

PAPER

A vertex model of recrystallization with stored energy implemented in GPU

To cite this article: Alejandro H J Sazo *et al* 2019 *Mater. Res. Express* **6** 055506

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

Materials Research Express



PAPER

A vertex model of recrystallization with stored energy implemented in GPU

RECEIVED
5 December 2018

REVISED
10 January 2019

ACCEPTED FOR PUBLICATION
21 January 2019

PUBLISHED
6 February 2019

Alejandro H J Sazo¹, Claudio E Torres^{1,2,5} , Maria Emelianenko³ and Dmitry Golovaty⁴

¹ Departamento de Informática, Universidad Técnica Federico Santa María, Chile

² Centro Científico Tecnológico de Valparaíso, Chile

³ George Mason University, Fairfax, VA United States of America

⁴ University of Akron, Akron, OH United States of America

⁵ Author to whom any correspondence should be addressed.

E-mail: alejandrosazo@usm.cl, ctorres@inf.utfsm.cl, memelian@gmu.edu and dmitry@uakron.edu

Keywords: Polycrystalline materials, grain growth, vertex model, numerical simulation, stored energy, recrystallization

Abstract

Recrystallization models and simulations have been the subject of much attention in materials community in the past decades due to this process having a significant effect on many technologically important materials characteristics. Statistical analysis performed close to the steady state requires large-scale simulations, which are often prohibitively expensive from computational point of view. Graphical Processing Unit (GPU)-based realizations provide a viable approach to addressing this challenge, yet they remain relatively under-explored in this context. In the present manuscript, we develop a fully-parallelizable matrix-free GPU-based algorithm for implementing a two-dimensional vertex model of recrystallization based on the stored energy formalism. Nucleation is assumed to take place at triple junctions and obeys a Metropolis-type criterion. We include a complete mathematical analysis of the nucleation model deriving conditions under which nucleation is successful. Stability analysis of the dynamics of a triple junction under the presence of bulk energy is provided. On the computational side, we propose a novel polling system for handling topological transitions to ensure robust GPU implementation. Single grain tests are performed for benchmarking purposes. Finally, a set of numerical experiments for large scale systems is presented to explore the effect of initial distributions of stored energy on several statistical characteristics.

1. Introduction

The recrystallization process consists of nucleation and growth of new grains in a deformed microstructure [1]. It is generally accepted that the stored energy plays a key role in recrystallization, as it provides a mechanism to record deformation of individual grains. The growth of nucleated grains is possible when the stored energy of new grains is lower than that of the surrounding grains [2]. Due to its significance in many manufacturing processes, recrystallization has been extensively studied over the last several decades, both experimentally and analytically [3]. Several competing approaches for simulating recrystallization have been proposed, ranging from modified vertex models to level sets, phase field and Monte Carlo simulations [4, 5].

A recent overview of recrystallization models is given in Orend *et al* [6]. All these methodologies have their own merit as they shed light onto various features of the material evolution. We briefly review some of the main approaches below, focusing specifically on their ability to handle computational challenges. The aim of this work is to develop a robust vertex-based recrystallization algorithm to be used in large-scale simulations that could present a viable alternative to the traditional approaches described below.

One of the favorite methods when it comes to grain growth and recrystallization simulations is the Monte Carlo Potts method. This approach has been extensively studied in [7–10] and has the advantage that nucleation can be assumed to happen anywhere in the domain. It is relatively inexpensive from numerical point of view, but demands careful calibration in order to reproduce realistic mesoscale features of the network. Luo *et al* [11]

developed a parallel version of the Potts grain growth model, but GPU-based realizations remain relatively under-explored in this context.

In phase field community, there also have been successful attempts to model this phenomenon, for instance, see the work of [12–14]. GPU-based accelerated schemes were implemented in [15, 16]. ‘Ultra-large-scale’ phase field grain growth simulations in [17] were able to handle 3 million grains. This allowed to perform extensive statistical analysis close to the steady-state, something small-scale simulations are unable to do. The authors are not aware of any similar scale simulation incorporating nucleation.

Bernacki *et al* [18] developed a finite-element model for recrystallization using a level-set framework. In this formulation, each grain is described by a single level set with neither overlap nor empty spaces between grains allowed. Nucleations were handled by introducing additional level-sets. A two-dimensional level-set approach to recrystallization was also considered in Hallberg [19]. Later, in [20], an extension of the model introduced in [18] was developed, where nucleation of new grains was handled via a combined probabilistic/deterministic approach. In [20], dynamics of triple junctions was studied numerically and the steady state of the system was determined. The methodology of [20] allowed for simulations of systems of 100 to 1000 grains. Mießen *et al* [21] proposed a two- and three-dimensional level-set algorithm adapted for ccNUMA architecture that allows highly efficient numerical simulations using OpenMP. Diffusion generated level set methods were proposed in [22], where about 4 million grains are nucleated and a comparative analysis is provided relative to Monte Carlo simulations.

Ilin *et al* [23] performed three-dimensional simulations of recrystallization using a finite-element model. The evolution of a grain boundary network was shown to result in complex shapes of grains that, in authors’ opinion, were resembling those observed in experiments. When the stored energy was incorporated into the model, higher complexity of the grain boundary network topology was observed. The numerical simulations in [23] were performed for systems of 100 grains.

Cellular automaton has also been used in recrystallization models in 2D and 3D [24], where domain decomposition strategies have been used to handle numerical simulations with MPI. The effect of the domain decomposition techniques implemented using computer network communication was investigated. In particular, the authors of [24] considered how the microstructure was influenced by the limits they placed on interactions in order to reduce communications between the sub-domains. More recent studies with cellular automaton include [25, 26].

Pieko *et al* [1, 27, 28] extended the two-dimensional vertex model of Kawasaki *et al* [29] and Weygand *et al* [30] by adding a stored energy term [1, 27, 28, 31]. It was demonstrated that these simulations matched experimental predictions in terms of the behavior of several materials properties of polycrystals, such as electrical resistance, internal stresses, micro-hardness, among others, see [1, 27, 31]. In [1] and [31] a stochastic vertex model with stored energy was considered under the assumption that nucleation is site-saturated. The computational experiments in this work were restricted to small-scale systems due to numerical limitations.

Mellbin *et al* [32] proposed a GPU-based algorithm, where they implement the computation of the stresses and tangent stiffness in GPU for a finite-element method code. They proceeded to develop a graph-based vertex model coupled with finite elements to allow nucleation, which is also implemented in GPU [33]. In a recent work of Mellbin *et al* [34], an extended analysis of vertex models was performed. In particular, topological transitions, nucleation at triple junctions and along grain boundaries together with GPU-based parallelization were considered.

In this work we focus our attention on producing a large-scale version of the vertex-type recrystallization simulation. In [35], we outlined a strategy for modeling evolution of two-dimensional grain boundary networks suitable for large-scale GPU implementation. Here we adapt the continuous vertex model formulation of Torres *et al* [36] to introduce a stored energy term that allows for modeling of recrystallization of large-scale systems. Our approach is a natural extension of the work presented in [1]. Specifically, we propose a matrix-free algorithm to derive the velocities of the triple junctions based on a gradient descent approach that enables a continuous description of the grain growth evolution. We analyze the conditions for successful nucleation. To the best of our knowledge, such analysis has not yet been carried out in this context. We propose a parallel algorithm for managing topological transitions in the grain boundary network to ensure robust GPU-based implementation. This allows for a fully parallelizable GPU-based implementation of all stages of the algorithm. The algorithm is designed to work with a relative small difference of stored energy between grains, neglecting the TA-type topological transitions, see [37].

The evolution of triple junctions is considered for both isotropic and anisotropic grain boundary energies. We thoroughly analyze the effects of the stored energy term on the dynamics of the evolution of the triple junctions and compare the results to those of a traditional vertex model. We demonstrate that the evolution of a single triple junction reaches a steady state that depends on the stored energy term. Finally, we compute statistical descriptors of the grain boundary network and explore how they are affected by the nucleation process. Note that all simulations performed in this work have been initialized using random Voronoi

tessellation. The role of initial distributions and possible applications of Inverse Monte Carlo techniques [38, 39] will be explored in future publications.

The paper is organized as follows: in section 2 we present the general framework of the gradient descent algorithm for handling the evolution equations. Section 3 presents the matrix-free algorithm for computing the gradient for the evolution of the grain network. In section 4 we study the dynamics of triple junction and the effect of the stored energy on it. Section 5 develops the mathematical conditions for obtaining a successful nucleation. Section 6 shows the main algorithm used for the evolution of the grain network. In section 7 we propose a parallel polling for handling topological transitions in a GPU. Section 8 shows an extensive list of numerical experiments. Finally, in sections 9 and 10 we present the conclusions and future work, respectively.

2. Vertex model with a stored energy term

In what follows, a 2-dimensional system of grains is a collection of non-overlapping polygonal regions (or grains) that completely cover a domain $\Omega \subset \mathbb{R}^2$. The curves separating adjacent grains are called grain boundaries that form a *grain boundary network* in Ω .

We emphasize that a 2-dimensional grain boundary network should be interpreted as a slice of its 3-dimensional counterpart. Adopting the terminology in [38], the neighboring 3-dimensional grains are separated by grain boundary surfaces which, in turn, intersect along triple junction lines. The triple junction lines terminate at vertex points (also known as quadruple junctions). Along a 2-dimensional slice, the traces of grain boundary interfaces and triple junction lines are grain boundary curves and *triple junction points*, respectively. Note that three grain boundary curves intersect at a given triple junction.

Following the notation from [1], the local energy of a vertex i considering the stored energy term is defined as follows:

$$E_i = \sum_{j \in \mathcal{N}_i} (\gamma^{(i,j)} \mathcal{L}_{i,j} + \mathcal{E}_{i,j} A_{i,j}),$$

where \mathcal{N}_i is the set of neighboring vertices to the vertex i , $\gamma^{(i,j)}$ and $\mathcal{L}_{i,j}$ are the grain boundary energy and the arc length of the boundary connecting the 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 and the grain area contributions, so:

$$E_i = \sum_{j=1}^3 \gamma^{(k_{i,j})} \mathcal{L}_{k_{i,j}} + \sum_{j=1}^3 \mathcal{E}_{g_{i,j}} A_{g_{i,j}},$$

where $k_{i,j}$ and $g_{i,j}$ are indices for the three boundaries and grains relative to vertex i . Now, we could define the total energy of the system as:

$$E = \sum_{k=1}^K \gamma^{(k)} \mathcal{L}_k + \sum_{g=1}^N \mathcal{E}_g A_g. \quad (1)$$

where K and N are the total number of boundaries and grains, respectively. Notice that if we normalize the total energy for a characteristic value of the grain boundary energy γ_0 , we obtain: $E = \gamma_0 \left(\sum_{k=1}^K \frac{\gamma^{(k)}}{\gamma_0} \mathcal{L}_k + \sum_{g=1}^N \frac{\mathcal{E}_g}{\gamma_0} A_g \right)$ and the quotient $\frac{\mathcal{E}_g}{\gamma_0}$ between stored energy and grain boundary energy will define the dynamics of the system. For clarity in the discussion, we consider $\gamma_0 = 1$ from now on.

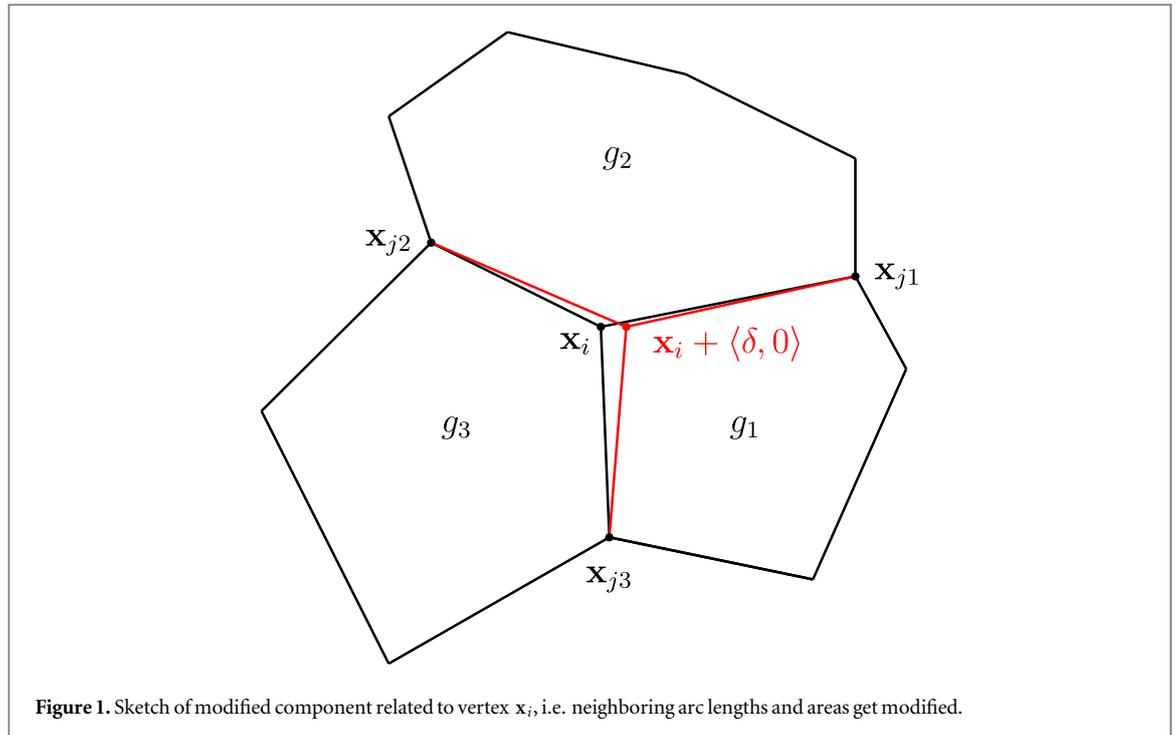
If we specify the dependency of the boundary arc lengths and grain areas with respect to the vertice positions $\{\mathbf{x}_i = \langle x_i, y_i \rangle\}$, we can build the evolution equations for this model such that it decreases its energy by a gradient

descent method, that is to compute $\dot{\mathbf{x}}_i(t) = \left\langle -\frac{\partial E}{\partial x_i}, -\frac{\partial E}{\partial y_i} \right\rangle$.

3. Numerical algorithm

Computing the *velocity* of each vertex is actually the computation of the gradient of the energy E from equation (1). Here we propose a matrix-free approach to approximate the gradient. The main advantage of this approach is that it only needs the numerical implementation of the computation of the total energy of the system, see equation (1).

A convenient way to approximate each partial derivative is with a finite difference approach. Consider the vector $\mathbf{X} \in \mathbb{R}^{2M}$ of stacked components of \mathbf{x}_i , this is:



$$\mathbf{X} = \langle x_1, x_2, \dots, x_M, y_1, y_2, \dots, y_M \rangle^T,$$

where M is the total number of triple junctions and T is the transpose operator. The m -th component of the gradient vector, say $\partial E(\mathbf{X})/\partial \mathbf{X}_m$ is approximated by:

$$\frac{\partial E(\mathbf{X})}{\partial \mathbf{X}_m} \approx \frac{E(\mathbf{X} + \delta \mathbf{e}_m) - E(\mathbf{X})}{\delta}, \quad (2)$$

where \mathbf{e}_m is the m -th canonical vector in \mathbb{R}^{2M} . The numerical evaluation of this approximation implies that the energy of the system must be recomputed twice for each vertex, one to compute $E(\mathbf{X})$ and the second one is $E(\mathbf{X} + \delta \mathbf{e}_m)$, where a small perturbation is added to the m -th component. According to (1), this means recomputing each individual vertex of energy each time. Fortunately, this can be handled efficiently by pointing out which terms of energy will be changed after the $\delta \mathbf{e}_m$ perturbation is added.

A quadratic cost would be required for a naive implementation. A more detailed analysis shows that the perturbation only affects the x or y component of a vertex \mathbf{x}_i , which implies that only the three related grain areas and the three grain boundary arc lengths connected to vertex \mathbf{x}_i will change, as shown in figure 1. So, instead of recomputing the energy for each one of the $2M$ components of the gradient, we can compute directly the difference of energies given by the perturbation. Expanding the difference of energies from (2) and taking into account that the perturbation effect is local, we obtain:

$$\begin{aligned} E(\mathbf{X} + \delta \mathbf{e}_m) - E(\mathbf{X}) &= \sum_{k=1}^K \gamma^{(k)} \Delta \mathcal{L}_k + \sum_{g=1}^N \mathcal{E}_g \Delta A_g \\ &= \sum_{j=1}^3 \gamma^{(k_{ij})} \Delta \mathcal{L}_{k_{ij}} + \sum_{j=1}^3 \mathcal{E}_{g_{ij}} \Delta A_{g_{ij}}, \end{aligned}$$

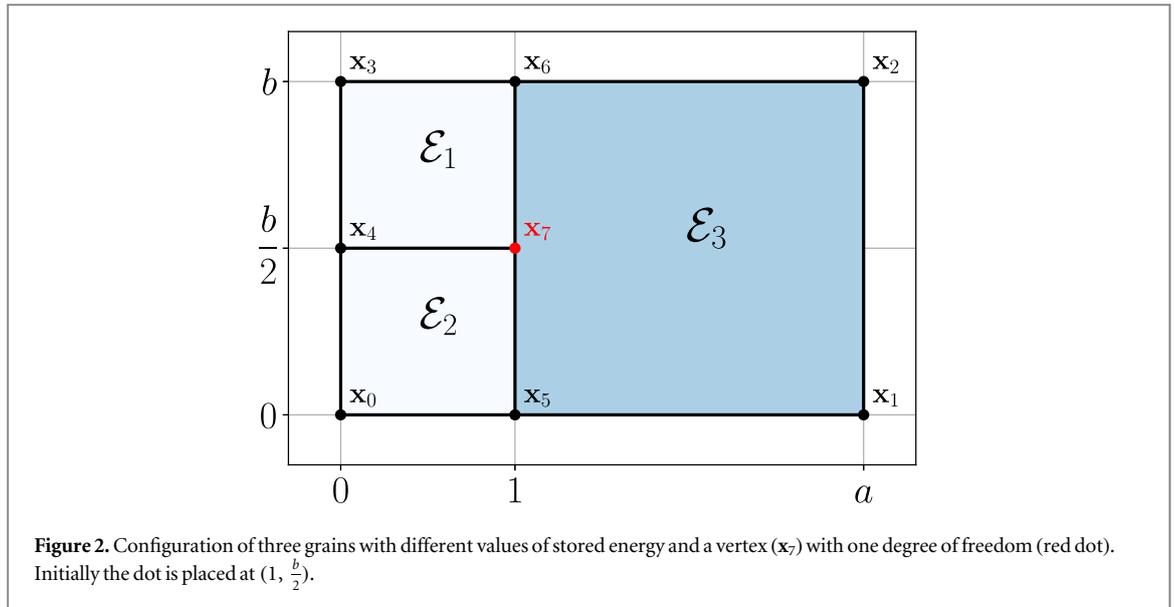
where $i = m \bmod M$. Notice that the three boundaries and grain areas are modified by adding a perturbation to the vertex i . Now, dividing by δ we obtain the right-hand-side of (2),

$$\frac{E(\mathbf{X} + \delta \mathbf{e}_m) - E(\mathbf{X})}{\delta} = \sum_{j=1}^3 \gamma^{(k_{ij})} \frac{\Delta \mathcal{L}_{k_{ij}}}{\delta} + \sum_{j=1}^3 \mathcal{E}_{g_{ij}} \frac{\Delta A_{g_{ij}}}{\delta}$$

Moreover, each estimation of the gradient can be computed efficiently in parallel since only local information of areas and arc lengths is needed.

4. The effect of stored energy on the dynamics of a triple junction

This section studies how the stored energy term affects the configuration and stability of the grain network. We extend the stability study done in [20] where only stable configurations are considered.



Considering a bounded domain where three grains coexist, where two of them have the same stored energy, i.e. $\mathcal{E}_1 = \mathcal{E}_2$, and the third has a different stored energy \mathcal{E}_3 , as shown in figure 2. To quantify the effect of the stored energy term, we let the central vertex located at $(x, b/2)$ move freely, and due to symmetry this will only move along the x -axis. Let A_1, A_2, A_3 be the areas of the grains with stored energy $\mathcal{E}_1, \mathcal{E}_2$ and \mathcal{E}_3 , respectively. Grain areas can be explicitly computed as:

$$A_1 = A_2 = \frac{b(1+x)}{4}, A_3 = ab - \frac{b(1+x)}{2}. \tag{3}$$

The total energy becomes:

$$E = \sqrt{b^2 + 4(-1+x)^2} + x + a(4 + b\mathcal{E}_3) + \frac{b(8 + \mathcal{E}_1(x+1) - \mathcal{E}_3(1+x))}{2}. \tag{4}$$

If we compute the derivative of (4) with respect to x we obtain,

$$\frac{dE}{dx} = 1 + \frac{4(x-1)}{\sqrt{b^2 + 4(x-1)^2}} - \frac{b\Delta\mathcal{E}}{2} \tag{5}$$

where $\Delta\mathcal{E} = \mathcal{E}_3 - \mathcal{E}_1$. Thus, the motion of the central vertex is influenced by the difference of stored energies and the height of the domain b . Notice that the domain width a does not affect the evolution. The steady state can be found making (5) equal to 0, which yields the following roots:

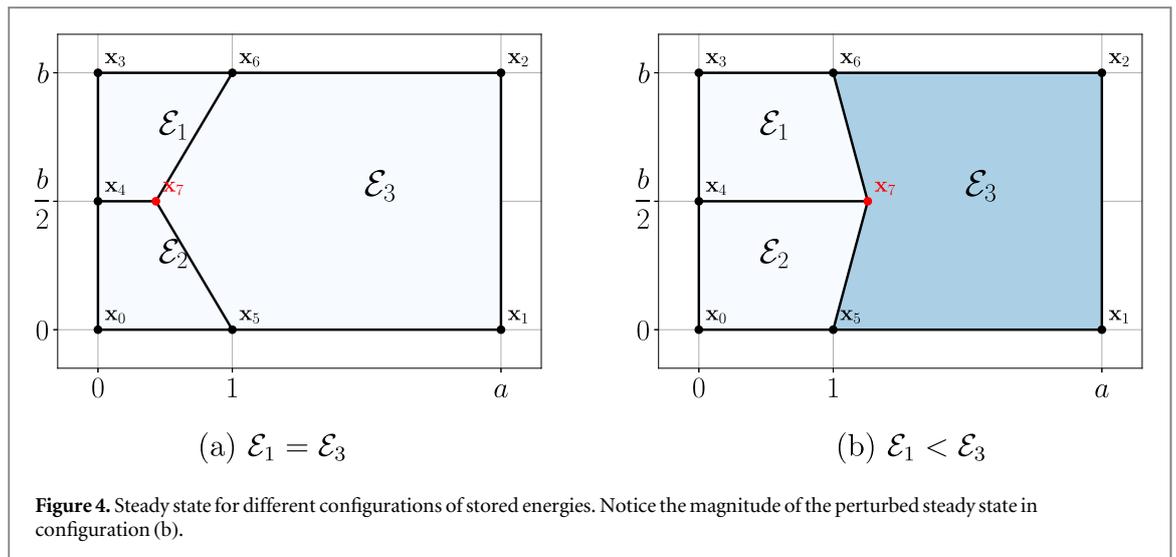
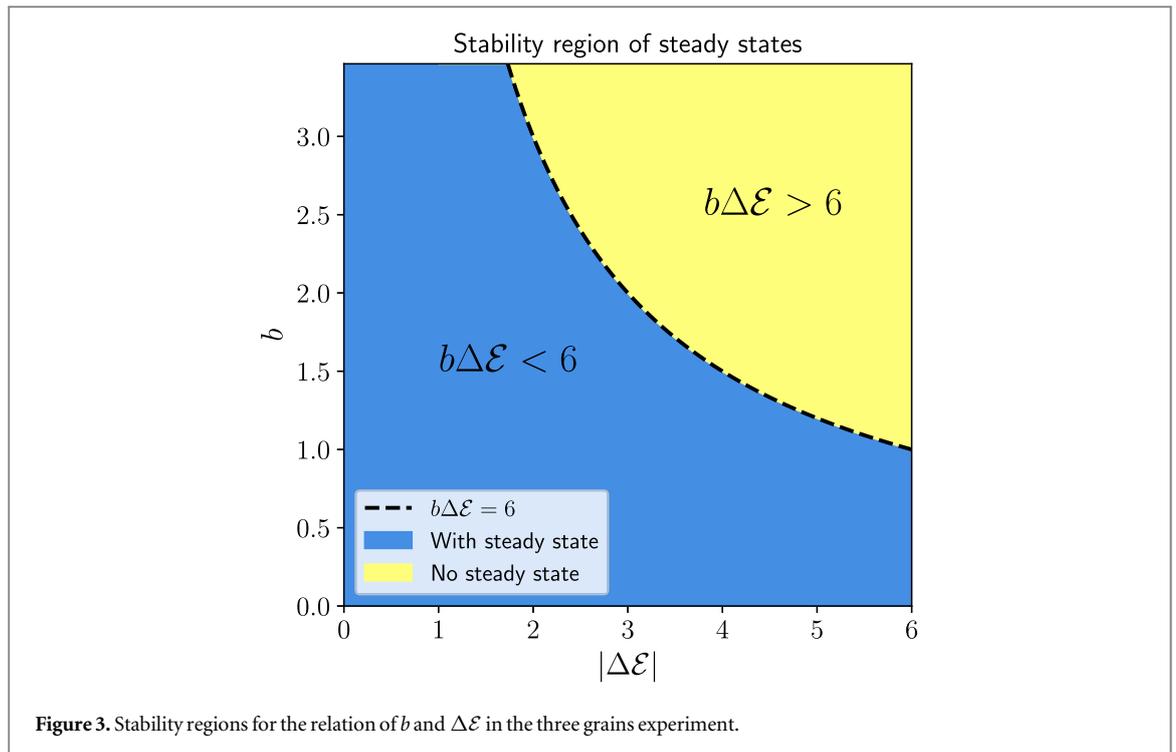
$$x_{\pm} = \frac{-24 + 2b\Delta\mathcal{E}(b\Delta\mathcal{E} - 4) \pm \sqrt{-b^2(b\Delta\mathcal{E} - 6)(b\Delta\mathcal{E} - 2)^2(b\Delta\mathcal{E} + 2)}}{2(b\Delta\mathcal{E} - 6)(b\Delta\mathcal{E} + 2)}. \tag{6}$$

Notice that under equal values of stored energy, only the domain height b defines the position of the steady state, which is:

$$x_{\text{steady}} = x_+ = 1 - \frac{b}{2\sqrt{3}}.$$

By analyzing the obtained roots, only x_+ has physical sense. The critical point, assuming $\Delta\mathcal{E} > 0$, is when $b\Delta\mathcal{E} < 6$, as shown in figure 3.

The concept of stability here refers to the interpretation of the vertex location in the domain. If the vertex stays inside the domain, we say that it is a stable steady state. If the vertex penetrates a grain without stopping, the configuration is said to be unstable. Figures 4(a) and (b) show two steady states. Figure 4(a) shows the steady state when there is no stored energy; therefore, the steady state is reached at dihedral angle $2\pi/3$. In the presence of a large stored energy, the steady state is situated away from the isotropic steady state as shown in figure 4(b). Again, when $b\Delta\mathcal{E} \geq 6$ the solution becomes unbounded. Figure 5 shows the trajectories for different values of stored energies. The black dashed line shows the critical point when the triple junction moves from a bounded solution to an unbounded solution, which are denoted as unstable solutions.

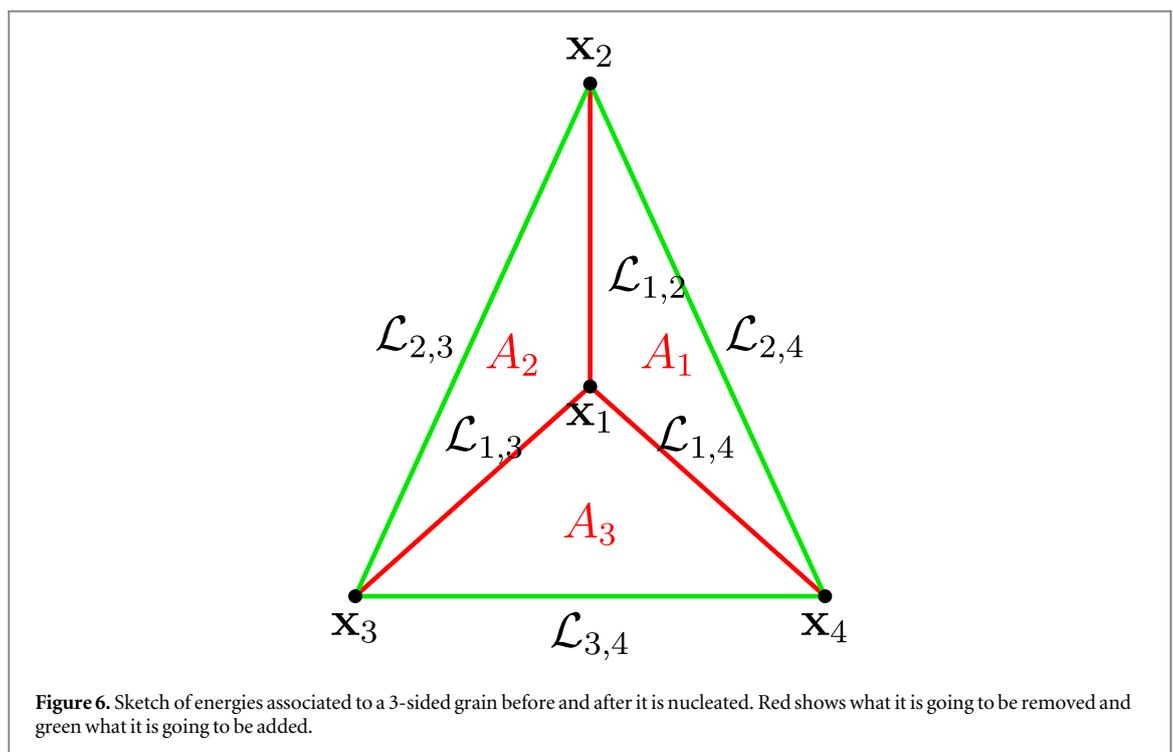
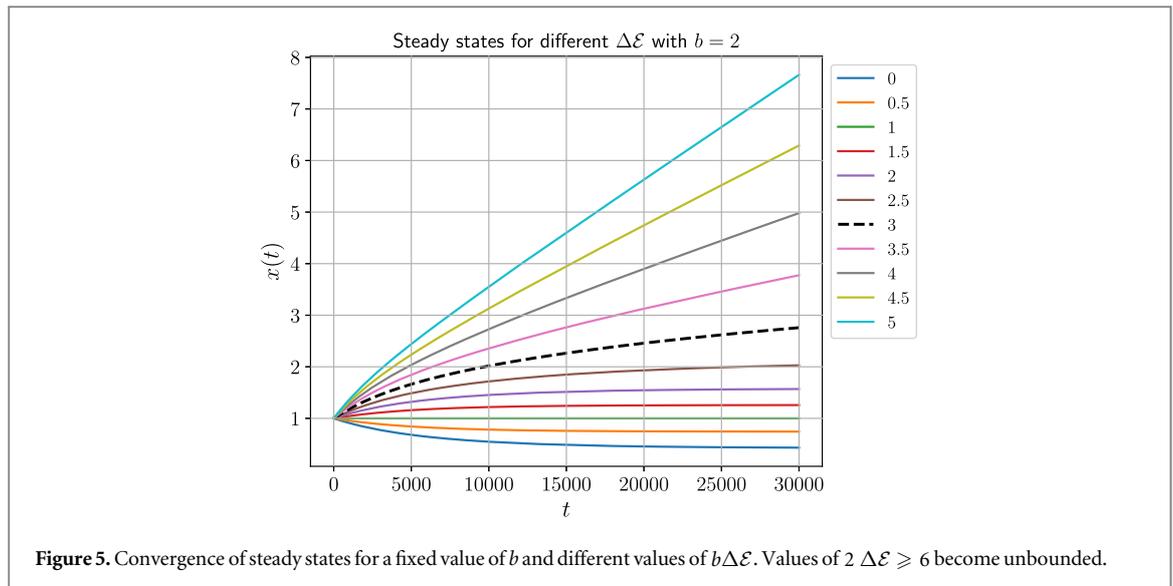


5. Nucleation and growth of three-sided grains

The nucleation process can be classified as a new type of topological change in the system, considering the already well-known topological changes: flipping and grain removal. As usual, this new type of topological change should be energy decreasing; otherwise, it may induce the removal of the nucleated grain right away. To analyze this topological change, we propose the following sketch (see figure 6). In this sketch we observe what we call the current configuration of energy E_0^Δ , in red, and a candidate configuration of energy E_1^Δ . The main idea is that for a candidate vertex x_i , where we could perform a nucleation, we can explicitly compute the difference of energy before and after the nucleation, i.e. $\Delta E^\Delta = E_1^\Delta - E_0^\Delta$. Therefore, as long as the difference is negative, we can conclude that nucleation will be successful. Thus, the ΔE^Δ is as follows,

$$\begin{aligned}
 E_0^\Delta &= \gamma^{(1,2)} \mathcal{L}_{1,2} + \gamma^{(1,3)} \mathcal{L}_{1,3} + \gamma^{(1,4)} \mathcal{L}_{1,4} + \mathcal{E}_1 A_1 + \mathcal{E}_2 A_2 + \mathcal{E}_3 A_3, \\
 E_1^\Delta &= \gamma^{(2,4)} \mathcal{L}_{2,4} + \gamma^{(2,3)} \mathcal{L}_{2,3} + \gamma^{(3,4)} \mathcal{L}_{3,4}, \\
 \Delta E^\Delta &= E_1^\Delta - E_0^\Delta,
 \end{aligned}
 \tag{7}$$

where x_i are the coordinates of the vertex i for $i \in \{1, 2, 3, 4\}$, $\mathcal{L}_{i,j}$ is the arc length from vertex x_i to vertex x_j , $\gamma^{(i,j)}$ is the grain boundary energy from vertex x_i to vertex x_j , A_1 is the area of grain with vertices $\{x_1, x_4, x_2\}$,



A_2 is the area of grain with vertices $\{x_1, x_2, x_3\}$, A_3 is the area of grain with vertices $\{x_1, x_3, x_4\}$, and \mathcal{E}_g is the stored energy of the grain associated to A_g for $g \in \{1, 2, 3\}$.

5.1. First attempt to nucleate a grain

The first attempt one could try to nucleate a three-sided grain is under no presence of the stored energy term. Here, we will show this is not possible under the conditions analyzed in the following. First, consider we will use the notation used in equation (7). Specifically, we assume $\mathcal{E}_1 = \mathcal{E}_2 = \mathcal{E}_3 = 0$, $\mathcal{L}_{1,2} = \mathcal{L}_{1,3} = \mathcal{L}_{1,4} = r$, and isotropic grain boundary energy $\gamma = 1$. This leads to the following variation of total energy:

$$\begin{aligned} \Delta E^\Delta &= \mathcal{L}_{2,4} + \mathcal{L}_{2,3} + \mathcal{L}_{3,4} - \mathcal{L}_{1,2} - \mathcal{L}_{1,3} - \mathcal{L}_{1,4} \\ &= \mathcal{L}_{2,4} + \mathcal{L}_{2,3} + \mathcal{L}_{3,4} - 3r. \end{aligned}$$

So, we defined the angle between the segment defined from x_1 to x_4 and from x_1 to x_2 as ω_1 , the angle between the segment defined from x_1 to x_2 and from x_1 to x_3 as ω_2 , and the angle between the segment defined from x_1 to x_3 and from x_1 to x_4 as ω_3 . These angles allow us to obtain the unknown arc lengths using the law of cosines as follows:

$$\begin{aligned}\mathcal{L}_{2,4} &= r \sqrt{2 - 2 \cos(\omega_1)}, \\ \mathcal{L}_{2,3} &= r \sqrt{2 - 2 \cos(\omega_2)}, \\ \mathcal{L}_{3,4} &= r \sqrt{2 - 2 \cos(\omega_3)}.\end{aligned}$$

Thus, the variation of total energy becomes,

$$\Delta E^\Delta = r(\sqrt{2 - 2 \cos(\omega_1)} + \sqrt{2 - 2 \cos(\omega_2)} + \sqrt{2 - 2 \cos(\omega_3)} - 3)$$

So, considering that $\omega_1 + \omega_2 + \omega_3 = 2\pi$ and that $\pi \geq \omega_i \geq 0$ for $i \in 1, 2, 3$ we obtain $\Delta E^\Delta \geq 0$. Therefore, since ΔE^Δ is always greater or equal than 0 no matter the value of r , it is not possible to nucleate a three-sided grains under no presence of stored energy. Moreover, if we compute the variation with respect to r we obtain again the positive term $\sqrt{2 - 2 \cos(\omega_1)} + \sqrt{2 - 2 \cos(\omega_2)} + \sqrt{2 - 2 \cos(\omega_3)} - 3$, under the same consideration. Thus, the grain will be eventually removed.

5.2. Nucleating a grain with stored energy: first variation analysis

To gain insight into the general conditions for allowing nucleation. Considering figure 6 and similarly to section 5.1, we assume that $\mathcal{L}_{1,2} = \mathcal{L}_{1,3} = \mathcal{L}_{1,4} = r$ and also the same dihedral angles ω_1, ω_2 and ω_3 introduced before. Moreover, we include here the grain boundary and stored energy dependence explicitly, this means areas A_1, A_2 and A_3 have stored energy $\mathcal{E}_1, \mathcal{E}_2$, and \mathcal{E}_3 , respectively. The grain boundary energy related to $\mathcal{L}_{i,j}$ is $\gamma_{i,j}$ for $i \in \{1, 2, 3, 4\}$ and $i \in \{1, 2, 3, 4\}$. This leads us to following:

$$\begin{aligned}\Delta E^\Delta &= \sqrt{2} r(\gamma_{2,4} \sqrt{1 - \cos(\omega_1)} + \gamma_{2,3} \sqrt{1 - \cos(\omega_2)} + \gamma_{3,4} \sqrt{1 - \cos(\omega_3)}) \\ &\quad - \left(r(\gamma_{1,2} + \gamma_{1,3} + \gamma_{1,4}) + \frac{1}{2} r^2 (\mathcal{E}_1 \sin(\omega_1) + \mathcal{E}_2 \sin(\omega_2) + \mathcal{E}_3 \sin(\omega_3)) \right).\end{aligned}\quad (8)$$

So, considering isotropic grain boundary energy γ (i.e. $\gamma = 1$), dihedral angle of $2\pi/3$ and $\mathcal{E}_1 + \mathcal{E}_2 + \mathcal{E}_3 = \tilde{\mathcal{E}}$. This leads us to the following simplification,

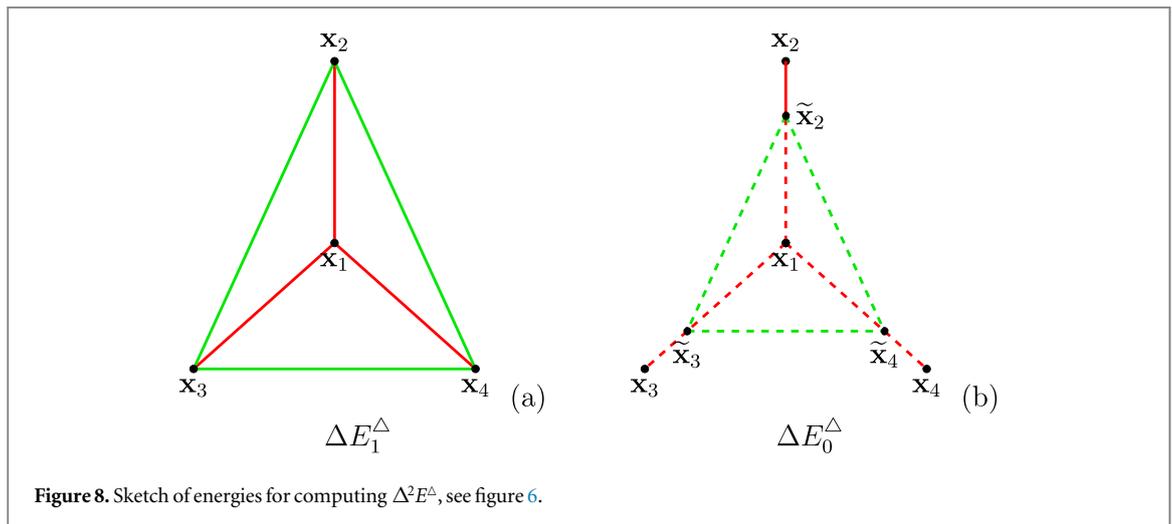
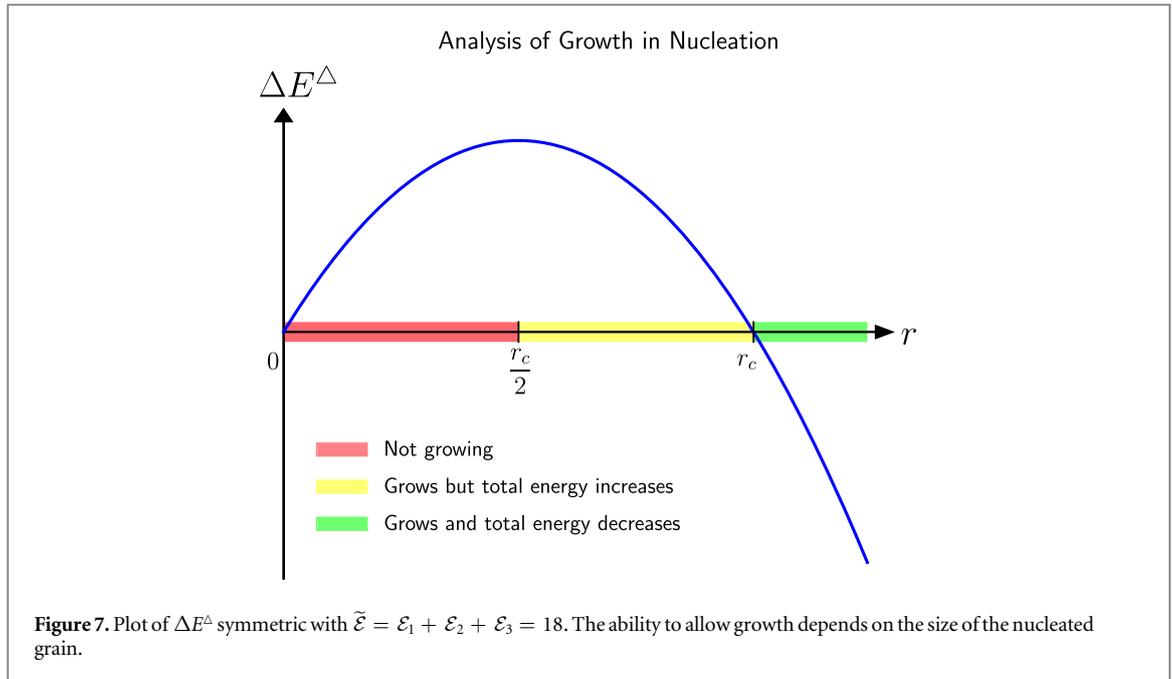
$$\begin{aligned}\Delta E^\Delta &= 3 \sqrt{3} r - 3 r - \tilde{\mathcal{E}} \frac{\sqrt{3}}{4} r^2 \\ &= 3 r \left(\sqrt{3} - 1 - \frac{\sqrt{3}}{12} \tilde{\mathcal{E}} r \right).\end{aligned}$$

From this expression, we determined that there are two critical points for $\tilde{\mathcal{E}} \neq 0$. These are $r_0 = 0$ and $r_c = \frac{4(3 - \sqrt{3})}{\tilde{\mathcal{E}}} \approx 5.071796(\mathcal{E}_1 + \mathcal{E}_2 + \mathcal{E}_3)^{-1}$. The first critical point $r_0 = 0$ indicates that nucleating the area 0 will add 0 energy, which is correct but we are not interested in that case. On the other hand, the second critical point is the one we are interested in, $r_c \approx 5.071796 \tilde{\mathcal{E}}^{-1}$, since from that point on we ensure we are not increasing the energy of the system and will let the nucleated grain grow. In figure 7 we observe a qualitative behavior of (7) for the symmetric case, with this plot showing that we require r from figure 6 to be at least $r_c = 4(3 - \sqrt{3})\tilde{\mathcal{E}}^{-1}$. This implies nucleating a large grain; otherwise, we will be adding energy to the system. An important consideration needs to be made; namely, if the nucleated grain has a radius r in the range $[0, r_c/2]$, the nucleated grain will not grow. On the other hand, if the radius of the nucleated grain is in the range $]r_c/2, r_c[$, the nucleated grain will grow but it will initially add energy to the system, and when the nucleated grain satisfies $r > r_c$, the nucleated grain will grow and will remove energy from the system. Figure 7 shows these ranges in red, yellow and green, respectively. Thus, the yellow zone is a candidate configuration to use a Monte Carlo approach.

5.3. A symmetric nucleation analysis: second variation analysis

For a successful energy decreasing nucleation, as discussed in the previous section, we have a constraint on the size of the grain to be nucleated; otherwise, the nucleated grain will not grow. Therefore, in the yellow zone there is a potential for a grain to grow but at a cost of a temporal increase of total energy. The main interest in this type of configuration is that it imposes a lower bound requirement for the size of the nucleated grain to make the nucleation successful.

A preliminary analysis indicates that the *slope* of the ΔE^Δ is negative in that range, and an energy decreasing model will favor the growth of the grain since it will decrease the energy. More rigorously, we need to compute $\Delta(\Delta E^\Delta) =: \Delta^2 E^\Delta$. Figure 8 shows a diagram of two possible nucleations at vertex \mathbf{x}_1 . In this case the nucleation on the left is the one explained previously in figure 6, which will be denoted here as ΔE_1^Δ . And the one on the right is a nucleation for a smaller grain for the same vertex \mathbf{x}_1 , which will be denoted as ΔE_0^Δ . These two *virtual* nucleations allow us to obtain $\Delta^2 E^\Delta = \Delta E_1^\Delta - \Delta E_0^\Delta$. Thus, if $\Delta^2 E^\Delta < 0$, we ensure that the nucleated grain may grow since it will decrease the energy of the system. This value will be denoted as the *nucleation threshold*.



Notice that under an anisotropic grain boundary energy the computation of $\Delta^2 E^\Delta$ will include each grain boundary energy term, which have to be computed with respect to the orientation of the nucleated grain. We set the orientation of the nucleated grain randomly in $[0, 2\pi]$.

From section 5.2, equation (8), we can estimate the limiting case for $\Delta^2 E^\Delta$. This mean to compute:

$$\lim_{\delta \rightarrow 0} \frac{\Delta E^\Delta(r + \delta) - \Delta E^\Delta(r)}{\delta} = \frac{\partial}{\partial r} \Delta E^\Delta(r), \text{ which gives us:}$$

$$\begin{aligned} \frac{\partial}{\partial r} \Delta E^\Delta = & \sqrt{2} (\gamma_{2,4} \sqrt{1 - \cos(\omega_1)} + \gamma_{2,3} \sqrt{1 - \cos(\omega_2)} + \gamma_{3,4} \sqrt{1 - \cos(\omega_3)}) \\ & - (\gamma_{1,2} + \gamma_{1,3} + \gamma_{1,4} + r(\mathcal{E}_1 \sin(\omega_1) + \mathcal{E}_2 \sin(\omega_2) + \mathcal{E}_3 \sin(\omega_3))). \end{aligned}$$

And considering $\gamma = 1$, dihedral angle of $2\pi/3$ and $\mathcal{E}_1 + \mathcal{E}_2 + \mathcal{E}_3 = \tilde{\mathcal{E}}$, we obtain:

$$\frac{\partial}{\partial r} \Delta E^\Delta = 3\sqrt{3} - 3 - \tilde{\mathcal{E}} \frac{\sqrt{3}}{2} r.$$

6. Stored energy vertex model with nucleation algorithm

The main algorithm of this model considers first the computation of boundary properties such as arc lengths and energies, and then the computation of grain areas. We choose to perform continuous nucleation, that is, nucleating at regular intervals of time in the simulation, denoted as $\Delta\tau$, see [40]. To select a nucleation vertex,

we follow a Monte Carlo approach with a probability threshold $\exp(-\kappa\Delta E^\Delta)$ with $\kappa = 1$. This is explained in algorithm 1.

Algorithm 1. Monte Carlo selection for nucleation location.

```

1: procedure MONTECARLONUCLINATION
2:  $\mathcal{X}_i \leftarrow$  Select a random vertex
3:  $r \leftarrow$  Select a random length between 0 and  $\min_k \mathcal{L}_k$  for  $k$  being the grain boundaries connected to  $\mathcal{X}_i$ 
4:  $\alpha \leftarrow$  Pick a random orientation between  $[0, 2\pi]$ 
5:  $\Delta E^\Delta \leftarrow$  Compute first variation with the randomly chosen arc length  $r$  and orientation  $\alpha$ 
6:  $\Delta^2 E^\Delta \leftarrow$  Compute second variation with the randomly chosen arc length  $r$  and orientation  $\alpha$ , this is the nucleation threshold
7: if  $\Delta E^\Delta < 0$  then
8:   return Accept nucleation of triple junction  $\mathcal{X}_i$  with arc length  $r$  and orientation  $\alpha$ 
9: else
10:  if  $\Delta^2 E^\Delta < 0$  then
11:     $p \leftarrow$  Pick a random number in  $[0, 1]$ 
12:    if  $p < \exp(-\kappa \Delta E^\Delta)$  then
13:      return Accept nucleation of triple junction  $\mathcal{X}_i$  with arc length  $r$  and orientation  $\alpha$ 
14:    else
15:      return Reject nucleation
16:    end if
17:  else
18:    return Reject nucleation
19:  end if
20: end if
21: end procedure

```

If there is a site for a successful nucleation, the grain is added and neighboring topology is updated.

The next stage considers the computation of the vertex velocities using the matrix-free approach described in section 3. Then, extinction times for each boundary are computed and we label boundaries that will flip, that is, when $t_{\text{ext}} \in [0, \Delta t]$. Here we must decide which flippings are safe to perform and therefore, we avoid neighbor transitions. Algorithm 5 is presented in section 7 and solves this issue. If a three-sided grain has a grain boundary as a candidate for flipping, a grain removal is performed. After this, we apply the remaining flippings, we update the vertice positions and the system evolves to the time $t + \Delta t$. Algorithm 2 summarizes the described procedure.

Algorithm 2. Stored Energy Vertex Model

```

1: procedure SEVM ( $n_{\text{iters}}$ , boundaries  $\Gamma$ , vertices  $\mathcal{X}$ , grains  $\mathcal{G}$ )
2:  $(t, n_{\text{iters}}) \leftarrow (0, 0)$ 
3: for  $iter = 0, \dots, n_{\text{iters}}$  do
4:    $\Gamma \leftarrow$  ComputeArclengthsAndEnergies( $\Gamma$ )
5:    $\mathcal{G} \leftarrow$  ComputeAreas( $\mathcal{G}$ )
6:   if  $iter \bmod T == 0$  then
7:      $\mathcal{X}_k \leftarrow$  MonteCarloNucleation
8:     if  $\mathcal{X}_k \neq \emptyset$  then
9:        $(\Gamma, \mathcal{X}, \mathcal{G}) \leftarrow$  Nucleate( $\mathcal{X}_k$ )
10:       $\Gamma \leftarrow$  ComputeArclengthsAndEnergies( $\Gamma$ )
11:       $\mathcal{G} \leftarrow$  ComputeAreas( $\mathcal{G}$ )
12:       $\mathcal{X} \leftarrow$  ComputeNucleationThreshold( $\mathcal{X}$ )
13:    end if
14:  end if
15:   $\mathbf{V} \leftarrow$  ComputeMatrixFreeVelocities( $\mathcal{X}$ )
16:   $\Gamma \leftarrow$  PollingSystem( $\Gamma$ )
17:   $(\Gamma, \mathcal{X}, \mathcal{G}) \leftarrow$  RemoveThreeSidedGrains( $\Gamma, \mathcal{X}, \mathcal{G}$ )
18:   $\Gamma \leftarrow$  ApplyFlips( $\Gamma$ )
19:   $\mathcal{X} \leftarrow$  UpdateVerticesPositions( $\mathcal{X}, \mathbf{V}$ )
20:   $iter \leftarrow iter + 1$ 
21:   $t \leftarrow t + \Delta t$ 
22: end for
23: end procedure

```

▷ See algorithm 5

As discussed in section 4, vertices exposed to grain with stored energy equal to 0 and other grains with stored energy different that 0, may become *unstable*. This could bring overlapping grains during the numerical experiments. To handle this, we only considered small differences of stored energy.

7. The parallel polling system for management of topological transitions

Topological transitions modify the grain structure and the data structure must reflect these relations at every time step. The update of this data structure is critical due to the consistency during numerical simulation and implementation performance. We discuss in this section the sequential management algorithm for topological transitions in a grain growth numerical simulation, and then we propose a parallel management algorithm that aims to handle topological transitions in a fully GPU-based-code.

7.1. Sequential management

In a sequential implementation of a vertex code, [36], the topological transitions, flippings and grain removal can be detected by estimating the time that a boundary will collapse, called extinction time (t_{ext}). If the extinction time lies between the current time t and one time step after $t + \Delta t$, i.e. within $[t, t + \Delta t]$, the boundary is considered a candidate to flip. A simple algorithm to solve the transitions within the time step is to sort the candidate boundaries with positive extinction $\in [t, t + \Delta t]$ into ascending order. We can then iteratively solve the conflicts by labeling those candidates involved in transitions of other candidates, and the labeled boundaries are inhibited to flip this time step. Once all the conflicts have been solved, the transitions are performed. Algorithm 3 summarizes the described management.

Algorithm 3. Sequential Management of Topological Transitions

```

1: procedure SEQUENTIAL
2:    $\Gamma \leftarrow$  Compute  $t_{\text{ext}}$  for each boundary
3:    $\Gamma_{\text{flip}} \leftarrow$  Select candidates boundaries to flip with  $t_{\text{ext}} \in [0, \Delta t]$ 
4:    $\Gamma_{\text{flip}} \leftarrow$  Sort boundaries by  $t_{\text{ext}}$  increasingly
5:    $\Gamma_{\text{tmp}} \leftarrow \emptyset$ , Empty list of boundaries visited
6:    $\mathcal{X}_{\text{tmp}} \leftarrow \emptyset$ , Empty list of vertices visited
7:   for each  $\Gamma \in \Gamma_{\text{flip}}$  do
8:      $\{x_i, x_j\} \leftarrow$  Vertices of  $\Gamma$ 
9:     if  $\mathcal{X}_{\text{tmp}} \cap \{x_i, x_j\} = \emptyset$  then
10:       $\Gamma_{\text{tmp}} \leftarrow \Gamma_{\text{tmp}} \cup \Gamma$ 
11:       $\mathcal{X}_{\text{tmp}} \leftarrow \mathcal{X}_{\text{tmp}} \cup \{x_i, x_j\}$ 
12:     else
13:       return  $\Gamma_{\text{tmp}}$ 
14:     end if
15:   end for
16: end procedure

```

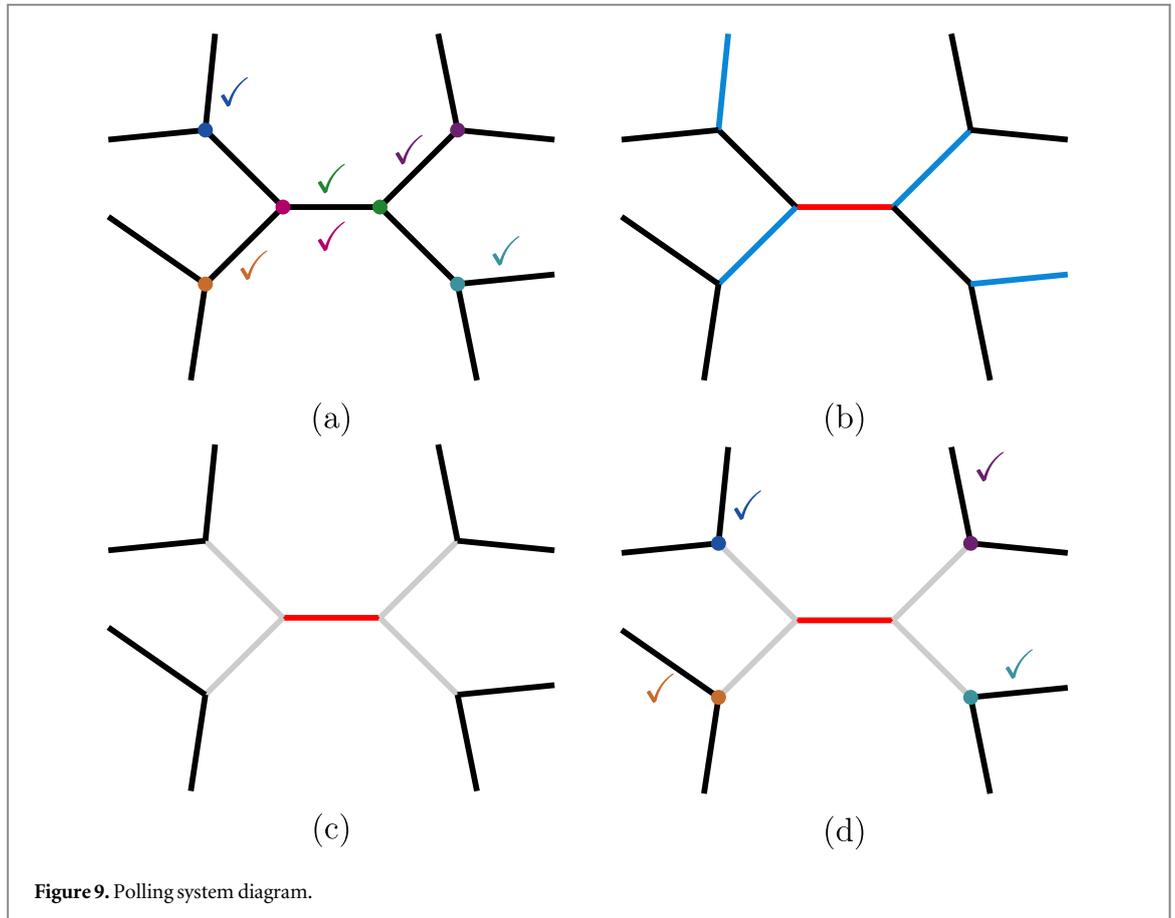
Due to the need for more significant statistics [20, 31, 34], hundreds of thousands of grains must be simulated. The sequential implementation of these models in CPU treats grain structure evolution and topological transitions sequentially. It is possible to parallelize the structure evolution in CPU and still solve the topological transitions in the traditional way from algorithm 3 but this process might become a bottleneck when we require large numerical simulations.

We chose to implement this model in a GPU and therefore, we expect to handle topological transitions by taking advantage of such architecture, that is, avoiding whenever is possible the sequential treatment of the grain network, i.e. data structure.

The parallel implementation in GPU is straightforward when we speak of evolving the data structure: in the CUDA programming language [41, 42] we can launch one thread per vertex or boundary to compute their velocities, and then update the positions. The problem of solving the topological transitions arises, since the mechanism to solve inconsistencies is sequential and parallelism over the data structure will generate race conditions. For example, naive implementation of the sequential algorithm in GPU with CUDA would cause that each thread working with a boundary to attempt a modification of the lists of boundaries visited, and this may produce a corrupt data structure. Thus, a new algorithm for parallel management of topological transitions must be built.

7.2. Parallel polling system

Since the concurrent access to a 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 of the boundary that among the three boundaries connected to it has the lowest possible extinction time within $[t, t + \Delta t]$.

We then have for each vertex a *voted* boundary, i.e. the boundary with the lowest positive time in $[t, t + \Delta t]$ and if the boundary has not been already been inhibited for flipping. This is stored for each vertex to avoid race conditions. Next, each boundary counts the obtained votes by referring to its vertices. The vote counting is the key, with two votes meaning that a boundary can flip since all its neighbor boundaries have chosen it, and it earns the right to label neighbor boundaries as inhibited for flipping in this time step. Notice that we only allow at most one boundary flipping for each vertex to avoid corrupting the data structure; however, several flippings could happen in parallel if they are not connected to the same vertices. Other boundaries will obtain one or zero votes, and thus they are not candidates for flipping. This polling is repeated until no further labeling is possible, that is, when the previous set of candidates is the same as the current iteration.

Figure 9 shows a diagram of the polling system. Consider a neighborhood of boundaries with their extinction times already computed. Figure 9(a) shows the vertex poll. Each vertex in this neighborhood will vote for their boundary with the lowest positive t_{ext} , and the check marks (votes) with the same color of a vertex means that that vertex voted for that boundary. We can see that some boundaries obtained zero, one or two votes. The case of a boundary with two votes is special because it implies that two vertices (pink and green) decided that their shared boundary is a good candidate for flipping and therefore no other boundary in the immediate neighborhood is allowed to be a flipping candidate. Figure 9(b) 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. The chosen boundary labels its neighbor boundaries as inhibited as shown in figure 9(c) in gray. Inhibited boundaries cannot be voted at the next iteration. Related vertices must vote between the remaining non-inhibited boundaries, as shown in figure 9(d).

It is considered that a grain boundary could have three states: candidate, inhibited, and uninhibited. This procedure can be seen as a fixed point iteration over a polling function $\text{Poll}(\mathbf{T})$, where the input is the current boundary set and it returns the inhibited states. When the polling function returns the same inhibited states, we will have found a fixed point of the function and thus we will have converged to a feasible solution. Algorithm 5 summarizes this procedure.

Algorithm 4. Polling Routine for Inhibit Boundaries

```

1: procedure POLL(boundaries  $\Gamma$ )
2:    $\Gamma \leftarrow$  Vertices votes for its uninhibited boundary with lowest  $t_{\text{ext}}$ 
3:    $\Gamma \leftarrow$  Boundaries counts the votes received
4:    $\Gamma \leftarrow$  Select uninhibited boundaries with two votes
5:    $\Gamma \leftarrow$  Inhibit neighbor boundaries of selected boundaries
6:    $S \leftarrow$  Inhibited state of boundaries
7:   return  $S$ 
8: end procedure

```

Algorithm 5. Parallel Polling System for Managing Topological Transitions

```

1: procedure POLLINGSYSTEM(boundaries  $\Gamma$ )
2:    $\Gamma \leftarrow$  Compute  $t_{\text{ext}}$  for each boundary
3:    $\Gamma \leftarrow$  Label boundaries to flip with  $t_{\text{ext}} \in [0, \Delta t]$ 
4:    $S_0 \leftarrow$  Set initial inhibited state for all boundaries
5:   do
6:      $S_1 \leftarrow S_0$ 
7:      $S_0 \leftarrow$  Poll( $\Gamma$ )
8:   while  $S_1 \neq S_0$ 
9: end procedure

```

The advantage of this polling system is that the vertices voting and boundaries counting can be performed in parallel since data is accessed as read-only. No race condition exists here since we avoided possible nearby flippings. One disadvantage may be the computational cost of performing the polling iteratively until convergence, but we have observed that it converges in a few iterations. Thus, this algorithm is in practice fast enough.

8. Numerical experiments

In this section we will show two main numerical experiments. The first one shows the growth of a single grain where nucleation is allowed either. The second correspond to a set of four different initial distributions of stored energy where we also modify the quotient $\eta = \Delta\tau/\Delta t$. The Monte-Carlo nucleation is allowed every η time steps.

8.1. Letting a single grain grow

This numerical experiment is perform with 2000 grains. Only one grain located at the center of the domain has stored energy equal to 0 and all the other grains have stored energy equal to 0.6.

As shown in figure 10, the single grain starts growing and almost cover the whole domain. An interesting observation is that the number of sides of the grain growth quickly, reaching 34 sides. A similar microstructure can be seen in [8].

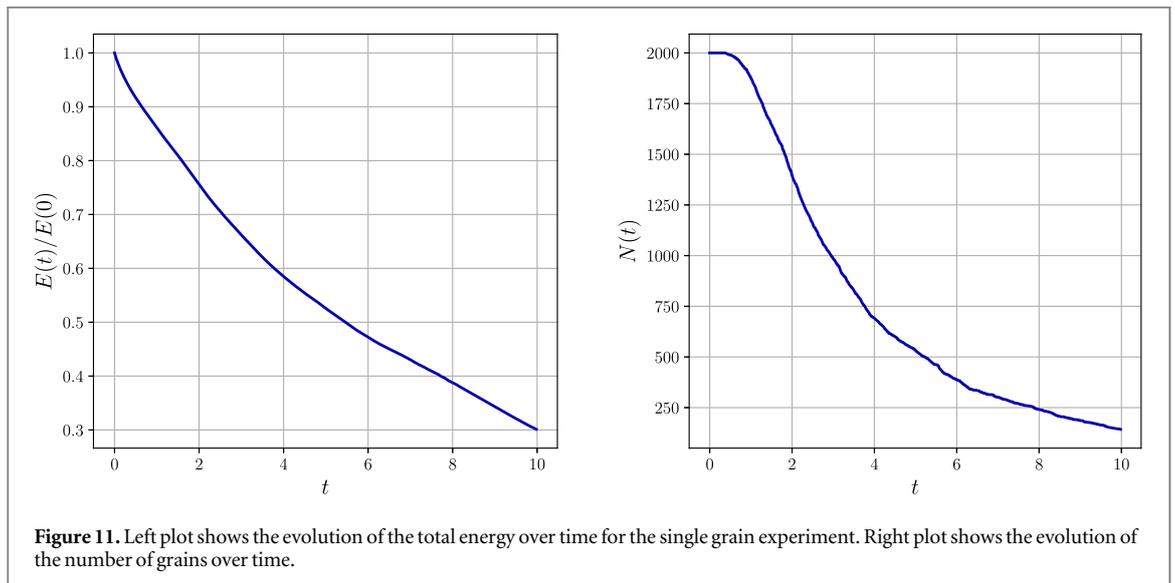
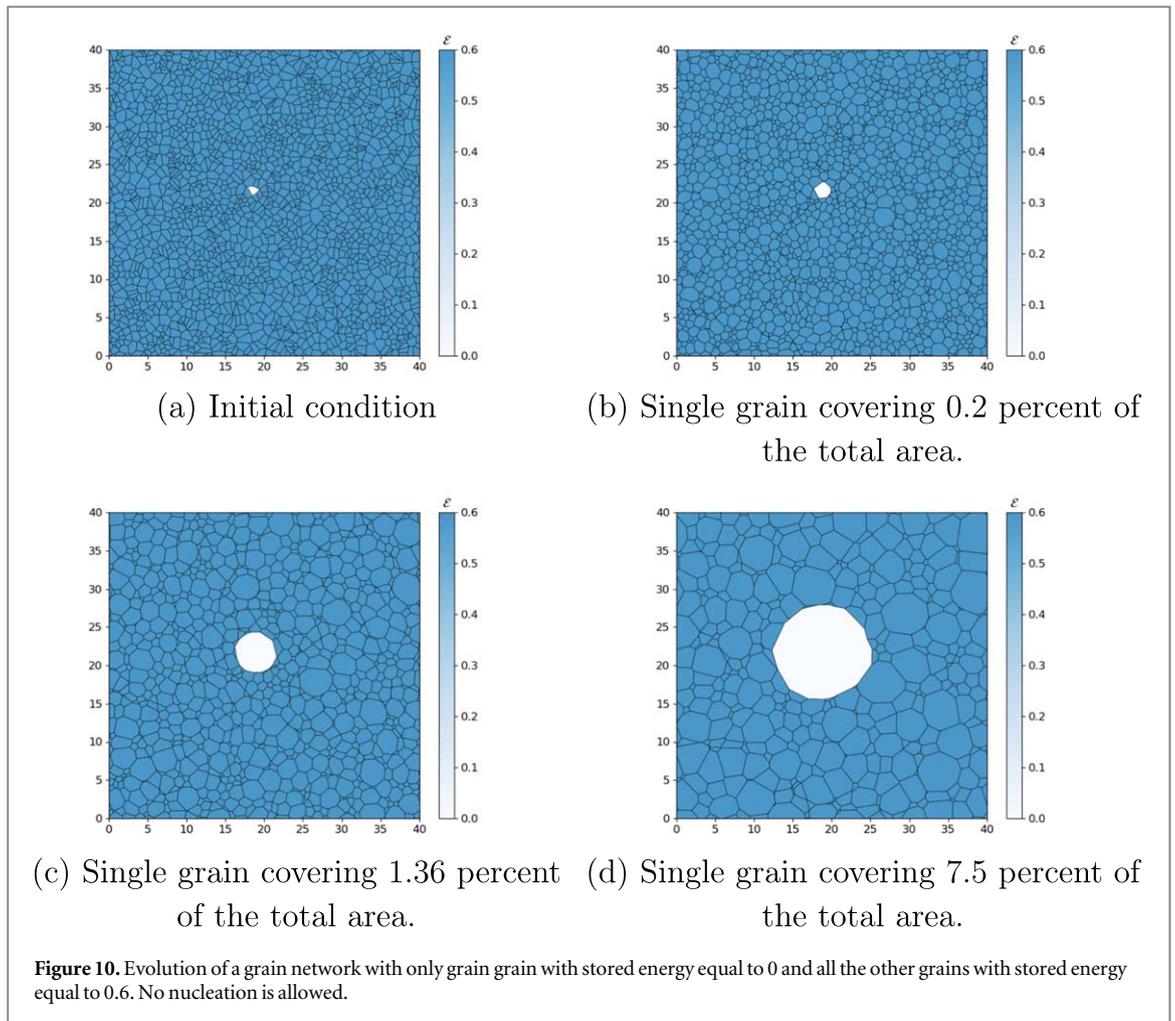
Figure 11 shows that the total energy of the system decreases as the single grain grows, accelerating the decay as time increases.

8.2. Set up for large scale nucleation with different initial stored energies

Numerical experiments were performed using 120000 initial grains following a Voronoi tessellation, but we allowed the nucleation to start at 100000 grains. This is because we need to have space in data structure to allow nucleation; otherwise, the nucleation will not be successful due to data structure constraints. The statistics will be computed from 100000 grains. Grain boundary energy anisotropy is added by using the following definition proposed by [43]:

$$\gamma(\Delta\alpha) = 1 + \frac{\varepsilon}{2}(1 - \cos^3(4\Delta\alpha)) \quad (9)$$

with $\varepsilon = 0.2$. The nucleated grain is built considering that the three new triple junctions are created along the grain boundaries associated to the selected triple junction. The locations of the new triple junctions correspond to a convex combination between the location (\mathbf{x}_i) of the candidate triple junction for nucleation and the location of each of its neighbor vertices (\mathbf{x}_j) along grain boundary Γ_j . The distance from the vertex where the



nucleated grain will be added to the new vertices is equal for the three cases, similarly to the analysis performed in section 5.2.

The initial distributions of stored energy are chosen to allow more grains with higher values of stored energy; thus, to make the nucleation more likely. We use the following initial distributions of stored energy:

SE-1 Constant stored energy equal to ν .

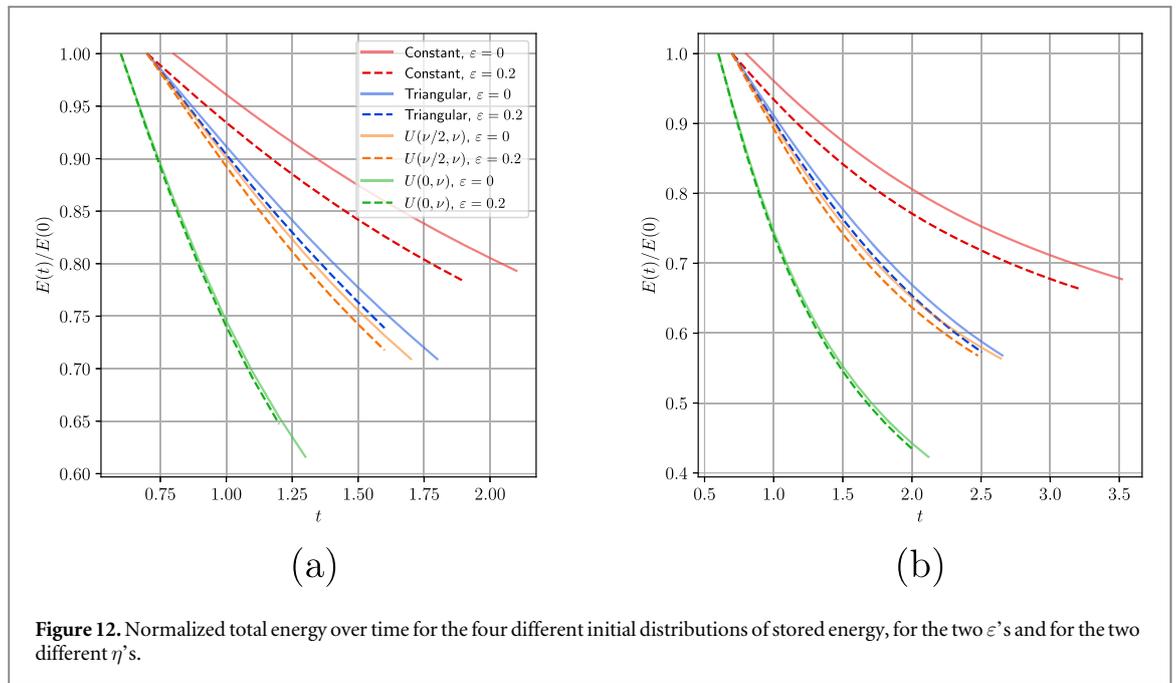


Figure 12. Normalized total energy over time for the four different initial distributions of stored energy, for the two ϵ 's and for the two different ν 's.

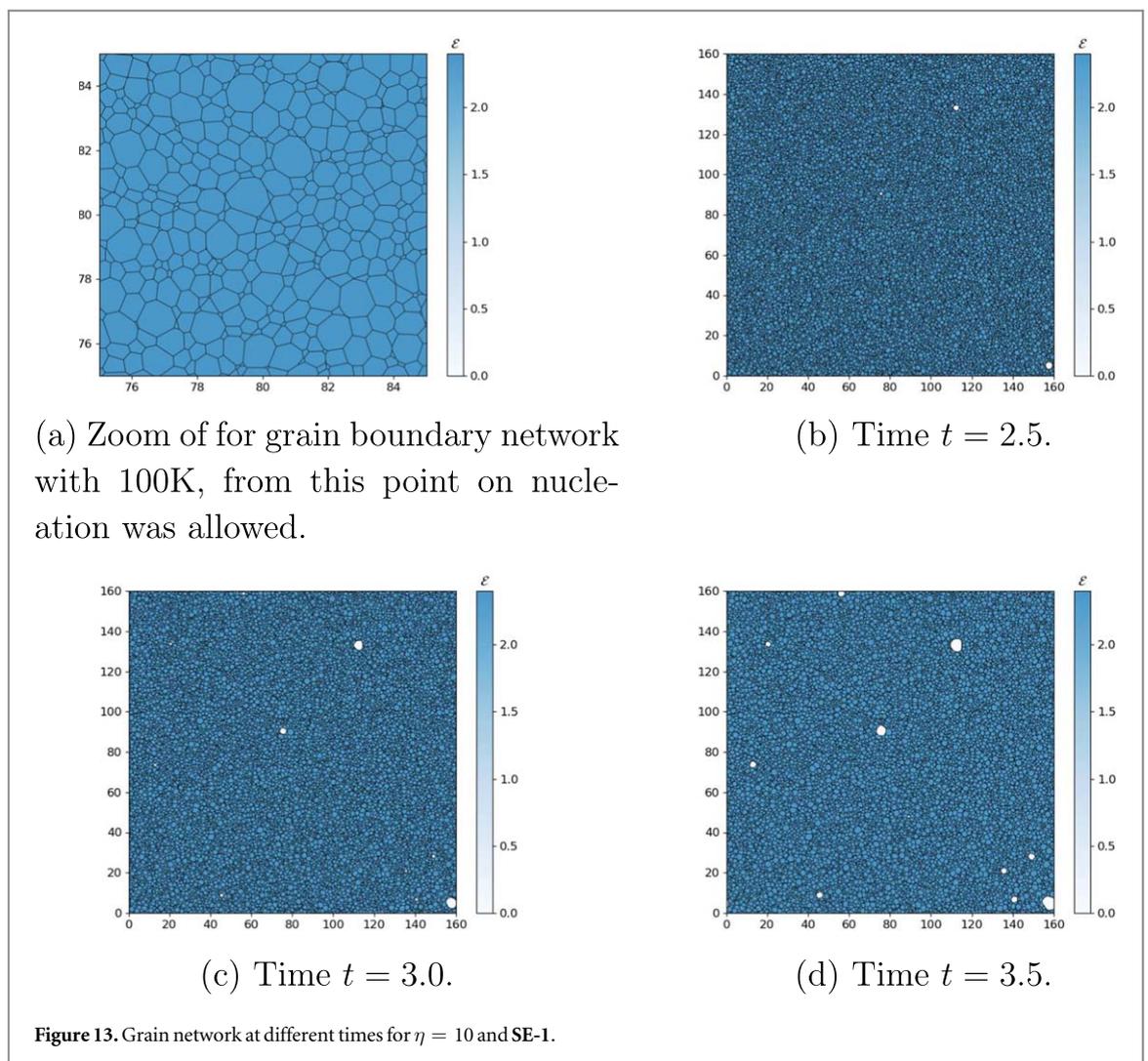
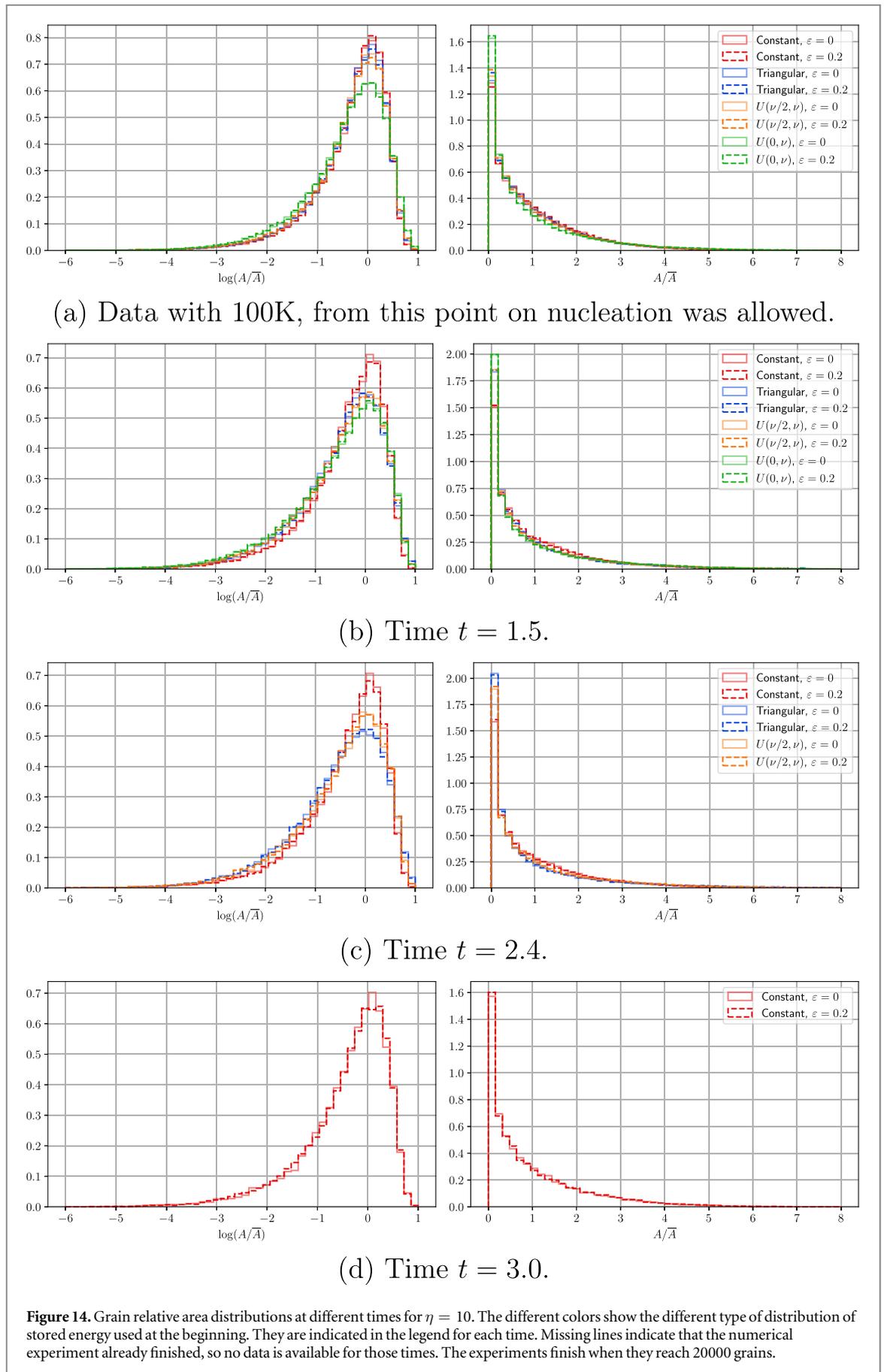
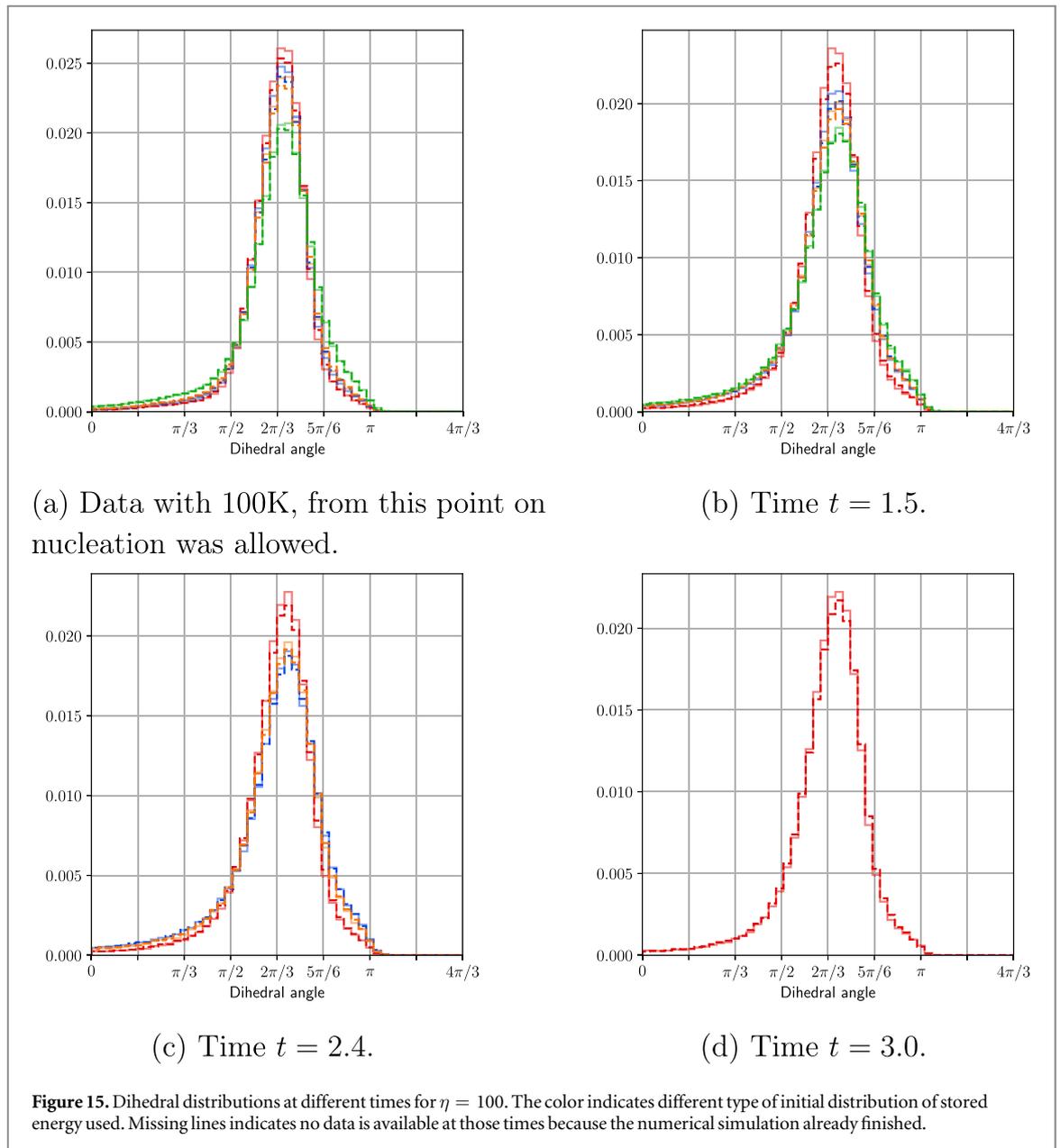


Figure 13. Grain network at different times for $\eta = 10$ and SE-1.





SE-2 Triangular distribution $\text{Triangular}(\nu/2, \nu, \nu)$.

SE-3 Uniformly randomly distributed in $[\nu/2, \nu]$.

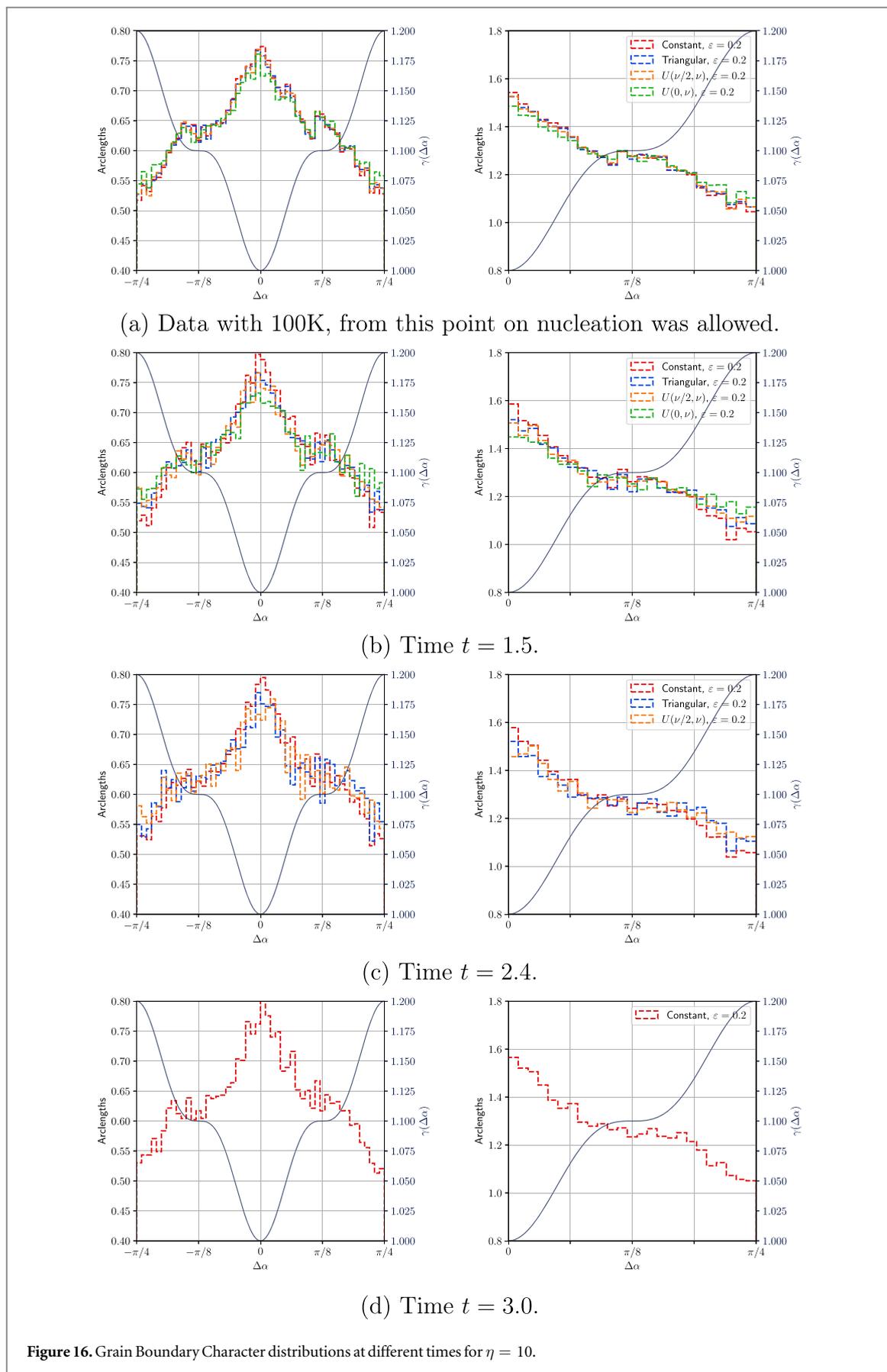
SE-4 Uniformly randomly distributed in $[0, \nu]$.

where $\nu = 2.4$. Similar values of \mathcal{E} were used in [27].

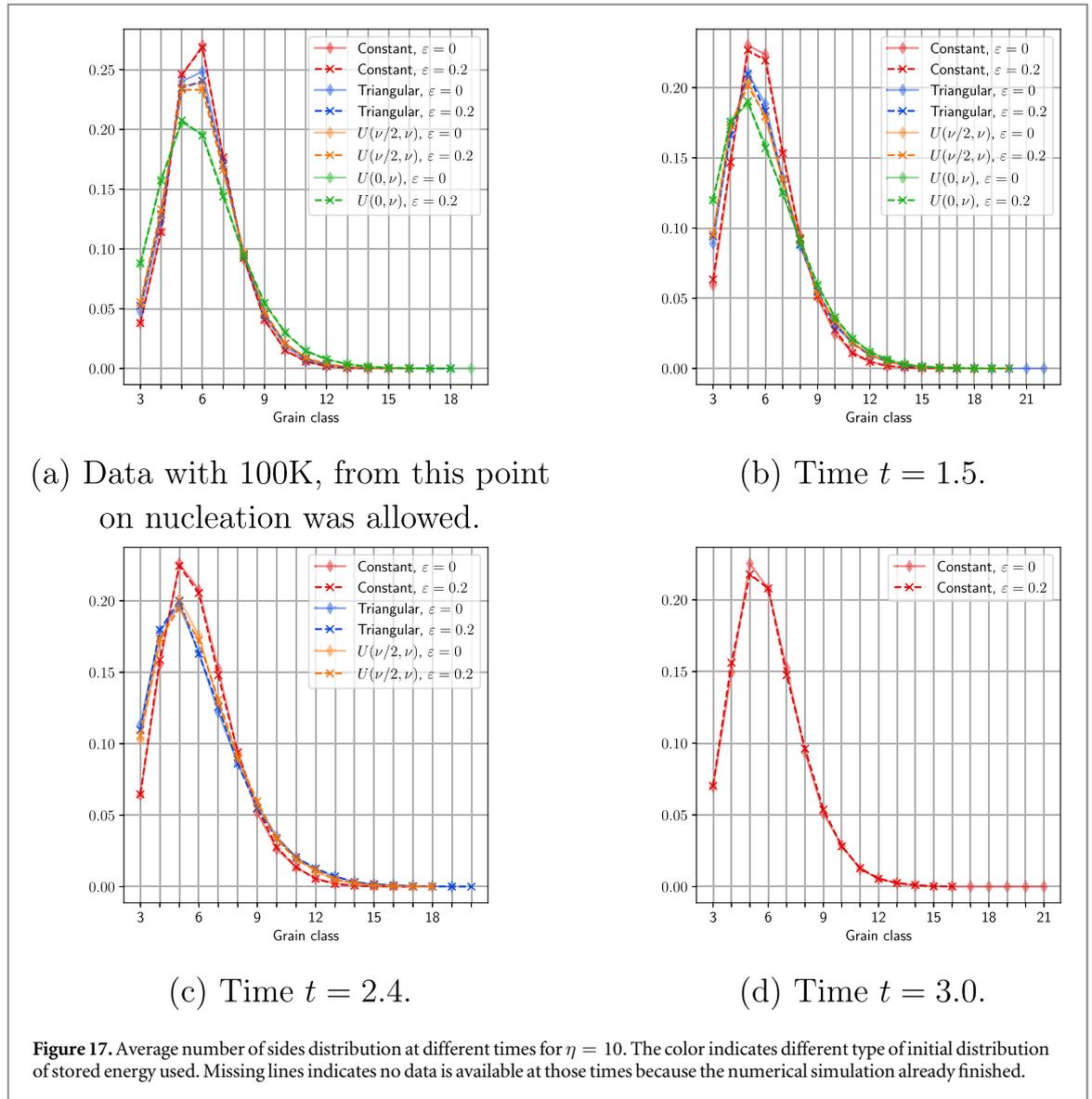
The maximum value of stored energy is ν , the grain boundary energy is in the range $[1, 1.2]$, and the computational domain used is $\Omega = [0, 160]^2$. We needed to increase the computational domain to be able to obtain $\Delta^2 E^A < 0$; otherwise, it would have not been possible to nucleate when many grains were present. We also consider periodic boundary conditions.

8.3. Energy minimization

Figure 12 shows the evolution of the total energy over time for all the initial distributions of stored energy. We also included the isotropic case, i.e. $\varepsilon = 0$, for reference. The continuous lines correspond to the isotropic cases and the dashed lines for the anisotropic one. The red curves correspond to the constant initial distribution of stored energy (**SE-1**), the blue curves correspond to the Triangular distribution $\text{Triangular}(\nu/2, \nu, \nu)$ (**SE-2**), the orange curves correspond to the uniform distribution $U(\nu/2, \nu)$ (**SE-3**), and the green curves correspond to the uniform distribution $U(0, \nu)$ (**SE-4**).



We note that there are three different evolution regimes for the total energies. The first correspond to SE-1, the second to SE-2 and SE-3, while the third corresponds to SE-4. This suggests that the value where the stored energy varies is more significant than its distribution. The numerical simulation were stopped when the dihedral



angles started to be significantly greater than π . This is because the algorithm is designed to work with small differences of stored energy. From now on, we only show the output for $\eta = 10$ since it was able to advance more in time than $\eta = 1$ without having dihedral angles much more greater than π . Also, the statistics look very similar.

8.4. Statistics for nucleation with MC

Figure 13 shows the grain structure for 4 different times for **SE-1**.

Figure 14 show the evolution of the distribution of the relative areas over time. Notice that at time $t = 2.4$, the data for **SE-4** is not shown because the numerical simulation reached the minimum number of grain, which is 20000 grains. The three regimes described before are still present in this plot. One may observe that the shape of the relative area distribution evolves further for the triangular and uniform initial stored energy distribution than it does in the constant case.

Figure 15 shows the distribution of the dihedral angle over time. We did not include the legend for visualization purpose, the legend from figure 14 is still valid. In this case, by time $t = 3$, several experiments have already reached the minimal number of grains and are not shown in the plots.

Figure 16 shows the development of the Grain Boundary Character Distribution (GBCD).

Figure 17 shows the average number of sides distribution. In this figure, we observe again that the initial distribution of stored energy **SC-1** behaves different with respect to the other numerical experiments. For time $t = 3.0$, we observe that we get grains with more than 20 sides for **SC-1**. The other feature observed is the proportion of 3-sided grains at times $t = 1.5$ and $t = 2.4$ for the numerical experiments **SC-2**, **SC-3** and **SC-4**, which is greater than the proportion for **SC-1**.

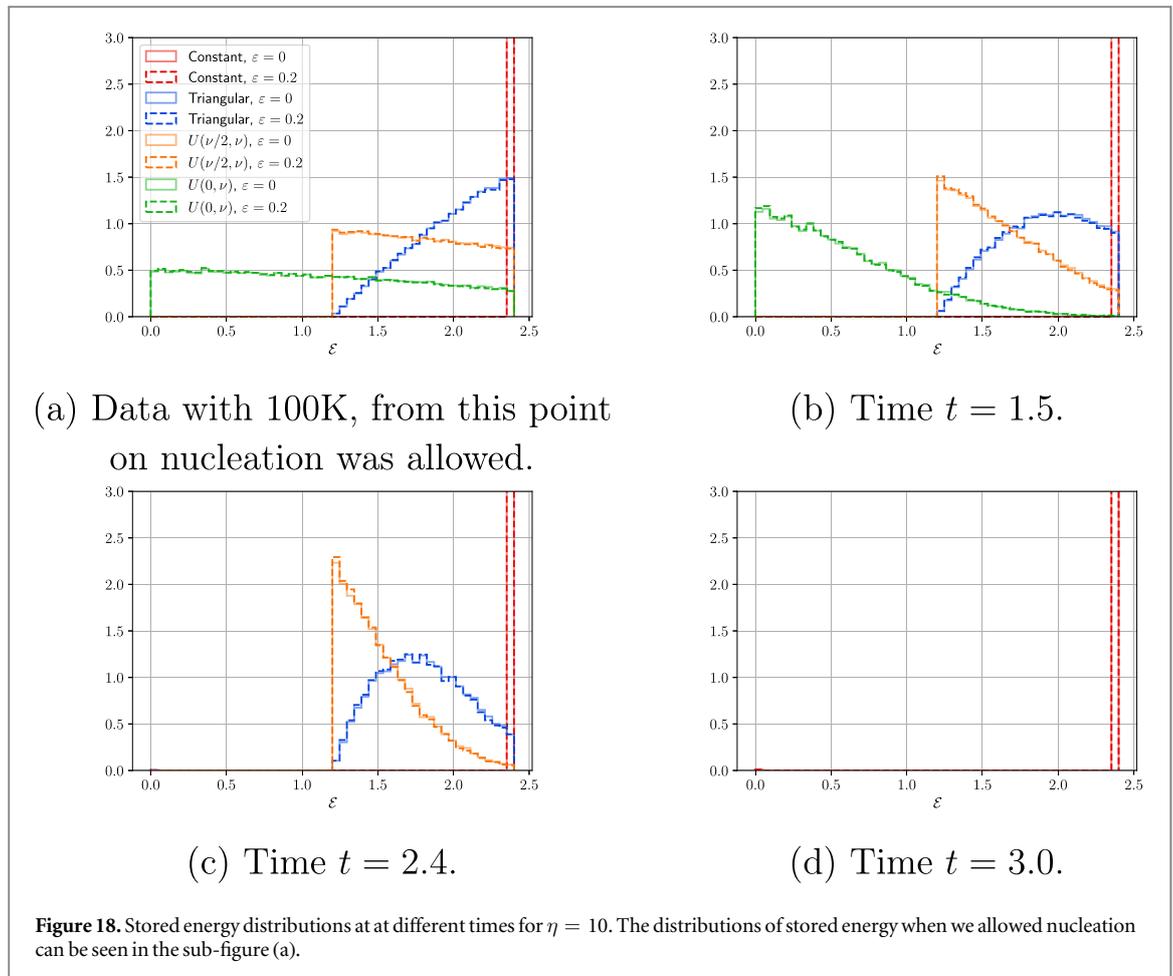


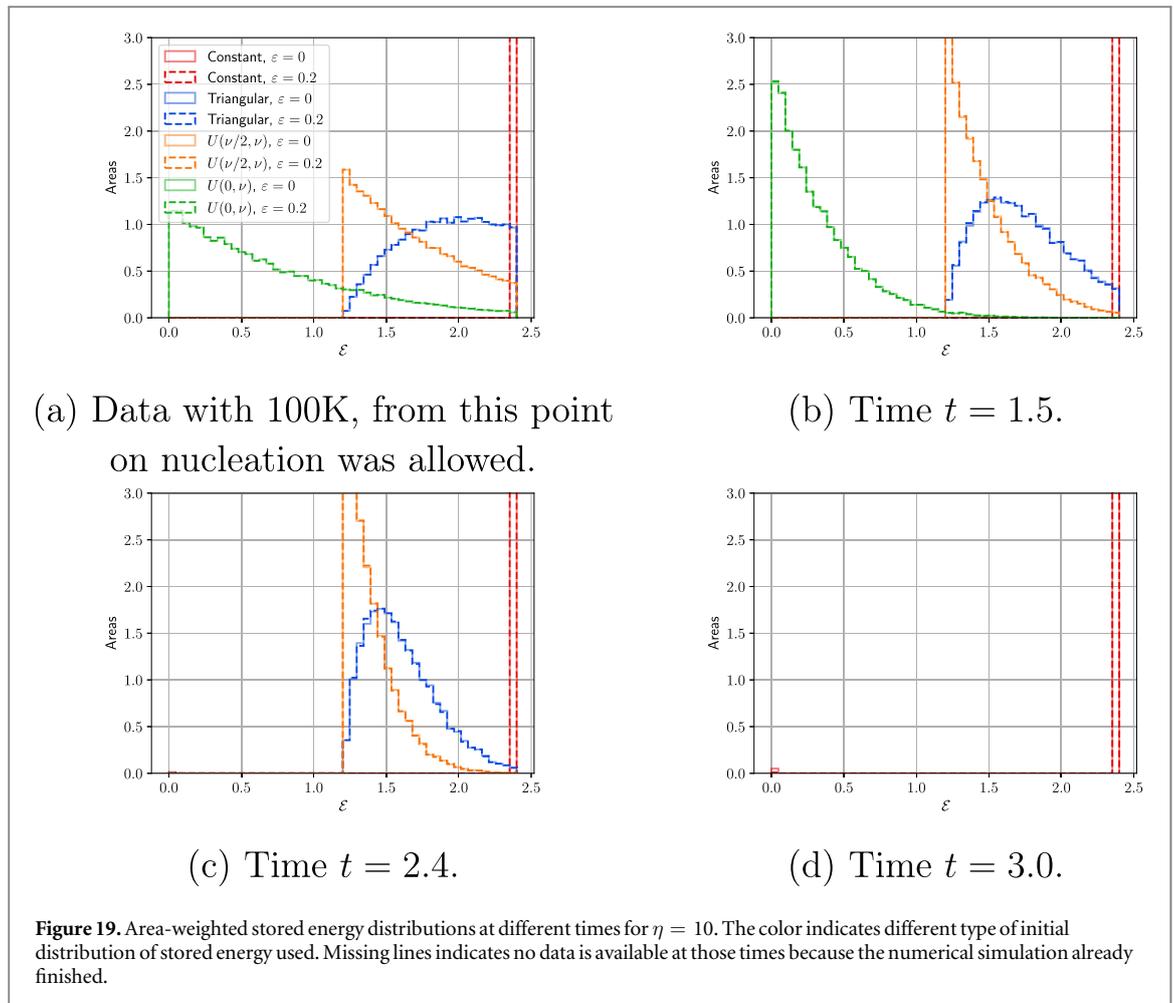
Figure 18 shows the normalized histogram for the stored energy and figure 19 shows the area-weighted histogram for the stored energy. In both cases, we observe the distributions for stored energy right when we started allowing nucleation for each numerical experiment SE-1, SE-2, SE-3, and SE-4. Notice that the y -axis has been limited to 3 but the red-bin is much larger than 3. This was added for visualization purposes. As observed in both cases, the lower values of stored energy are preferred; thus, they become more important as time evolves.

Misorientation distribution function (MDF) is shown in figure 20. It shows the preference to misorientations close to zero from the beginning.

Tables 1 and 2 show the number of successful and unsuccessful attempts to nucleate a grain. This is done for the isotropic and anisotropic settings for the initial conditions SE-1, SE-2, SE-3 and SE-4 with $\eta = 10$. The nucleations attempts labelled as ‘R’ show the number of rejected attempt, they are rejected because $\Delta E^\Delta > 0$ and $\Delta^2 E^\Delta > 0$; thus, an unfeasible site for successful nucleation. The nucleation attempts labeled as ‘MC-R’ are feasible since $\Delta E^\Delta > 0$ and $\Delta^2 E^\Delta < 0$, but they were rejected by the Monte-Carlo algorithm 1. Similarly, ‘MC-A’ shows the number of nucleation site with $\Delta E^\Delta > 0$ and $\Delta^2 E^\Delta < 0$ and that were accepted by the Monte-Carlo algorithm 1. Attempts labeled as ‘A’ indicate the sites where the nucleation was accepted immediately since $\Delta E^\Delta < 0$.

We observe that nucleation attempts decreases from SE-1 to SE-4. Anisotropic settings seem to reduce the number of successful attempts. Note that most of the rejected nucleation comes from unfeasible sites, whilst all the accepted nucleations come from the Monte-Carlo algorithm.

Figure 21 shows the average area over time and the percentage of nucleated area over time for each initial distribution of stored energy. The same color configuration described before is used for clarity. The first column is for $\eta = 1$ and the second column for $\eta = 10$. Similarly as before, the curves are plotted until the number of grains is 20000. Thin curves (continuous and dashed) correspond to the average area over time, this is shown on the left y -axis. Thick curves (continuous and dashed) correspond to the percentage of area of nucleated grains, this is shown on the right y -axis. Notice that for sub-figures (d), (f)–(h), the thicker curves are very close to 0. As it can be observed in figure 21(a), the constant distribution of stored energy was the numerical experiment that was able to reach the largest percentage of area nucleated. All the other cases were less significant and reached the minimal number of grains before they would have achieved a higher percentage of nucleated area.



9. Conclusions

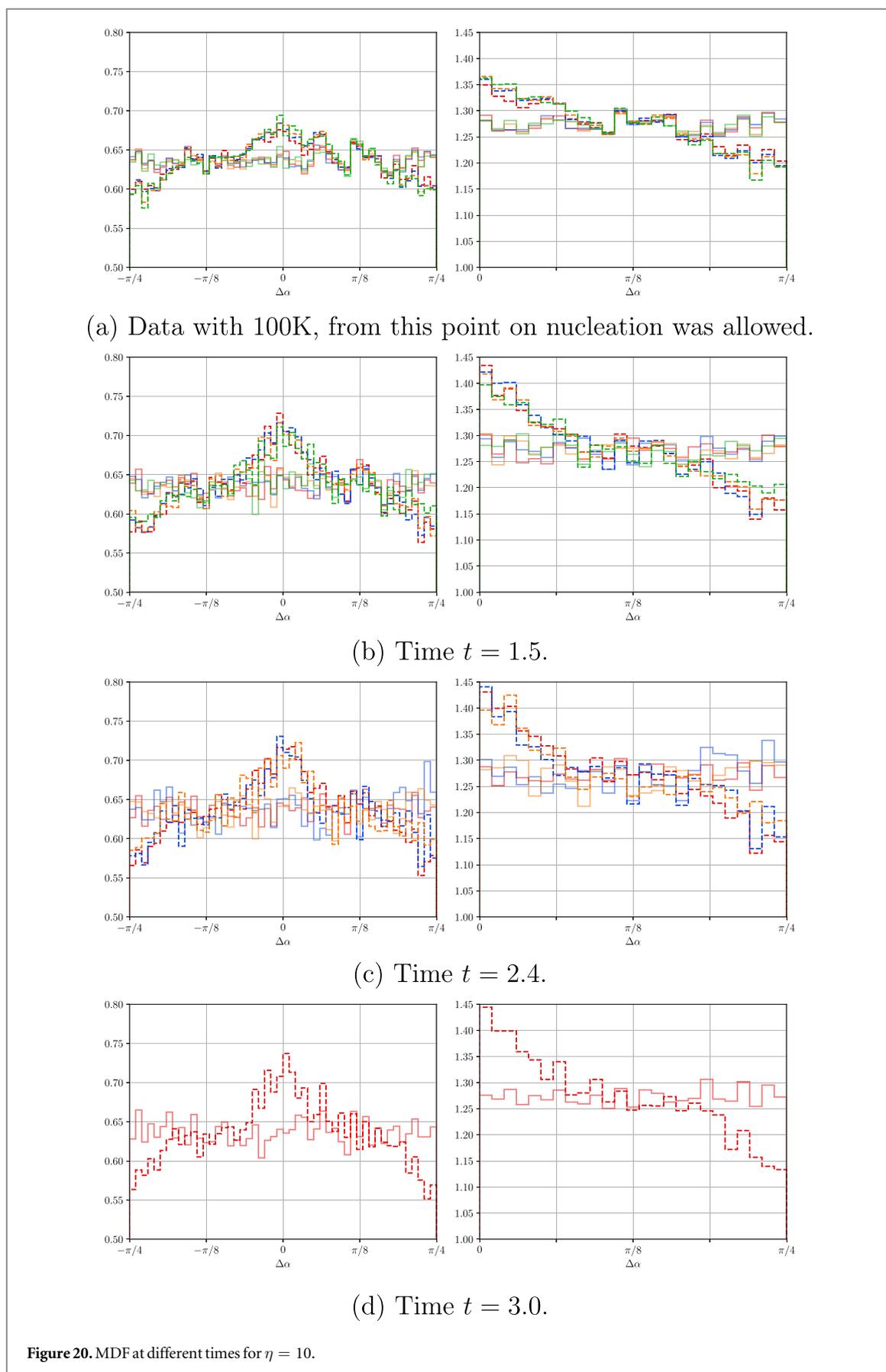
In this work we extended continuous formulation of the two-dimensional vertex model to incorporate stored energy of the grains. The basic mechanism has been previously used by several authors, but not in the context of the vertex model formulation adopted herein. We provide complete description of the Monte Carlo method used in the process of choosing the nucleation site and analyze conditions under which the nucleus can grow. We observed that while nucleated grains may add energy to the system, they eventually decrease the total energy in the course of the system evolution.

An extensive set of statistics is harvested and compared for four different initial distributions of stored energy. Our numerical experiments confirm the expected behavior for the GBCD and the emergence of the long tail for relative area distribution. While the statistics in this slow nucleation regime do not change shape when compared to the pure grain growth phase, we noticed dependence of the distributions on the range of the initial stored energy. For instance, for higher average value of stored energy (**SE-1**) there are fewer grains with small areas than one expects to find in the relative area statistics.

One of the distinguishing features of our vertex model is that it allows for parallel implementation. In building the GPU framework we overcame several computational challenges, such as the parallel management of the topological transitions and the efficient computation of the dynamics of the network of grains. Parallelization of the random selection of candidate vertices in multiple threaded architecture has also been successfully implemented. To support reproducibility, we make the CUDA code available in [GitHub](#), see [44].

10. Future work

The numerical analysis presented here shows that we can let grains grow with a stored energy term in a stable way, that is, avoiding vertices and boundaries crossing each other. For large differences of the stored energy between neighboring grains, we may have triple junctions with dihedral angles that exceed π . These triple junctions are precisely those that could cross one of the opposite boundaries in a given grain. The resulting



topological transition can be handled with an extension of the present continuous vertex model by including a different type of topological transition, called TA, see [1, 34]. As future work, we would like to incorporate this topological transition into our computational model. Larger difference of stored energy will be implemented in

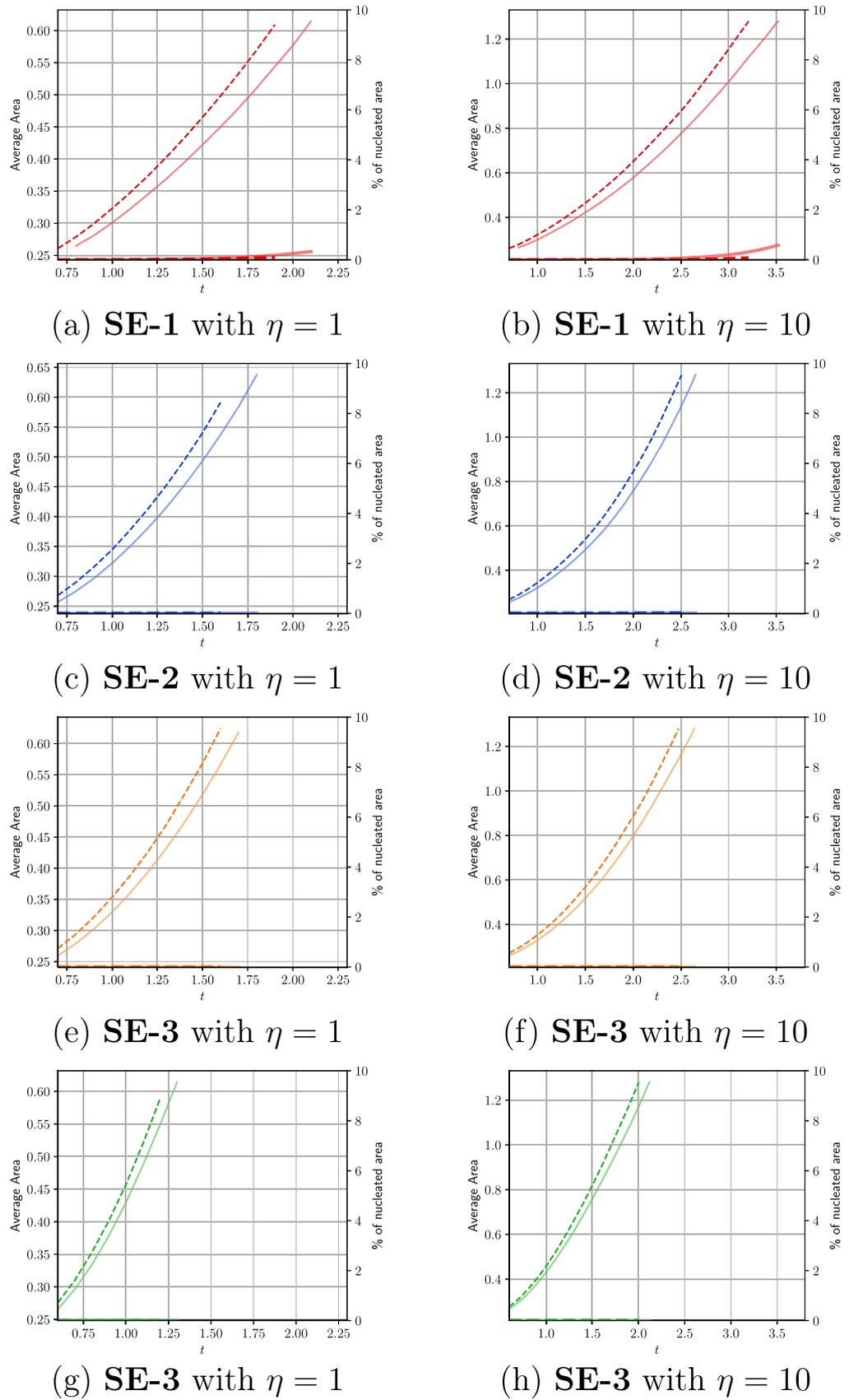


Figure 21. Average area and percentage of nucleated area over time. The color indicates different type of initial distribution of stored energy used. Thin curves (continuous and dashed) correspond to the average area over time, this is shown on the left y-axis. Thick curves (continuous and dashed) correspond to the percentage of area of nucleated grains, this is shown on the right y-axis. Notice that for sub-figures (d), (f), (g), and (h), the thicker curves are very close to 0.

Table 1. Nucleation results per experiment with $\eta = 1$.

Experiment	R	MC-R	MC-A	A	Nucleation attempts
Constant, $\varepsilon = 0$	2816	30	72	0	2918
Constant, $\varepsilon = 0.2$	2613	13	33	0	2659
Triangular, $\varepsilon = 0$	2401	4	4	0	2409
Triangular, $\varepsilon = 0.2$	2126	0	2	0	2128
$U(\nu/2, \nu), \varepsilon = 0$	2240	1	0	0	2241
$U(\nu/2, \nu), \varepsilon = 0.2$	2152	1	0	0	2153
$U(0, \nu), \varepsilon = 0$	1673	0	0	0	1673
$U(0, \nu), \varepsilon = 0.2$	1541	0	0	0	1541

Table 2. Nucleation results per experiment with $\eta = 10$.

Experiment	R	MC-R	MC-A	A	Nucleation attempts
Constant, $\varepsilon = 0$	530	4	21	0	555
Constant, $\varepsilon = 0.2$	485	11	12	0	508
Triangular, $\varepsilon = 0$	385	0	6	0	391
Triangular, $\varepsilon = 0.2$	372	1	1	0	374
$U(\nu/2, \nu), \varepsilon = 0$	392	1	0	0	393
$U(\nu/2, \nu), \varepsilon = 0.2$	371	0	0	0	371
$U(0, \nu), \varepsilon = 0$	311	0	0	0	311
$U(0, \nu), \varepsilon = 0.2$	295	0	0	0	295

parallel, for instance, by extending the polling system proposed here. Another future research direction is to study the effect of initial grain distribution on steady state characteristics.

Acknowledgments

This work has been partially funded by—CCTVal, CONICYT PIA/Basal FB0821, FONDECYT 11160744. AS acknowledges the support of the Dirección de Postgrado y Programas, UTFSM, Chile. ME acknowledges support provided by the US National Science Foundation CAREER grant DMS-1056821. We also give thanks to the outstanding student Pablo Ibarra for his contributions.

ORCID iDs

Claudio E Torres  <https://orcid.org/0000-0003-3548-4773>

References

- [1] Piękoś K, Tarasiuk J, Wierzbowski K and Bacroix B 2008 Stochastic vertex model of recrystallization *Comput. Mater. Sci.* **42** 36–42
- [2] Van Leeuwen C and Van Der Eerden J P 1977 Nucleation growth processes: a Monte Carlo simulation *Surf. Sci.* **64** 237–50
- [3] Humphreys J, Rohrer G S and Rollett A 2017 *Recrystallization and Related Annealing Phenomena* 3rd edn (Amsterdam: Elsevier)
- [4] Doherty R D, Hughes D A, Humphreys F J, Jonas J J, Juul Jensen D, Kassner M E, King W E, McNelley T R, McQueen H J and Rollett A D 1997 Current issues in recrystallization: a review *Materials Science and Engineering: A* **238** 219–74
- [5] Rollett A D 1997 Overview of modeling and simulation of recrystallization *Prog. Mater. Sci.* **42** 79–99
- [6] Orend J, Hagemann F, Klose F B, Maas B and Palkowski H 2015 A new unified approach for modeling recrystallization during hot rolling of steel *Materials Science and Engineering A* **647** 191–200
- [7] Radhakrishnan B, Sarma G B and Zacharia T 1998 Modeling the kinetics and microstructural evolution during static recrystallization Monte Carlo simulation of recrystallization *Acta Mater.* **46** 4415–33
- [8] Holm E A, Miodownik M A and Rollett A D 2003 On abnormal subgrain growth and the origin of recrystallization nuclei *Acta Mater.* **51** 2701–16
- [9] Zhang J-X, Wen H and Liu Y-T 2011 Monte Carlo model in metal recrystallization simulation *Journal of Shanghai Jiaotong University (Science)* **16** 337–42
- [10] Cetinel H, Kayacan O and Ozaydin D 2013 Investigation of nucleation and grain growth in 2-dimensional systems by using generalized Monte Carlo simulations *Physica A: Statistical Mechanics and its Applications* **392** 4121–6
- [11] Luo X, Zhao W, Liu B S and Y M Zhang 2011 A feasible parallel Monte Carlo algorithm to simulate templated grain growth *Advanced Materials Research* **332–334** 1868–71
- [12] Moelans N, Godfrey A, Zhang Y and Juul Jensen D 2013 Phase-field simulation study of the migration of recrystallization boundaries *Phys. Rev. B* **88** 054103
- [13] Gentry S P and Thornton K 2015 Simulating recrystallization in titanium using the phase field method *IOP Conference Series: Materials Science and Engineering* **89** 012024

- [14] Podmaniczky F, Tóth G I, Pusztai T and Gránásy L 2016 Investigating nucleation using the phase-field method *J. Indian Inst. Sci.* **96** 161–78 <https://hungary.pure.elsevier.com/hu/publications/investigating-nucleation-using-the-phase-field-method>
- [15] Takaki T, Ohno M, Shibuta Y, Sakane S, Shimokawabe T and Aoki T 2016 Two-dimensional phase-field study of competitive grain growth during directional solidification of polycrystalline binary alloy *J. Cryst. Growth* **442** 14–24
- [16] Shi X, Huang H, Cao G and Ma X 2017 Accelerating large-scale phase-field simulations with GPU *AIP Adv.* **7** 105216
- [17] Miyoshi E, Takaki T, Ohno M, Shibuta Y, Sakane S, Shimokawabe T and Aoki T 2017 Ultra-large-scale phase-field simulation study of ideal grain growth *NPJ Computational Materials* **3** 25
- [18] Bernacki M, Resk H, Coupez T and Logé R E 2009 Finite element model of primary recrystallization in polycrystalline aggregates using a level set framework *Modelling Simul. Mater. Sci. Eng.* **17** 064006
- [19] Hallberg H 2013 A modified level set approach to 2D modeling of dynamic recrystallization *Modelling Simul. Mater. Sci. Eng.* **21** 085012
- [20] Bernacki M, Logé R E and Coupez T 2011 Level set framework for the finite-element modelling of recrystallization and grain growth in polycrystalline materials *Scr. Mater.* **64** 525–8
- [21] Mießen C, Velinov N, Gottstein G and Barrales-Mora L A 2017 A highly efficient 3D level-set grain growth algorithm tailored for ccNUMA architecture *Modelling Simul. Mater. Sci. Eng.* **25** 084002
- [22] Elsemy M, Esedoglu S and Smereka P 2011 Large-scale simulations and parameter study for a simple recrystallization model *Phil. Mag.* **91** 1607–42
- [23] Ilin D N, Bozzolo N, Toulorge T and Bernacki M 2018 Full field modeling of recrystallization: Effect of intragranular strain gradients on grain boundary shape and kinetics *Comput. Mater. Sci.* **150** 149–61
- [24] Kühbach M, Barrales-Mora L A and Gottstein G 2014 A massively parallel cellular automaton for the simulation of recrystallization *Modelling Simul. Mater. Sci. Eng.* **22** 075016
- [25] Zhang T, Lu S H, Zhang J B, Li Z F, Chen P, Gong H and Wu Y X 2017 Modeling of the static recrystallization for 7055 aluminum alloy by cellular automaton *Modelling Simul. Mater. Sci. Eng.* **25** 0–16
- [26] Han F, Tang B, Kou H, Li J and Feng Y 2015 Cellular automata simulations of grain growth in the presence of second-phase particles *Modelling Simul. Mater. Sci. Eng.* **23** 065010
- [27] Piękoś K, Tarasiuk J, Wierzbowski K and Bacroix B 2005 Recrystallization study using two-dimensional vertex model *Archives of Metallurgy and Materials* **50** <http://yadda.icm.edu.pl/yadda/element/bwmeta1.element.baztech-article-BSW3-0014-0007>
- [28] Piękoś K, Tarasiuk J, Wierzbowski K, Gerber P and Bacroix B 2005 Monte-Carlo version of vertex model applied to recrystallization modeling *Mater. Sci. Forum* **495–497** 1183–8
- [29] Nagai T, Ohta S, Kawasaki K and Okuzono T 1990 Computer simulation of cellular pattern growth in two and three dimensions *Phase Transit.* **28** 177–211
- [30] Weygand D, Brechet Y and Lepinoux J 1998 A vertex dynamics simulation of grain growth in two dimensions *Phil. Mag. B* **78** 329–52
- [31] Piękoś K, Tarasiuk J, Wierzbowski K and Bacroix B 2008 Generalized vertex model of recrystallization—application to polycrystalline copper *Comput. Mater. Sci.* **42** 584–94
- [32] Mellbin Y, Hallberg H and Ristinmaa M 2014 Accelerating crystal plasticity simulations using GPU multiprocessors *Int. J. Numer. Methods Eng.* **100** 111–35
- [33] Mellbin Y, Hallberg H and Ristinmaa M 2015 A combined crystal plasticity and graph-based vertex model of dynamic recrystallization at large deformations *Modelling Simul. Mater. Sci. Eng.* **23** 045011
- [34] Mellbin Y, Hallberg H and Ristinmaa M 2017 An extended vertex and crystal plasticity framework for efficient multiscale modeling of polycrystalline materials *Int. J. Solids Struct.* **125** 150–60
- [35] Gómez A H J S, Pablo I S, Ariel S R, Casas F J, Torres C E, Emelianenko M and Golovaty D 2018 Evolution of two-dimensional grain boundary networks implemented in GPU *Comput. Mater. Sci.* **160** 315–33 submitted
- [36] Torres C E, Emelianenko M, Golovaty D, Kinderlehrer D and Ta'asan S 2015 Numerical analysis of the vertex models for simulating grain boundary networks *SIAM J. Appl. Math.* **75** 762–86
- [37] Piękoś K, Tarasiuk J, Wierzbowski K and Bacroix B 2004 Development of two-dimensional vertex type model *Mater. Sci. Forum* **467–470** 653–8
- [38] Zhao P and Guo Y 2018 Grain size effects on indentation-induced defect evolution and plastic deformation mechanism of polycrystalline materials *Comput. Mater. Sci.* **155** 431–8
- [39] Fan Z, Wu Y, Zhao X and Lu Y 2004 Simulation of polycrystalline structure with Voronoi diagram in Laguerre geometry based on random closed packing of spheres *Comput. Mater. Sci.* **29** 301–8
- [40] Rollett A D and Manohar P 2006 The Monte Carlo method *Continuum Scale Simulation of Engineering Materials: Fundamentals—Microstructures - Process Applications* ed D Raabe, F Roters, F Barlat and L-Q Chen (Hoboken, NJ: Wiley) pp 76–113 ch 4
- [41] Nickolls J, Buck I, Garland M and Skadron K 2008 Scalable parallel programming with CUDA *Queue* **6** 40–53
- [42] NVIDIA Corporation 2018 NVIDIA accelerated computing, CUDA zone <https://developer.nvidia.com/cuda-zone>, Accessed: 2018-09-23
- [43] Kinderlehrer D, Livshits I and S Ta'asan 2006 A variational approach to modeling and simulation of grain growth *SIAM J. Sci. Comput.* **28** 1694–715
- [44] Sazo Gómez A H J, Torres C E, Emelianenko M and Golovaty D 2018 Vertex model with stored energy and nucleation in GPU source code <https://github.com/tclaudioe/SE-grain-growth-GPU>, Accessed: 2018-10-30