exists here since we avoided possible near flippings. One disadvantage is the computational cost of perform the polling iteratively until convergence, but the number of topological transitions over time tend to decrease, thus this algorithm is in practice fast enough.

Algorithm 6.2 Polling Routine for Inhibit Boundaries

procedure POLL($\mathcal{S}$)
    $\mathbf{\Gamma}_{\text{flip}} \leftarrow$ Clear boundaries count to 0 and vertices votes
    $\mathbf{\Gamma}_{\text{flip}} \leftarrow$ Vertices votes for their uninhibited boundary with lowest t_{ext}
    $\mathbf{\Gamma}_{\text{flip}} \leftarrow$ Boundaries counts the votes received
    $\mathbf{\Gamma}_{\text{flip}} \leftarrow$ Select uninhibited boundaries with two votes
    $\mathbf{\Gamma}_{\text{flip}} \leftarrow$ Inhibit neighbor boundaries
    return $\mathcal{S}$ ▷ New inhibited state of boundaries
end procedure

Algorithm 6.3 Parallel Polling System for Managing Topological Transitions

procedure POLLING_SYSTEM
    $\mathbf{\Gamma} \leftarrow$ Clear flip and inhibited state for each boundary
    $\mathbf{\Gamma} \leftarrow$ Update t_{ext} for each boundary
    $\mathbf{\Gamma}_{\text{flip}} \leftarrow$ Boundaries to flip with $t_{\text{ext}} \in [0, \Delta t]$
    $\mathcal{S}_0 \leftarrow$ Initial inhibited state of boundaries
    for $i : 1, \ldots, n$ **do**
        $\mathcal{S}_i \leftarrow$ POLL($\mathcal{S}_{i-1}$)
        if $\mathcal{S}_i = \mathcal{S}_{i-1}$ **then**
            return
        end if
    end for
end procedure

Chapter 7

Grain Statistics

In order to compare two dimensional grain arrangements with three dimensional ones, a virtual slice must be extracted from the later. [14]

Chapter 8

Three Dimensional Implicit Model for Grain Growth

In a three dimensional setting, grain boundaries surfaces of polyhedra instead of curves delimiting a plain grain. Triple junctions are now triple lines where three grains meet. The single point where four grains meet are called quadruple junctions. Formally, consider a cube unit domain $[0,1]^3 \subset \mathbb{R}^3$ with periodic boundary conditions. Notation is similar to the two dimensional setting. Grains are still defined as N disjoint regions as (2.1), that is, a set of polyhedra. Let $\mathbf{\Gamma}$ the set of grain boundaries delimiting the grains as in (2.2), but now these are polygonal surfaces. Instead of triple junctions, we have the set $\mathcal{X}$ of M quadruple junctions.

A classic approach for generating initial conditions for grain growth simulation consists in the Voronoi tessellation of a domain from a random set of initial points [1, 2, 6, 7, 15, 17]. The proposed model extends the idea of vertex-driven models to three dimensions by maintaining grain surfaces flat and triple lines straight during grain structure evolution. This implies that vertices can't be moved individually as vertex model works, otherwise grain surfaces would become curved. In order to impose these restrictions it is proposed to move the generator points used for the generation of the Voronoi tessellation instead. This implies that every time we move the generator points, we need to build a new tessellation. As the tessellation always gives flat surfaces, the imposed restriction of flat faces and straight triple lines are always accomplished [13].

The motivation for this model is that two dimensional topological transitions are already difficult to implement and check. Chapter 6 shows how cumbersome can be this process even if we introduce sophisticated mechanisms. In a three dimensional setting the possibilities for topological transitions increases since grain boundaries are now surfaces instead of curves [2]. This model handles implicitly topological transitions, that is, to avoid modifications, consistency checking and repairs to underlying data structure, only relying in how the structure is generated via tessellations.

The tessellation is build using a certain group of points in the domain. Let $\mathcal{P}$

the set of generator points of a tessellation such that:

$$\mathcal{P} = \mathcal{P}(t) = \{\mathbf{P}^{(1)}, \mathbf{P}^{(2)}, \ldots, \mathbf{P}^{(N)}\}.$$

Grain structure evolves when generator points moves and each new tessellation after generator motion is considered a new state of the grain structure. This continuous motion and tessellations are stable in front of small changes of the generator points [11]. Grain removal is assumed to happen when a grain decreases its volume until a certain minimum value is reached. The grain is removed by removing the related generator point.

The total energy of the system is inspired in the total energy of a two dimensional grain structure from (2.7) where instead of integrating along curve grain boundaries, we integrate over grain surfaces.

$$E(t) = \sum_{k=1}^{K} \int_{\Gamma^{(k)}} \sigma_k \, dA, \tag{8.1}$$

where σ_k is the grain boundary energy per unit of area. In order to simulate the grain structure evolution we must ensure that (8.1) decreases. The following velocity equation for the generator points is proposed:

$$\dot{\mathbf{P}}^{(g)} = \sum_{m=1}^{M} \gamma^{(m,l)} \mathbf{T}^{(m,l)}, \tag{8.2}$$

where m and l indicates quadruple junctions sharing a triple line, $\gamma^{(m,l)}$ is an energy term related to the triple line and $\mathbf{T}^{(m,l)}$ is the unit tangent vector to the triple line. Consider $\mathcal{X}_g$, $g = 1, \ldots, N$ the set of all the quadruple junctions that belongs to a grain g. Each quadruple junction $\mathcal{X}_g$ has four triple lines and thus four neighbor junctions, but only three of them lies in g, the other triple junction belongs to an adjacent grain g', this is the considered triple line in (8.2) between m and l and generates an unit tension $\mathbf{T}^{(m,l)}$ that will affect the motion of the generator point $\mathcal{P}^{(g)}$.

8.1 Numerical Experiments

In order to test the proposed model, two numerical experiments were performed. The first is related to how the evolution equation does indeed minimize the total energy of the grain structure. The second experiment shows that grains obtained in tessellations over time evolve and gain or lose faces through topological transitions that are not explicitly handled.

8.1.1 Energy Minimization

Recall that the proposed energy equation in (8.2) is not derived from the total energy in (8.1).

8.1.2 Topological Transitions

Bibliography

[1] K. Barmak, E. Eggeling, D. Kinderlehrer, R. Sharp, S. Ta'asan, A. Rollett, and K. Coffey. Grain growth and the puzzle of its stagnation in thin films: The curious tale of a tail and an ear. *Progress in Materials Science*, 58(7):987–1055, aug 2013.

[2] L. A. Barrales-Mora, G. Gottstein, and L. S. Shvindlerman. Three-dimensional grain growth: Analytical approaches and computer simulations. *Acta Materialia*, 56(20):5915–5926, 2008.

[3] A. Ferro and M. Fortes. The Elimination of Grains and Grain Boundaries in Grain Growth. *Interface Science*, 5(4):263–278, 1997.

[4] C. Herring. Surface tension as a motivation for sintering. *The physics of powder metallurgy*, 27(2):143–179, 1951.

[5] D. Kinderlehrer, J. Lee, I. Livshits, S. Ta'asan, and P. Yu. Multiscale modeling and simulation of grain boundary evolution. In *44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 1611, 2003.

[6] D. Kinderlehrer, I. Livshits, and S. Ta'asan. A Variational Approach to Modeling and Simulation of Grain Growth. *SIAM Journal on Scientific Computing*, 28(5):1694–1715, 2006.

[7] E. A. Lazar, J. K. Mason, R. D. MacPherson, and D. J. Srolovitz. A more accurate three-dimensional grain growth algorithm. *Acta Materialia*, 59(17):6837–6847, 2011.

[8] W. W. Mullins. Two-dimensional motion of idealized grain boundaries. *Journal of Applied Physics*, 27(8):900–904, 1956.

[9] K. Piękoś, J. Tarasiuk, K. Wierzbanowski, and B. Bacroix. Generalized vertex model of recrystallization–application to polycrystalline copper. *Computational materials science*, 42(4):584–594, 2008.

[10] K. Piękoś, J. Tarasiuk, K. Wierzbanowski, and B. Bacroix. Stochastic vertex model of recrystallization. *Computational Materials Science*, 42(1):36–42, 2008.

[11] D. Reem. The geometric stability of voronoi diagrams with respect to small changes of the sites. In *Proceedings of the twenty-seventh annual symposium on Computational geometry*, pages 254–263. ACM, 2011.

[12] F. Sausset and G. Tarjus. Periodic boundary conditions on the pseudosphere. *Journal of Physics A: Mathematical and Theoretical*, 40(43):12873, 2007.

[13] A. H. J. Sazo and C. E. Torres. (Unpublished) An Implicit-Transition Model for Numerical Simulation of 3D Grain Growth. In *2017 36th International Conference of the Chilean Computer Science Society (SCCC)*, 2017.

[14] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri. NIH Image to ImageJ: 25 years of image analysis. *Nature methods*, 9(7):671–675, 2012.

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

[16] G. B. Thomas. *Cálculo, varias variables*. Pearson Education, 12 edition, 2010.

[17] C. E. Torres, M. Emelianenko, D. Golovaty, D. Kinderlehrer, and S. Ta'asan. Numerical Analysis of the Vertex Models for Simulating Grain Boundary Networks. *SIAM Journal on Applied Mathematics*, 75(2):762–786, 2015.

[18] L. N. Trefethen. *Spectral Methods in MATLAB*, volume 10. SIAM, 2000.

[19] B. Van der Boor. Curvature-driven grain growth. Master thesis, Delft University of Technology, 2016.