PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à MINES ParisTech

Towards highly efficient massive-multidomain simulations in the context of microstructural evolutions

Soutenue par
Sebastian FLOREZ
Le 30 Novembre, 2020

École doctorale n°264
ED SFA

Spécialité
Mécanique Numérique et Matériaux

Composition du jury :

Jean-François REMACLE Professeur, École Polytechnique de Louvain	*Rapporteur*
Julien BRUCHON Professeur, École Nationale Supérieure des Mines de Saint-Étienne	*Rapporteur*
Joëlle DEMURGER Ingénieure de Recherche Senior, ASCOMETAL	*Examinateur*
Luis BARRALES-MORA Professeur Associé, Georgia Tech Lorraine	*Examinateur*
Modesar SHAKOOR Maître assistant, Institut Mines-Télécom Lille Douai	*Examinateur*
Hugues DIGONNET Chargé de recherche, École Centrale de Nantes	*Examinateur*
Thomas TOULORGE Dr. Ingénieur de Recherche Senior, Cenaero	*Membre invité*
Pascal DE MICHELI Dr. Ingénieur R&D, Transvalor SA	*Membre invité*
Marc BERNACKI Professeur, PSL MINES Paristech	*Directeur de thèse*

MINES ParisTech | PSL

Aknowledgements

After finishing all the other contents of this manuscript, I have spent the last hour trying to figure out how to start this acknowledgments section. I have to say that thanking all the people who have had an impact in the works of this thesis and in the path I took so I could do this work, is not an easy task.

I know I must thank my supervisor Marc Bernacki for all the support he has given me during the past 3 years, it is just that I do not how to express how thankful I am to him, for being such a great person and leader, able to give wise counsel in any aspect of life (work-life and *the rest of it*) and generous enough to share that wisdom with his PhD students[1]. So, Marc, I will only say, thank you for being such a good mentor.

I would also like to extend my gratitude to Thomas Toulorge, with whom I had the pleasure to exchange during the beginning of my PhD thesis[2]. Thank you very much for being there during this period and for being interested in my work even after. I really appreciate your advice and all your ideas, they were very helpful during the implementation of my works.

It is also very difficult to put into words the feeling of gratitude I want to express to my family, my Dad, the most tenacious and perseverant man in the world[3], but also the most understanding father I could have asked for[4]. Thank you, Dad. My Mom, who taught me to always give the best of me in any matter of life and so I did, thank you, Mom. My sisters, Bianny, the spoiler, Lule, the one with common sense, and Lady, the one without it. Thank you sisters for pampering me during all these years and also for all those occasions when you have put my interests before your own, I am in your eternal debt. My brother in law, Nelson, thank you for being there for Lule and me, always with your big smile and your good sense of humor.

[1]Even if that wisdom is usually given in the form of french proverbs that you would have to look up in google to understand them. Just kidding, thank you very much Marc.

[2]Thomas told me one afternoon that we would go for some cigarettes. He never came back. I knew he would not, he does not smoke!

[3]Almost stubborn when it comes to restoring vintage cars.

[4]Even when I crashed one of his vintage cars.

To my colleagues and friends, Ludovic, Brayan, Julien, Baptiste, and Alexis, thank you for all the laughs, all the discussions, all your help, and all the burgers. It has been a pleasure to share my time at CEMEF with you, I would not change a bit of it.

To my friends, Andrei and Sergio, all the memories that I have with you (those from our childhood and those from now) have been and will continue to be, a highly influential part of my life. We are but our memories and I am happy to share a big part of them with you.

Karen, my colleague, my friend, my love. Mori, I most certainly do not know the words that would express my gratitude towards you, and I do not expect to find them soon enough. You know however that this work would have been impossible if not were for you, for your support, for your contagious tenacity to accomplish anything, for making me remember that this part of life (the work part) is only as good as *the rest of it*. I am glad that *the rest of it* I got to share it with you, I love you.

Introduction

The ability of the human mind to create new forms of technology is rather exceptional, take for instance the works presented in [1, 2] where the basis of what was called "a-machines" (later called Turing Machines) was first described, or the analogous works given in [3] where the notions of λ-calculus were introduced. Both works independently answered to the Entscheidungsproblem (the decision problem) [4] as unsolvable. Although the proof of unsolvability of this problem has had a great impact on the understanding of the limits of algorithms, these works have also set the foundation of modern digital computing which is responsible for the exponential development of our technologies, our society, and the way we perceive the world and universe today.

Of course, one of the many domains that owes its success to digital computing and computer science is the field of numerical simulations, which have helped to revolutionize the engineering of new devices and systems and to recreate physical situations/problems that would be very difficult (near impossible) to manage by conventional mathematical approaches. The complexity of modern engineering is also in constant growth as a response to the progressive demand for more efficient and sustainable systems. Here our society becomes every day more dependent on computer science and numerical simulations (even though it is still a very reserved topic for some computer enthusiasts, engineers, and scientists) and is therefore accompanied by a necessary enhancement of their performance.

Generally, two strategies allow increasing the efficiency of a computer-based numerical simulation: the first consists in enhancing the hardware used to perform such simulation (hardware optimization) and the second, to improve the algorithms behind the simulation without harming the accuracy of its predictions (software optimization). Many efforts in the last few decades have been concentrated on increasing the clock speed of microprocessors. Continuous advances in hardware in the last few years have been focused on reducing the cost and increasing the power efficiency of microprocessors. Usually, an increase of the transistor density allows for a lower power consumption (performing more instructions without getting too hot), which usually limits the CPU performance. Also, a higher transistor density can provide more cores per chip, allowing higher performance in multitasking or enabling more threads for applications optimized to run in a parallel context. At the time of writing, last generation high-end com-

puters (accessible to the general public) include processors with up to 64 cores, achieving speeds of up to 4.5 GHz, and Random Access Memory (RAM) slots with a maximum size of 32 Gb (although most processors only support RAM slots of up to 16 Gb of size).

Although hardware optimizations rely exclusively on the hands of the lead companies in the sector, software optimizations for computer-based numerical simulations are scattered around the private and public domains, and are generally produced on demand for specific applications. Of course, a wide range of software improvements can be done to improve the global efficiency of a computational procedure. Improvements on the Operative System (OS) or in the compiler/interpreter of the algorithm in question can produce huge improvements in its CPU-time. However, these aspects are usually taken as an accessory on the enhancement (or in the creation of new) numerical strategies where the main endeavor still consists in upgrading the main numerical algorithms responsible for the simulation.

One example of a framework that has constantly reach its limits in terms of computational performance is the field of massive multi-domain simulations. This terminology can be understood as the modeling of physical phenomena in a given domain partitioned in a large number of sub-domains, each presenting its own characteristics. Another vision may be that of the evolution of complex interface networks in a domain with heterogeneous properties depending on these interfaces. A field of predilection for these kinds of simulations is that of materials science. Indeed, almost all the materials that surround us can be studied in this context of massively multi-domain problems when they are explored at the mesoscopic scale. Existing defects within matter can be formulated in terms of energy. The microstructure, through its evolutions, aims generally to minimize this stored energy. This field is obviously academically and industrially exciting as microstructures are of prime importance concerning the final in-use material properties (mechanical strength, fatigue limit, crack resistance, stress corrosion resistance,...). Precise numerical modeling of materials is then a topic of prime importance largely due, firstly, to the demonstrated value of predictive simulations of materials behavior, in greatly reducing the time and cost of developing new materials and, secondly, to the theoretical interest of such strategy to improve our understanding of materials science phenomena. In the context of monophase solid metallic materials, the simulations can take into account a wide range of variables (grain orientations, the curvature of interfaces, dislocation density, etc.) that translate into displacements and interactions between Grain Boundaries (GB). The efficient treatment of the moving interfaces between grains is then critical, considering the number of grains needed (hence the number of interfaces) to form a valid Representative Volume Element (RVE) of the material (usually composed of thousands of grains).

This special field of research is the main focus of the software DIGIMU® developed at CEMEF and Transvalor company in collaboration with several partners: Safran, Aubert & Duval, Framatome, CEA, Timet, ArcelorMittal, Constellium and Ascometal. DIGIMU® currently includes 2D and 3D tools for the generation of large polycrystals and the modeling of GB migration (GBM) due to grain growth (GG), recrystallization (ReX) in dynamic (DRX), post-dynamic (PDRX), metadynamic (MDRX) or static (SRX) conditions, and Smith-Zener pinning mechanism (SZP) under the influence of second phase particles (SPPs). Additionally, recent developments have been focused on the proposition of new models able to take into account anisotropic grain boundary properties, the influence of dynamic SPPs and coupled crystal plasticity models.

DIGIMU® has been developed employing Finite Element (FE) technology using unstructured FE meshes, coupled with the Level-Set (LS) approach [5]. This numeric formulation has the advantage of being able to take into account very large deformations of the domain with relative ease. This is very valuable for the metal forming industry as usual thermomechanical treatments (TMT) are within this regime. Numerous improvements have been made to the DIGIMU's core algorithm in the last few years regarding its performance and robustness [6, 7]. Of course, the FE-LS model is not the only approach capable to perform simulations for microstructural evolutions and a great interest remain for alternative (existing or new) models, usable, in a similar manner as in DIGIMU®, in a large deformation context.

Currently, the DIGIMU approach has been optimized to the point where increasing its performance is a very difficult task. However, recent studies [8] have mentioned that a great amount of computational time (more than the 70% of the total time) is spent by the remeshing algorithm, used to *capture* the grain boundaries with an anisotropic refined mesh. This is why in this work, an entire chapter (Chapter 2) will be dedicated to the study of the performance of the FE-LS model, using alternative remeshing techniques that could potentially replace the current strategy employed by DIGIMU.

Moreover, a new method able to perform massive-multidomain simulations in a large deformation context will be introduced. The new method is inspired by Front-Tracking [9, 10, 11, 12, 13] and Vertex [14, 15, 16, 17, 18] approaches in the sense that geometrical properties are only computed at the interfaces and the migration of the GBs is defined thanks to a Lagrangian scheme. Of course, the main ambition of the new approach is the improvement of the computational performance when simulating evolving microstructures while maintaining an equivalent accuracy.

The majority of the works presented in this manuscript have been published [19, 20] or are under review for publication [21, 22]. The chapters of this manuscript

follow approximately the order of publication. As such, the reader may find some repetitions involving the definition of some notions, especially at the beginning of each chapter. We have considered that these repetitions do not harm the context at hand, but are instead helpful to guide the reader to the principal aspects of each chapter.

In the first chapter of this manuscript, the notions involving the dynamics behind the evolutions of microstructures (mainly for GG and ReX) will be given followed by a description of the existing *full-field* models used in this context. Additionally, a brief insight into the treatment of Eulerian methodologies will be discussed as well as some examples of applications using the concept of LS as an input to obtain body-fitted meshes.

In the second chapter, a thorough analysis of the performance of the current model employed by DIGIMU® will be performed. A section focused on the last advances proposed for the LS-FE method will be given. Additionally, different numerical configurations will be tested, corresponding to different remeshing strategies often performed in the context of FE models to capture regions of interest in a domain, which in the context of microstructural evolutions are the GBs. The limits of the FE-LS in terms of performance will be explored and the best remeshing configuration will be identified.

The third chapter will introduce the new 2D methodology denominated TOpological REmeshing in lAgrangian framework for Large interface MOTION (ToRealMotion, hereafter TRM) as an alternative for the modeling of massive-multidomain problems applied to GG. The presented method maintains the discretization of the interior of the domains, using an evolving unstructured triangular mesh (valid for a FE study) and treats the topological events such as the disappearance of domains or the creation of interfaces using selective local remeshing operations.

Then, in the fourth chapter, the parallel implementation of the TRM model will be presented: the algorithms developed to address the parallel framework will be explained and several test cases will be performed to characterize the speed-up of the TRM model in parallel. Here, tests involving up to 560000 grains will be performed, being the first approach (to the knowledge of the author) to attempt simulations of this magnitude in the context of unstructured finite element meshes.

The fifth chapter will explain the developments of the TRM model oriented to the modeling of ReX. Here, the experience acquired through the development of DIGIMU® will be used and translated to the TRM approach, in addition to some necessary implementations: a nucleation method, responsible for the appearance of new grains, and a phenomenological approach to use the dislocation density in

the form of Stored Energy (SE) as an input for the GBM.

The sixth chapter will explore the use of the TRM model in the context of anisotropic GB properties. Special attention will be given in this context to multiple junctions, and the formulation of a GBM model taking into account such anisotropic quantities.

Finally, a short last part will set the perspectives for futures works and potential applications of the TRM model.

Oral and written communications

These works have lead or contributed to the following written and oral communications.

Articles

- Sebastian Florez, Modesar Shakoor, Thomas Toulorge and Marc Bernacki, A new finite element strategy to simulate microstructural evolutions, Computational Materials Science 172 (2020) 109335.

  doi:10.1016/J.COMMATSCI.2019.109335.

- Sebastian Florez, Karen Alvarado, Daniel Pino Muñoz and Marc Bernacki, A novel highly efficient Lagrangian model for massively multidomain simulations applied to microstructural evolutions, Computer Methods in Applied Mechanics and Engineering 367 (2020) 113107.

  doi:10.1016/j.cma.2020.113107.

- Julien Fausty, Brayan Murgas, Sebastian Florez, Nathalie Bozzolo, Marc Bernacki. A new level set finite element formulation for anisotropic grain boundary migration. Applied Mathematical Modelling, Under review.

  arXiv:2006.15531.

- Sebastian Florez, Julien Fausty, Karen Alvarado, Brayan Murgas and Marc Bernacki, A novel highly efficient Lagrangian model for massively multidomain simulations: parallel context, Modelling and Simulation in Materials Science and Engineering, Under review.

  arXiv:2009.04424.

- Sebastian Florez, Karen Alvarado and Marc Bernacki, A new front-tracking Lagrangian model for the modeling of dynamic and post-dynamic recrystallization, Modelling and Simulation in Materials Science and Engineering, Under review.

  arXiv:2009.08368.

- Karen Alvarado, Sebastian Florez, Baptiste Flipon, Nathalie Bozzolo and Marc Bernacki, A level set approach to simulate grain growth with an evolving population of second phase particles, Modelling and Simulation in Materials Science and Engineering, Under review.

- Sebastian Florez, Brayan Murgas, Karen Alvarado, Julien Fausty and Marc Bernacki, Extension of a front-tracking Lagrangian model for the modeling of heterogeneous and anisotropic grain growth, In preparation.

Proceedings

- Sebastian Florez, Modesar Shakoor, Thomas Toulorge, Marc Bernacki, Body-fitted finite element discretization for moving interfaces in context of microstructure evolutions. CSMA 2019, 14ème Colloque National en Calcul des Structures, May 2019, Giens, France

Book Chapters

- Marc Bernacki, Nathalie Bozzolo, Pascal de Micheli, Baptiste Flipon, Julien Fausty, Ludovic Maire and Sebastian Florez. Numerical Modeling of Recrystallization in a Level Set Finite Element Framework for Application to Industrial Processes. Recrystallization: Types, Techniques and Applications, Nova, 2019, 978-1-53616-737-5.

Posters

- Sebastian Florez, Marc Bernacki. A new full field framework to model grain growth in FE context. ReX & GG 2019 - 7th international conference on recrystallization and grain growth.

Presentations

- Sebastian Florez, Thomas Toulorge, Marc Bernacki. Body-Fitted Remeshing of Moving Interfaces Applied to Microstructural Evolutions, Materiaux 2018, Strasbourg, France.

- Sebastian Florez, Thomas Toulorge, Marc Bernacki. Impact of body-fitted finite element discretizations for moving interfaces applied to microstructural evolutions. ADMOS 2019 - International Conference on Adaptive Modeling and Simulation, May 2019, Alicante, Spain.

- Sebastian Florez, Modesar Shakoor, Thomas Toulorge, Marc Bernacki. Body-fitted finite element discretization for moving interfaces in context

of microstructure evolutions. CSMA 2019, 14ème Colloque National en Calcul des Structures, May 2019, Giens, France

- Marc Bernacki, Nathalie Bozzolo, Charbel Moussa, Daniel Pino Muñoz, Pascal de Micheli, Sebastian Florez et al.. Towards the full field modeling of microstructure evolutions during metal forming industrial processes. 7th International Conference on Recrystallization and Grain Growth - ReX & GG 2019, Aug 2019, Ghent, Belgium.

- Sebastian Florez, Thomas Toulorge, Marc Bernacki Body-Fitted Finite Element Discretizations for Moving Interfaces in Context of Microstructure Evolutions, Journées Matériaux Numériques 2019, Amboise, France.

Contents

Chapter 1

Literature Review

Most mathematical/numerical models are based and/or developed to reproduce physical phenomena. However, the formulation of such a model is not, in general, the unique nor the truthful approach for a given physical mechanism. e.g. microstructural evolutions such as GG and ReX have been attempted to be described by a series of phenomenological and numerical models, all having their strengths and weaknesses. In this chapter, some of these models will be described, particularly the so-called *full-field* (FF) models, which have been successfully applied in this context, to reproduce morphological and statistical quantities, critical for the metal-forming industry. Of course, the intention is not to offer an extensive definition of all models, but the notions involving their mathematical foundations.

Then, a brief discussion will be held to contextualize current strengths and weaknesses of the presented models, both from a numerical (pertinence of the results and ease of implementation) and computational perspectives. In this context, we will list the FF models developed by the community at the time of writing, but of course, this list is in constant evolution given the increasing number of methods being developed.

1.1 An insight over microstructural evolutions of monophase metallic materials

Modern notions used to describe and to study metal alloys are centered on the fact that, at the mesoscale, their structural composition is made of *grains*, each representing a *crystal*[1] with its own crystallographic orientation and chemical composition [24]. At this scale, the *microstructure* of the material appears as an aggregate of such grains delimited by the denominated grain boundaries. Such an arrangement is commonly known as a *polycrystal*.

The physical (mechanical, thermal, electrical, magnetic...) properties of a given metal alloy can be drastically changed by the application of a thermomechanical treatment. In fact, a TMT can alter the structure of the metal at the microscopic scale, changing, firstly, the polycrystal structure formed by grains and GBs and secondly, in some cases, the local chemical composition of the alloy through the segregation/absorption of given elements (which can form or dissolve phases present in the material). For simplicity, in this work, we will focus on microstructural evolutions that do not produce a change of phase of the material, hence in monophase microstructures.

1.1.1 Microstructural evolutions during TMT

When hot metal forming is considered, in the context of a monophase polycrystal, three principal mechanisms are involved in the dynamics of the microstructural state of the material: recovery, ReX and GG [25].

Recovery mechanisms are associated with the minimization of the energy accumulated in the bulk of the material in the form of crystallographic defects (dislocations), during deformation. Recovery is then defined as the mechanism, taking place under the influence of temperature, where dislocations interact with each other and annihilate or rearrange in energetically lower configurations. Recovery, however, is not normally related to the evolution of grain boundaries as it is a relatively homogeneous process taking place inside the crystallographic structure, hence being able to create *substructures* (or subgrains) inside a grain but not GBM.

Recovery, however, is not able to completely eliminate defects inside the crystal and in most cases the energy dissipated through this mechanism is not very representative. Generally, very high concentrations of dislocations within a material lead to a much more dissipative mechanism called recrystallization. This mechanism is indeed divided in two parts: Firstly, recovered or deformed zones

[1] A crystal structure is a state of matter characterized by the regular arrangement of a unit motif over relatively large atomic distances [23].

rearrange into new *dislocation-free* crystallites (new grains) presenting its own crystallographic orientation, hence introducing new GBs. Secondly, The new grains growth at the expense of the high *stored energy* present in their surrounding matrix (surrounding grains).

Even though ReX can lead to grain structures with a very low concentration of dislocations, the resulting microstructure may still contain energy in the form of GBs, which leads to GBM by curvature flow, i.e. the GG mechanism. During GG, the energy minimization is then driven by the reduction of the total amount and energy density of grain boundaries, *Curved grain boundary migrates toward its center of curvature [26]*, inducing to a higher mean grain size by the disappearance of small grains and the grow of large grains.

Even though these mechanisms may appear simple, their combination during simple modern TMTs can result in very complex microstructures. Small changes in the parameters of the treatment can lead to completely different final microstructural states. While characterizing a metallic material at the grain scale is a very mature field of research [27, 24, 28, 29, 30, 31, 32], the prediction of such states in function of the known variables of the TMT applied to a given material is not straightforward. Multiple models have been developed to give a more understanding point of view regarding these aspects, and in the last few years, the metal forming industry has become more and more interested in such numerical solutions. The next section is dedicated to this field of research, listing some of the most relevant numerical models, specifically those denominated *full-field* models, intended to simulate microstructural evolutions using a full description of the polycrystal at the mesoscopic scale.

1.2 Modeling of ReX and related phenomena

One of the first models involving the migration of a GB, based on the observations given in [33, 26, 34], was proposed in [35, 36]. It was assumed that a driving force acting on the boundary was only produced by the surface tension of the boundaries, hence being directly proportional to its curvature. These conclusions were also based on the similarities encountered between the shapes of cells in foams and of metal grains, which were pointed out by the authors in [34, 37].

Consequently, in [36] it was proposed, in an isothermal context, the equation:

$$G = M\gamma\kappa, \tag{1.1}$$

where G is the GBM rate, γ is the surface energy of the GB, κ is the local magnitude of the curvature in 2D (trace of the curvature tensor in 3D) of the interface, and M is the *mobility*. Large discussions concerning the meaning of

M are present in the literature. In the context of hot metal forming and when modeling at the mesoscopic scale is considered, M is generally defined through an Arrhenius law:

$$M = M_0 * e^{-Q_t/RT}, \tag{1.2}$$

where R is the ideal gas constant, Q_t is a term defining the thermal activation energy of the material and M_0 is a constant material property.

Note that in [26], the concept of direction of boundary migration was already studied: *curved grain boundary migrates toward its center of curvature.* This direction of migration is the same as the one given by the normal vector $-\vec{n}$ to the grain boundary (pointing in its convex direction), making the vector of the velocity of boundary migration $\vec{v_c} = -G\ \vec{n}$ and thus using Eq. 1.1:

$$\vec{v_c} = -M\gamma\kappa\vec{n}, \tag{1.3}$$

where Equation 1.3 is generally accepted in the literature as the velocity of migration of grain boundaries under the influence of surface tension.

Even though the understanding of the local behavior of grains boundaries in a microstructure was already quite advanced at that time, the limitations in computational power did not allow the study of a microstructure in a full field context, and in [36] these assumptions where instead used in the development of a phenomenological law aiming to predict the value of the mean grain size (the well-known Burke & Turnbull model).

In [36] it is also mentioned that another proposition for a driving force acting during grain growth was given by Jan Czochraiski[2] in [39][3]: *grains formed by recrystallization have residual strain energy and upon further heating the more perfect ones will grow at the expense of the less perfect ones.* This postulate was perhaps the first explanation of another driving pressure produced by the SE of grains due to plastic deformation [25]:

$$[E] = \tau[\rho], \tag{1.4}$$

where $[E]$ and $[\rho]$ denote respectively the difference in the SE and the dislocation density between two neighboring grains and τ is the dislocation line energy.

Of course, a relation can be established for the velocity of GBM due to SE effects [40]:

[2]More information about the life and the huge contribution to the microelectronics industry of Jan Czochraiski can be found in [38].

[3]Although reference [39] is well cited in [36] there is no record of this publication in the major scientific databases.

$$\vec{v_e} = M[E]\vec{n}, \tag{1.5}$$

however, this mechanism as a driving force was not taken into account in [36] and it had not a proper application until the apparition of full field models. The concept of stored energy will be rediscussed in chapter 5.

Later, the relations given in Eq. 1.3 and in Eq. 1.5 were contextualized into a single relation: $\vec{v} = MP\vec{n}$ where the term P (the total driving pressure) summarizes the effects given by the surface tension and the stored energy, (as well as any other sources producing a driving force not studied in this work) [25, 40].

1.3 Classical Full Field Models for Microstructural Evolutions

Predicting mean quantities such as the mean grain size of a microstructure can be carried out using physical-based phenomenological laws which typically are based on a large list of simplifications, such as the analytical form of grains, typically taking shapes of circles or spheres. These *mean field* models [41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51] have been progressively enhanced over time and correspond to a good compromise in terms of accuracy of the predictions and computational performance [51]. Mean field models are a pertinent choice, if for the considered application, it is only necessary to know the qualitative evolution of a few parameters (such as mean grain size, grain size distribution or recrystallized fraction). However, mean field models are unable to accurately predict polycrystal topology and the morphology of grains. Moreover, heterogeneous properties (anisotropy of γ and M, population of SPPs...) can lead to much more complex evolutions, which are not accessible to phenomenological and mean field approaches. In such a context, it is necessary to perform simulations in a *full field* framework, where the simulated domain reflects a one-to-one map to the real microstructure[4] being simulated. In other words, full field simulations aim to obtain a higher level of detail by simulating the evolution of real microstructures where grains are *mutually interacting grains [52]* and not of a hypothetical statistical representation of a microstructure as in mean field models.

In this section, a list of full field models will be given, explaining the concept behind each one of them, and some of the results available in the literature. Of course, I do not pretend here to provide an exhaustive study of all these methods, but to give some of the relevant information about each, which will allow to

[4]Here a real microstructure refers to a topologically connected microstructure [52] obtained by a Scanning Electron Microscopy (SEM) or 3D X-ray imaging techniques [53, 54] or to a virtual microstructure generated using the concept of Voronoi or Laguerre Voronoi cells [55, 56, 57].

better place the works presented in this PhD thesis.

There are typically two main families of methods to describe moving interfaces: Front-Capturing (FC) and Front-Tracking (FT) Methods [58]. FC models implicitly (indirectly) define interfaces by *capturing* the change of state of a field variable. Two examples of FC approaches are given below: Multi Phase Field (MPF) and LS models. FT models explicitly (directly) define interfaces using a set of interconnected segments (in 2D) or surfaces (in 3D). By this definition, Vertex models (also described below) are FT models, as vertices are connected between them to form the segments that define the boundaries between domains.

1.3.1 Monte Carlo models

The first full field model for the simulation of microstructural evolutions was presented in [52] using a MC model:

The simulation domain is represented by a regular lattice where the microstructure is mapped. The lattice is conformed of sites with polyhedral shapes (triangular, squared or honeycomb shapes) typically taking the form of pixels in 2D or voxels in 3D. Each site i is given an orientation S_i in the form of an integer between 1 and Q (where Q defines the maximum number of possible orientations in the microstructure), and grain boundaries are defined to be the edges between two neighbors sites with different orientation. Grain boundary energy is derived from the Hamiltonian:

$$H = \sum -J(\delta_{S_i S_j} - 1), \tag{1.6}$$

where the summation term is carried out for all nearest neighbors sites j of site i and δ_{ah} is the Kronecker delta symbol; and J is a boundary energy term given by the unlikeness of two consecutive sites. If two neighbors sites have a different orientation, they contribute the amount of J to the energy of the system and zero otherwise.

Boundary migration is then obtained by the integration of an MC approach over the simulation domain: a site i and a new trial orientation S_i^* (different from S_i) are selected both at random. A transition occurs (S_i is attributed the value of S_i^*) depending on the transition probability factor W which classically takes the form:

$$W = \begin{cases} e^{(-\Delta G/K_b T)} & \Delta G > 0 \\ 1 & \Delta G \leq 0 \end{cases} \tag{1.7}$$

where ΔG is the change in energy caused by the orientation S_i^* applied to site i, K_b is the Boltzmann constant and T is the absolute temperature. If the local change in the energy is less than or equal to zero the transition is applied, while

for changes in the local energy greater than zero, the probability of reorientation is given by $e^{(-\Delta G/K_b T)}$.

The unit of time for these simulations is represented by the *Monte Carlo Step*, which refers to N reorientation attempts, where N is the total number of sites in the lattice. Each site may be chosen several times which also means that another may not be tested.

Results obtained with this model forty years ago set a great achievement on the understanding of global microstructural evolutions. Indeed, MC approaches have brought a new perspective to the discussion of GBM, compared to existing mean field approaches. Better agreement with existing 2D experimental data was obtained, for example, experimental observations of GG given in [59] showed a global behavior of the mean grain size $\overline{R}$ as follows:

$$\overline{R} = kt^n, \tag{1.8}$$

where k is a constant (typically described by an Arrhenius equation), t is the time and the exponential factor n is fitted from the experimental data with a mean value of 0.4.

The author in [52] emphasizes that while phenomenological models (like Burke & Turnbull models [36, 60]) predict a value of n around 0.5, the MC approach was able to predict values of n of around 0.41 ± 0.03 for high values of Q, hence being in "better" agreement with experimental data. Moreover, the author also remarks that literature had attributed the differences between the exponential factor of phenomenological models and experimental data to the presence of impurities, preferred grain orientation, or second phase particles. This reasoning was disproved and the differences were attributed to the fact that *phenomenological models are unable to reproduce behaviors of systems with a strong dependency over the vertices in a topologically connected domain [52].* This discussion also illustrates the undeniable interest of FF approaches.

The MC approach was presented in numerous articles, all aiming to respond to different mechanisms involved in microstructural evolutions. In [61] the influence of SPPs was studied, in [62] abnormal grain growth (AGG) due to SPPs was discussed and in [63] the influence of anisotropic grain boundaries (misorientation dependence) based on a Read-Shockley formulation was introduced [64]. The latter was indeed the first model attempting to simulate microstructural evolutions in such a context.

Further studies of the MC method added the influence of SE, this was done by the modification of Eq. 1.6, in order to take into account a bulk energetic term [65]:

$$H = \sum -J(\delta_{S_i S_j} - 1) + E_j, \tag{1.9}$$

where E_j denotes the SE of site j.

This modification allowed the authors in [65] to perform for the first time ReX simulations using a full field approach: Homogeneous[5] [65] and Heterogeneous [66] SRX under site saturated[6] and constant nucleation rate[7] were performed, showing good agreement with theoretical predictions and experimental data.

These studies were later extended to the simulation of DRX [67] where the dependence of the SE was correlated as proportional to the dislocation density ρ which in turn was related to the flow stress σ as proportional to the square root of ρ. Moreover, the rate of change of SE values ΔE_i was set constant for all sites i during a simulation, hence heterogeneities on the SE were able to be simulated as the sites belonging to new nuclei were attributed a value of $E_i = 0$, this is, in turn, reflected as to obtain a homogeneous and constant SE value per grain. No deformation of the domain (hence no plastic strain) was taken into account under the assumption that grain shapes tend to remain equiaxed [68]. Additional studies in this matter were developed later in [69, 70, 71] where the influence of dynamic recovery, nucleation rate mechanisms and temperature were respectively discussed.

Additional studies [72] have been carried out in order to enhance the original algorithm, by reducing the range of possibilities for the reorientation mechanism, taking into account the orientation of the neighbors sites only. This modification allows a grain mobility independent of the number of considered orientations (Q) and avoid coalescence between neighbor grains with the same orientation. However, it was shown that coalescence events were not a critical mechanism when studying a large number of grains as this mechanism appears rarely on the original algorithm. These studies also showed a different behavior regarding the exponential factor n and a better fitting of the grain size orientation to a log-normal curve.

More recently, in [73], a *hybrid* MC/CA model was proposed, intended to gather the strengths of the two models: curvature driven GBM, "accurately" reproduced from the point of view of MC, and the modeling of primary ReX "accurately" reproduced from the point of view of CA models. See the next section for a description of CA models.

[5]Historically, the terminology homogeneous ReX is used when the nucleation takes place regardless of the nuclei position, contrary to Heterogeneous ReX when the nucleation is oriented to appear at specific locations such as grain boundaries and triple junctions.

[6]Site saturated nucleation corresponds to a SRX scenario where the totality of nuclei involved in the simulation takes place instantaneously, typically at the beginning of the simulation.

[7]Constant nucleation rate refers to the fact that the number of nuclei per unit of volume of unrecrystallized material per unit time is constant.

1.3.2 Cellular Automaton models

The concept of cellular automaton was first introduced in [74], here the authors focused on the study of self-reproducible biological processes. Later, in a series of published works [75] Stephen Wolfram starts the study of one-dimensional cellular automata based in a systematic set of rules, these studies let to the publication of a controversial work aiming towards a fundamental theory of physics [76] under the speculation that the universe is essentially discrete and behave as a cellular automaton.

CA models in general have to define at least the following properties [77]:

- The geometry of cells (hence the spatial dimension is also defined).
- The states a cell is able to take.
- A formal definition of the neighborhood of a cell.
- The transitioning rules for a cell, defining the state of each cell in the next time step taking into account its old state and the state of the cells in its neighborhood.

Applications on microstructural evolutions using the concept of CA were first introduced by Hasselbarth [77] for simulations of ReX: the presented model consisted of a 2D lattice composed of squared sites, these sites had two states, they were either recrystallized or not. At the beginning of the simulation, all sites were set to unrecrystallized, then different rules for nucleation were considered: site saturated nucleation and constant nucleation rate, where the nucleation sites were chosen at random. The GBM due to the appearance of new grains was modeled following two different approaches: the first where cells with at least one recrystallized neighbor are changed to the recrystallized state and the second where additionally to the presence of a ReX cell in its neighborhood, the transition depends on a probability function. Finally, all cells of the lattice are evaluated but transitions only occur at the end of a time step, hence all states in the new increment are determined by the states in the old increment.

The influence of a heterogeneous nucleation rate and growth rate were studied in [77]. The results of this work showed a good agreement with the Johnson and Mehl [78], Avrami [79, 80, 81] and Kolmogorov [82] (JMAK) model regardless of the neighborhood definition of cells. Although this article set the first attempt to simulate microstructural evolutions, involving a rather complex mechanism such as ReX, this work did not implement a mechanism based on neither GB energy nor SE. Instead, the model was defined through the assumption of a hypothetical heterogeneity on the SE field, translated into high or low nucleation rates and in high or low GBM rates (for example allowing cells transitions only every n^{th} increments in certain regions).

The works of Hesselbarth motivated further studies where the influence of second phase particle impingement was studied [83] under the assumption that recrystallization could not happen in cells representing a particle, leading to reasonable quantitative results.

Later, the idea to use CA on microstructural evolutions lead to the development of a GG model using a similar concept [84]. Here, some of the components of the MC method were used and some others were developed.

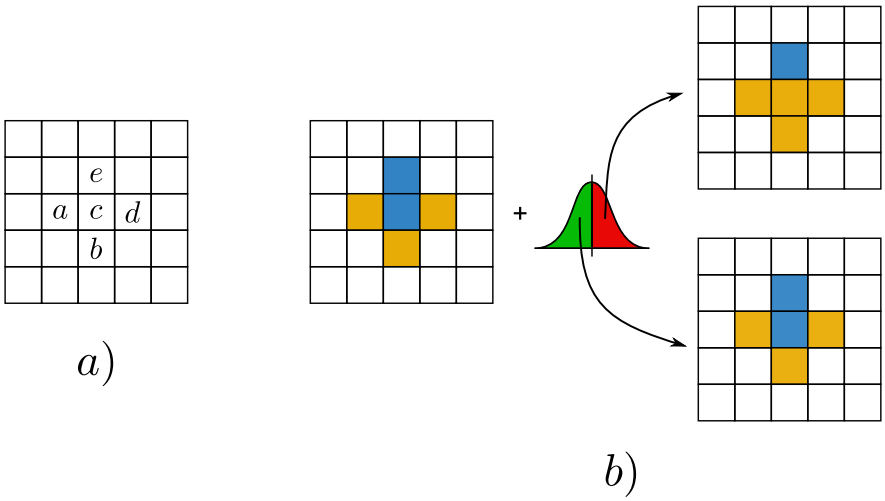


Figure 1.1: a) Von Neumann's definition of neighbor cells (a, b, c, d, e) of a cell c, the center cell c is also regarded as a neighboring cell. b) Energy barrier between two states (*blue* and *orange*). The center cell evolves to state *orange* when its energy is sufficiently high. Redrawn from source: [84].

Figure 1.1a gives the Von Neumann's definition of neighbor cells which were used in [84] (contrary to [77] where multiple definitions of neighborhood were used), here a central cell c is surrounded by four neighboring cells (a, b, d, e). The transition of cell c is then determined by the following rules:

1. The states are defined by the integers from 1 to Q (where Q similarly to MC represents the maximum number of possible grain orientations), instead of *recrystallized* and *unrecrystallized* states as in [77], moreover, Q takes the number of 1000.

2. If at least three surrounding cells (a, b, d, e) have the same state as the central cell c, the latter maintains its state.

3. Cell c must overcome the energy barrier to modify its state. This is done as in MC methods by the use of a probabilistic law with a similar form of Eq. 1.7: $W = e^{(-E_b/K_bT)}$ where E_b is an energy barrier that the cell must overcome to transition, see Fig. 1.1b.

This approach is nearly identical to the one introduced in [72] for MC methods (Eq. 1.6), except for the strategy adopted for transitioning cells: this is given by rule 2 and the fact that transitions only occur at the end of a time step and all cells are evaluated, instead of the use of a random selection of cells and the notion of the MC steps.

Additionally, the energetic term E_b was held constant for all cells while the temperature T was considered high enough to approve all transitions ($W = 1$).

Stochastic CA models for the simulation of ReX and GG in 2D and 3D were developed in [85, 86], where the authors focused on the implementation of CA models using a phenomenological equation [60][8] as a transitioning rule, allowing to treat the evolution of microstructures on a real space and time scale by the use of realistic boundary mobility and energy data. The stochastic behavior of the model relies on the fact that the transitioning of the orientation of a given cell is performed with a certain probability by a Monte Carlo step, once this transition probability was determined by the phenomenological law. Additionally, to perform simulations of ReX in 3D, this model was able to use data issued from simulations of crystal plasticity in order to set the initial state of the simulated domain.

Even though the CA model was already able to model PDRX (i.e. the use of SE as a driving force) satisfactory, the ability to correctly simulate boundary migration due to curvature-flow was still an open subject. This motivated the works presented in [73] where a *hybrid* model was presented, using the inherent properties of the MC method to *approach* curvature-flow behavior. The resulting model was applied to simulate a Gibbs-Thomson like metastable configuration, giving statistically satisfactory results in a given range of data.

Since [85, 86, 73], stochastic CA models employ probabilistic equations of the form $P = v/v_{max}$ where P is the probability of transition, v is the velocity in the considered cell and v_{max} is the maximum possible velocity in the lattice (computed using instant microstructural states and field properties). the choice concerning the transition of the cell is based on the probability P and, of course, all transitions occurring, as established by the CA model, at the end of a time step. In this context, the development of an accurate description of GG remains, however, an open question. This is due to the high difficulty to evaluate geometrical properties (such as curvature) in configurations using this type of domain discretization. For example, the works presented in [87] in the context of solidification applications, particularly in the use of a template-based curvature computation, motivated recent developments on the CA model [88, 89] in the

[8]The actual mathematical form of the phenomenological law used in [85, 86] was not the original form of the linearized symmetric rate equation for thermally activated GBM under the influence of free energy gradients presented in [60], but an analog probabilistic form.

context of SRX and DRX.

Further studies involving the CA method can be found in [90] where the authors make a thorough comparison of the MC and the CA models in a SRX context. In [91] the authors present a detailed description of the implementation of a parallel algorithm to perform cellular automata computations in the context of microstructural evolutions. A benchmark of the parallel implementation was performed and the results can be found in Fig. 1.2, where the *speed up*[9] is plotted against the number of computational nodes (CPUs) used, and for various sizes of the CA lattice.

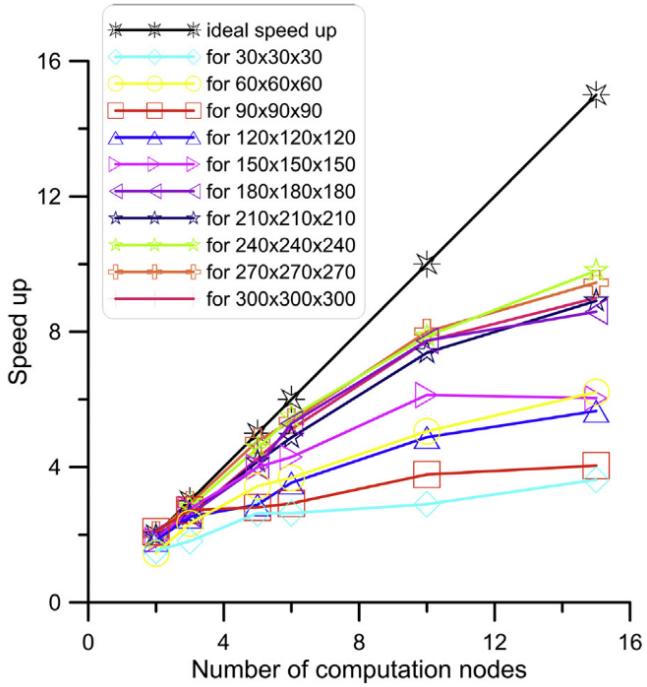


Figure 1.2: Performance results of the CA framework of Rauch et al. Source: [91].

Finally, a new numerical model based on CA was proposed in [92, 93] for the simulation of DRX. The model couples a FE resolution to deal with domain deformation while Random Cellular Automata (RCA) models the nucleation and GBM. RCA models rely on the assumption that the cellular automata cells directly correspond with finite element integration points, allowing remeshing if needed from the analogous FE mesh while the notion of neighboring is taken based on the distance of cells. This model is still in development and remains a promising alternative to model complex microstructural evolutions.

[9]The speed up in [91] was obtained as a result of the quotient of a sequential algorithm execution time and a parallel execution time using a specific number of cores

1.3.3 Vertex Models

One of the first Vertex models applied to microstructural kinetics was the one presented by Soares et al. [94] based on the works of [95] applied to the evolution of soap froth. This model considers that the evolution of an interface network can be simulated from the movement of its triple junctions (multiple junctions with three connections), a proposition first introduced in [96], which enables to avoid the computation of curvature by the use of straight lines as grain boundaries. The motion equation of the triple junctions used by Soares et al. was derived from a thermodynamics study (unlike the one proposed in [96]) and can be summarized as:

$$\vec{v} = M \sum \vec{\varepsilon}_i, \tag{1.10}$$

where $\vec{v}$ and M are respectively the velocity and mobility of a vertex and $\vec{\varepsilon_i}$ denotes each of the line tensions acting at a vertex. In this model, vertices at all times only have 3 possible connections, avoiding higher order multiple junctions configurations by means of unitary topological changes (see transformations **T1** and **T2** of Fig. 1.3). Moreover, M and $|\varepsilon_i| = \varepsilon$ were assumed constant, hence the model was tested only under isotropic conditions. In a similar manner as in MC studies, the model showed agreement in the context of GG with phenomenological laws with an exponent factor of $n = 0.52$ in Eq. 1.8.

A more general model was proposed by Kawasaki et al. in [14]. Here the authors were able to propose an implicit equation for the computing of the velocity of each vertex, which is based on a dissipative equation of motion of the form:

$$\frac{\partial R}{\partial \vec{v}} = -\frac{\partial \nu}{\partial \vec{r}}, \tag{1.11}$$

where R represents the Rayleigh dissipation function and ν the potential function. This equation can be described as: the amount of free-energy dissipated following the movement of the vertices with a velocity $\vec{v}$ is equal to the negative change on the potential energy following a change in the position of vertices $\vec{r}$. Kawasaki et al. then derive its implicit formulation as:

$$\frac{1}{3} \sum \frac{|\vec{r}_{ij}|}{M_{ij}} \vec{n}_{ij} \otimes \vec{n}_{ij} \cdot (\vec{v}_i + \frac{1}{2}\vec{v}_j) = -\sum \gamma_{ij} \frac{\vec{r}_{ij}}{|\vec{r}_{ij}|} = -\sum \gamma_{ij} \vec{t}_{ij}, \tag{1.12}$$

where $\vec{r}_{ij}$ is the vector $\vec{r}_i - \vec{r}_j$, $\vec{r}_i$ is the position of vertex i, $\vec{n}_{ij}$, γ_{ij} and M_{ij} are the normal unit vector, the boundary energy and the mobility of the grain boundary segment delimited by vertices i and j, $\vec{v}_i$ is the unknown velocity of vertex i, $\vec{t}_{ij}$ is the unit vector in the direction of $\vec{r}_{ij}$ and the summation is carried out for each vertex j connected to a given vertex i.

Equation 1.12 can be approximated by averaging over all the directions of $\vec{n}_{ij}$ and $\vec{v}_j$ which yields:

$$\frac{1}{6M}\left(\sum|\vec{r}_{ij}|\right)\vec{v}_i = -\gamma\sum\vec{t}_{ij}, \tag{1.13}$$

this equation being introduced as **model II** by Kawasaki et al. allows to explicitly compute the velocity at each vertex i in function of the position of its connected vertices. More information about the differences between Eq. 1.12 and Eq. 1.13 can be found in Appendix A of [14].

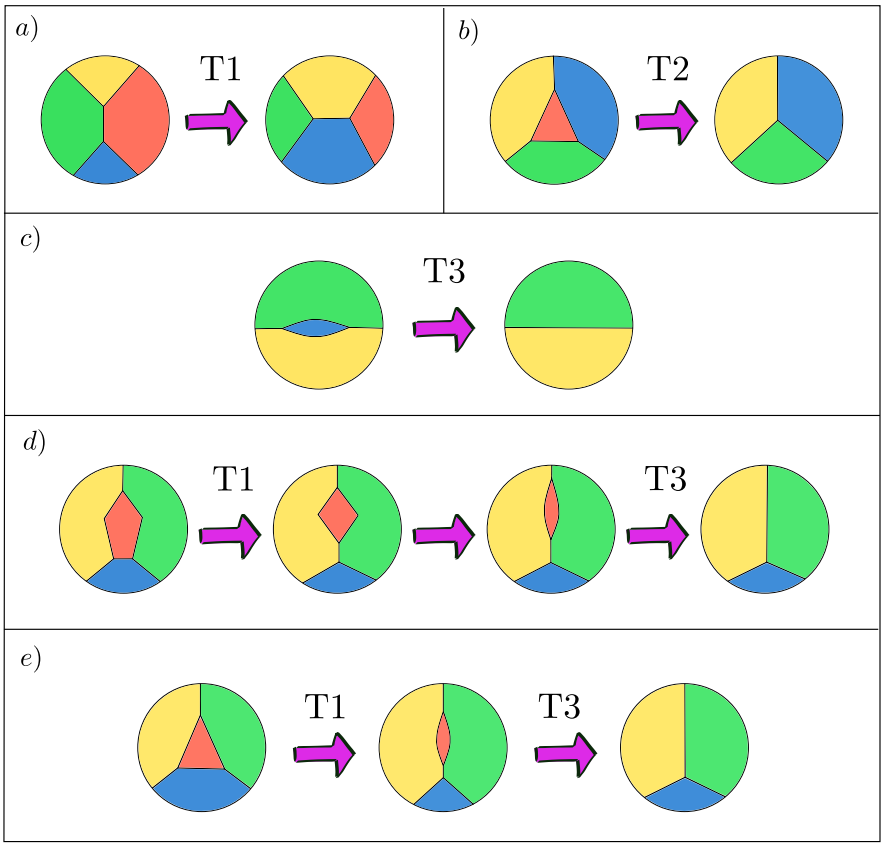


Figure 1.3: 2D Topological transformations: a) recombination (T1), b) annihilation of small three-sided grain (T2), c) elimination of a grain with two triple points (T3), d) an example of the occurrence of a T3 transformation and e) T2 transformation shown as equivalent to a sequence of T1 and T3 transformations. Source: [15]

Ten years later, the works of Weygand et al. [15] extended the concept of Kawasaki et al., by implementing a vertex model capable of modeling curved grain boundaries. The model used *real* and *virtual* vertices, where the virtual vertices were used to discretize the grain boundaries between triple junctions (real vertices). Weygand et al. also used directly Eq. 1.12 in order to compute the velocity values of all vertex (both real and virtual) instead of the explicit approximated Eq. 1.13 used by Kawasaki et al. in their study. Furthermore,

they also introduce a topological operation inherent to the use of virtual vertices, corresponding to the topological operation **T3** given in Fig. 1.3.

Additionally, in [15] the notion of micro-stepping was first applied in this context:

Micro-stepping is necessary given the unstable nature of some vertices that disrupt the smoothness of grain boundaries (vertices that are too off positioned regarding the tendency of their contiguous vertices) and which may appear after a topological change or another numerical phenomenon. Micro-stepping was then implemented by using the following reasoning: no segment should shrink nor extend more than a fraction $f = dl/l$ of its length l (where the reported value for f was 0.5), leading to a maximum time step allowed per vertex.

Finally, the movement of vertex i was done using the following equation:

$$\vec{r_i}(t + dt) = \vec{r_i}(t) + \vec{v_i}(t)dt, \tag{1.14}$$

where t is the time and dt is the time step. This equation was used both for hole steps and micro steps. Using this model, Weygand et al. found that multiple junctions evolve so to fulfill their equilibrium angles [25].

The works of Weygand et al. then lead to a series of publications where ReX [97] and the effects of SPPs [98] were studied, as well as the influence of the reduced mobility at multiple junctions and grain boundaries [99, 100] over GG, and finally, extending its work to three dimensions [101, 100, 16].

A second kind of Vertex models were introduced by Frost et al. [102], where the main difference is that real vertices were positioned at their angles of equilibrium while the movement of virtual vertices is derived directly from their local curvature using Eq. 1.3.

Vertex models continue to be developed given their very attractive way to handle phenomenon such as grain boundary anisotropy (see for instance [17]), their accurateness and their computational performance. However, a much greater effort must be done (in terms of implementation) in order to handle coherent topological transformations, mainly in 3D, comparatively speaking to models such as Monte Carlo, Cellular Automata, Phase-Field or Level-Set for which topological transformations are treated automatically.

Other FT approaches different from the Vertex model can be found in the literature. In [9] in the context of 2D GG, the authors introduce a FT methodology where instead of using a dissipative equation of motion, a variational approximation of the curvature of the interface is preferred, to compute the velocity at every vertex except for those being triple junctions (see section 6.4 of [9] for more information). Triple junctions are positioned in order to respect the equilibrium

in a similar manner to the works of Frost et al. [102]. Moreover, the study presented in [9] is not only dedicated to the modeling of 2D GG but also to the more general context of surface minimization, for which the author gives many examples in 2D and 3D. Another example of FT models is given in [10], where a FE model, based on a variational formulation for GB motion by viscous drag, is used to solve the equations governing the grain boundary motion of an arbitrary shaped surface and its interaction with SPPs in 3D. The model was later extended in [11, 12] to take into account motion by curvature flow but in the context of a single grain boundary; hence, real polycrystal structures were not tested.

Finally, another Front-Tracking model was proposed in [13] where the movement of grain boundaries is dictated by the minimization of the total energy of the system by moving each node along the energy gradient towards a lower total energy state. In this model the energy term can be defined via SE, GB energy, and non-conservative reaction terms. The evaluation of the energy gradient field is determined for every node via a *local* finite-difference method centered at the node in question, and the movement is made in the direction of the maximal energy reduction. The authors tested this model in the context of isotropic and anisotropic GBM where melting was simulated through a second phase, attributing an "energy-based" melting-fraction function where the more the melting fraction deviates from the target value, the more the melting/crystallization energy dominates. However, no information regarding the GB anisotropy function was given, even though some results shows anisotropic GB behavior between the solid-solid and solid-liquid boundaries.

1.3.4 Phase Field models

Another model that can be used for the modeling of microstructural phenomena at the mesoscale, is the Phase-Field (PF) model [103]. The first applications of these models in the context of microstructural evolutions were aimed at the simulation of the solidification of a melted substance (hence the name "Phase-Field") [104, 105]. This model has been extended to the study of many physical phenomena such as solid-state phase transformations, crack propagation, dislocation dynamics, GG and ReX, and more. The model is developed under the assumption that the definition of a microstructure, both the compositional/structural domains and the interfaces, can be made, as a whole by using a set of *field variables* [106]. These variables are defined as *conserved* (meaning that they have to satisfy a given local conservation condition) and *non-conserved*. The definition of microstructures using such a way, reduces the topological complexity of a multidomain problem, as interfaces between grains or between phases evolve implicitly when these variables evolve. Furthermore, the evolution of these field variables (hence of the microstructural state they describe) is given by the minimization of the free energy of the system F, which can be expressed as [103]:

$$F = \int_V \left[f(c_1, c_2, ...c_n, \eta_1, \eta_2, ...\eta_p) + \sum_{i=1}^{n} \alpha_i (\nabla c_i)^2 + \sum_{i=1}^{3} \sum_{j=1}^{3} \sum_{k=1}^{p} \beta_{ij} \nabla_i \eta_k \nabla_j \eta_k \right] dV, \tag{1.15}$$

where f is the local free energy density, c_i and η_i correspond to the conserved and non-conserved field variables and α_i and β_{ij} are gradient energy coefficients. The choice about the form of the functional f depends in general on the physical mechanism being simulated and the parameters being used. Two models aiming to simulate the same physical process can indeed have a different form of the local free energy density function and still give similar results for the predicted steady-state.

Once the expression for the local free energy density f is set by the physical phenomena to be simulated, the total free energy F can be minimized solving a set of PDEs denominated the Cahn-Hilliard [107] and Allen-Cahn [108] equations:

$$\frac{\partial c_i}{\partial t} + \nabla M_{ij} \nabla \frac{\partial F}{\partial c_j} = 0, \tag{1.16}$$

$$\frac{\partial \eta_p}{\partial t} + L_{pq} \frac{\partial F}{\partial \eta_q} = 0, \tag{1.17}$$

where L_{pq} and M_{ij} are kinetic parameters related to the dissipation of energy of the system (usually atom or interface mobility).

The PF model was first applied to GG by Chen et al. in [109]. In this context, the PF method uses a set of non-conserved field variables η_i in order to define the structural domain and the interfaces of the microstructure. The use of several field variables in this context was later denominated in [110] as the Multi Phase-Field (MPF) method, as opposed to the PF method, which only employs one field variable intended to simulate dual phase change problems [111]. In [109], these field variables are defined over a 400x400 squared lattice, taking values in the range [-1,1], and are set to smoothly evolve at interfaces between grains (of course this is not strictly possible given the discrete properties of Eulerian meshes). The local free energy density used was:

$$f = \sum_{i=1}^{p} \left[-\frac{k_\alpha}{2} \eta_i^2 + \frac{k_\beta}{4} \eta_i \right] + k_\gamma \sum_{i=1}^{p} \sum_{j>i}^{p} \eta_i^2 \eta_j^2, \tag{1.18}$$

where k_α, k_β, and k_γ are phenomenological parameters. Furthermore, the free energy expression used in [109] for the non-conserved field variables derived from Eq. 1.15 is given by:

$$F = \int_V \left[f(\eta_1, \eta_2, ...\eta_p) + \sum_{i=1}^{p} \frac{\beta}{2} \nabla^2 \eta_i \right] dV, \tag{1.19}$$

where $\beta_{ij} = \delta_{ij}\beta/2$, δ_{ij} is the Kronecker delta, and the term $\nabla^2\eta_k$ corresponds to the laplacian of the field variable η_k.

Using Eq. 1.18 in Eq. 1.19, then replacing F in Eq. 1.17 and ignoring the cross terms of the kinetic dissipation energy parameter L_{pq} (hence $L_{pq} = L_p$), the final set of kinetic equations can be written:

$$\frac{\partial \eta_i}{\partial t} = -L_i \left[k_\alpha \eta_i + k_\beta \eta_i + 2k_\gamma \eta_i \sum_{j \neq i}^{p} \eta_j^2 - \beta \nabla^2 \eta_i \right], \quad i = 1, 2, ...p. \tag{1.20}$$

which in [109] was solved using a forward Euler technique. Results for the evolution of this model using 36 field variables (i.e. 72 orientations) are given in Fig. 1.4. Of course, in such a case, multiple grains were attributed to one single phase field, allowing coalescence. The authors claim that a large number of field variables is ideal to simulate this kind of problem. They provide results indicating that the growing exponent n fits almost perfectly the value of 0.5 for simulations using 4 and 36 field variables, even though simulations using 4 field variables did not obtain a good agreement with classical microstructure topologies. Here the model uses $k_\alpha = k_\beta = k_\gamma = 1.0$, $\beta = 2.0$, and $L_i = 1.0$ for all i and a time step of $\Delta t = 0.1$.

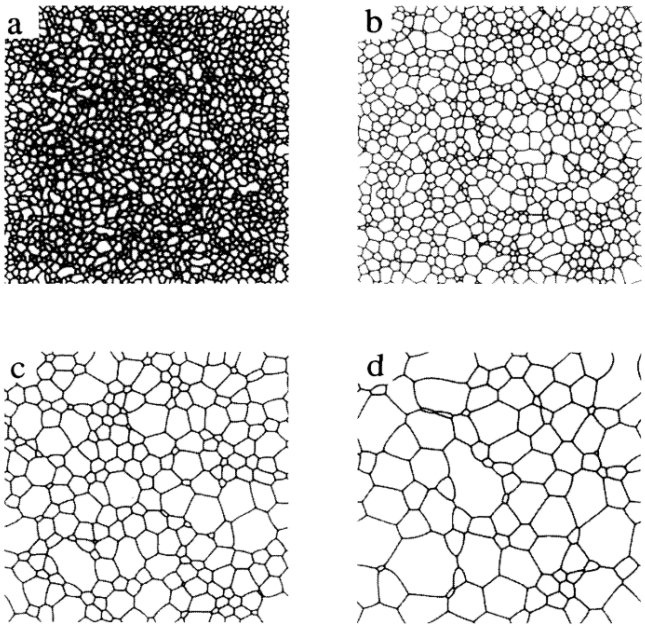


Figure 1.4: Evolution of the grain boundary network due to GG for the phase field model of Chen et al., using $k_\alpha = k_\beta = k_\gamma = 1.0$, $\beta = 2.0$, and $L_i = 1.0$ for all i and a time step of $\Delta t = 0.1$, for the time steps: a) 200, b) 1000, c) 4000 and d) 10000. Source: [109].

The model presented by Chen et al. denominated *Continuum-Field* model,

has been further studied [112, 113, 114, 115, 116] and extended to take into account inclination-dependent and misorientation-dependent anisotropic grain boundary properties [117, 118, 119, 120, 121], GBM under the influence of SE [122] and ReX [123].

A second kind of MPF model for the simulation of microstructural evolutions was introduced by Steinbach et al. [110, 124, 125, 126, 127]. Steinbach adopted a slightly different approach where field variables η_i take values in the range [0,1] (this was later also adopted by Chen's group in his approach) and where:

$$\sum_{i=1}^{p} \eta_i = 1. \tag{1.21}$$

This is given by the fact that for Steinbach, the distinct phases represent volumetric fractions, thus their sum not being able to be greater than 1 in any region of the domain. Similarly to Chen's works, Steinbach's works were also extended to misorientation-dependent anisotropic grain boundary properties [128, 129, 130] and ReX [131, 132, 133].

1.3.5 Level-Set models

The final category of numerical approach able to simulate multidimensional problems (hence microstructural evolutions) discussed in this state of the art, is the denominated LS model.

Level-Set models were introduced by Osher et al. in [134], as a simple, and, robust alternative for the analysis of moving interfaces and shapes. It uses the concept of level-set functions to implicitly define sharp interfaces (contrary to PF models which use a diffuse description of interfaces), i.e, the LS method defines an interface $\Gamma(t)$ as the zero-isovalue of a smooth function $\phi(x,t)$, defined over a domain Ω:

$$\phi(x,t) = 0, \quad \forall\ x\ \in\ \Gamma(t) \tag{1.22}$$

where typically $\phi(x,t)$ takes positive values in the region delimited by $\Gamma(t)$ and negatives outside. This formulation is of great significance as boundaries are defined through time as the evolution of a signed smooth field $\phi(x,t)$, and complex operations in the topology of $\Gamma(t)$ (coalescence, appearance and disappearance of domains) are handled at the level of $\phi(x,t)$, instead of being manipulated directly on the definition of interfaces as in vertex-models. This model has been successfully applied to the propagation and interaction of fronts in multiple numerical models [135, 136].

In practice, the form of $\phi(x,t)$ is commonly taken as the euclidean distance to the interface $\Gamma(t)$:

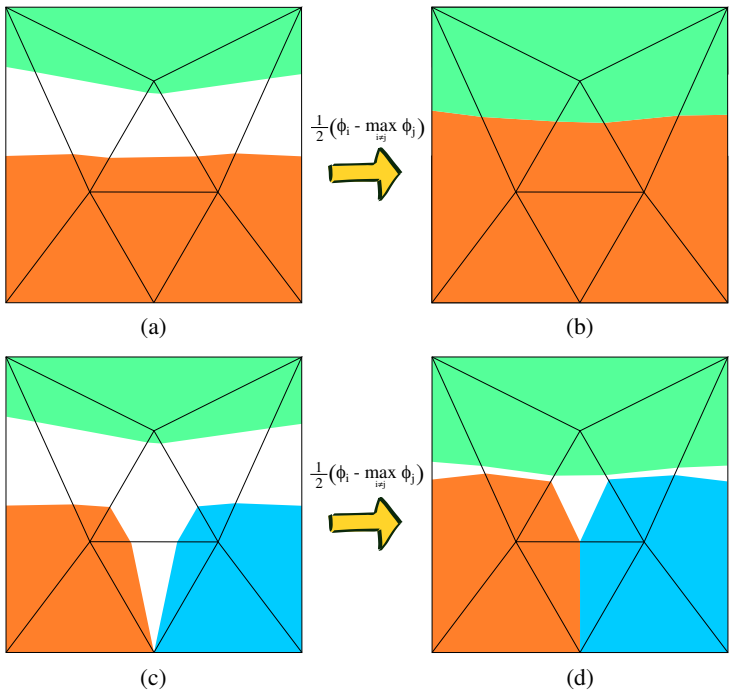


Figure 1.5: Global treatment in order to eliminate non-physical vacuum regions on a FE discretization. Two colored LS (a) with no treatment of vacuum regions, (b) result after applying Eq. (1.24). Three colored LS (c) with no treatment of vacuum regions (d) result after applying Eq. (1.24). Source: [19].

$$\phi(x,t) = d(x,\Gamma(t)), \quad \forall\ x\ \in\ \Omega \tag{1.23}$$

where $d(x,\Gamma(t))$ is the signed distance function of a point x to the interface $\Gamma(t)$ which can be obtained thanks to the so-called LS *reinitialization* operation (see below section 1.3.5). This framework is very advantageous as multiple geometric properties (normal, curvature...) of $\Gamma(t)$ can be obtained as a function of $d(x,\Gamma(t))$, thus as a function of $\phi(x,t)$.

The modeling of multidomain (more than 2) systems forming a partition of the global domain and using the LS method was proposed in [137]. Here, the authors propose to give each simulated domain a LS field, and introduce a *coupling* mechanism for the treatment of non-physical overlapped (or vacuum) regions:

$$\hat{\phi}_i = \frac{1}{2}\left(\phi_i - \max_{j \neq i} \phi_j\right), \quad \forall i = 1 \dots N, \tag{1.24}$$

where $\hat{\phi}_i$ is then used as the corrected LS function ϕ_i. The effect of this treatment is illustrated in Fig. 1.5 for a 2D configuration.

The works presented in [137] set the basis for the treatment of physical problems such as GG and ReX which are strongly influenced by the motion of multiple

junctions.

The evolution of interfaces can be simulated in this context by the transport equation:

$$\frac{\partial \phi_i}{\partial t} + \nabla(\phi_i \vec{v}) = 0, \tag{1.25}$$

where $\vec{v}$ is a velocity field defined over Ω, which is determined from the physical process being simulated (see Equations 1.3 and 1.5).

The studies presented in [137] were then extended in [138] to take into account heterogeneous GB properties (for example, as a function of the length of interfaces) and bulk properties (for example, as a function of the total area of a grain) using the works presented in [139]. Additionally, in [138], the restriction on the creation of voids and overlaps was implemented in a variational sense (hence being enforced in the resolution of the FE problem). However, the computational performance of the methods presented in [138] is still questionable and in general, simpler LS models are used.

The Level-set method was further studied by Bernacki et al. [5, 140] where the concepts in [137] were extended to take into account SE values. Further works on the development of the LS method involve ReX [141, 6, 7, 142], heterogeneous and anisotropic grain boundary properties [143, 144, 23, 145], the influence of second-phase particles [146, 147] and the coupling of the LS method with Crystal Plasticity (CP) simulations [148, 149]. Additionally, some of these works concerning the enhancement of the LS-FE algorithm which are also used in the context of this manuscript (mainly in chapters 2 and 3), are summarized in section 1.3.5.

Moreover, the LS method has been also applied in the context of a fast Fourier transform (FFT) resolution for GG and SRX [150, 151, 152, 153], being able to perform simulations with a very large number of grains (up to 650000 initial grains). FFT-LS approaches are, however, limited to a context without deformation (SRX, GG, MDRX) as they rely on the use of regular grids, contrary to the FE-LS model which is able to employ unstructured meshes (hence to remesh) for simulations subjected to very large deformations [7].

Global LS functions and Reinitialization

Until now, the treatment of interfaces in a polycrystal has been described assuming that each LS function ϕ_i represents the boundary of a grain. In simulations involving thousands of grains, however, which are generally necessary for physical representativity, the computational cost of such approach becomes prohibitive because Eq. (1.25) must be solved thousands of times on the whole domain, additionally, of course, to the memory management problem of storing the LS fields.

This problem can be overcome by the use of Global Level-Set (GLS) functions, which include multiple grains in a single distance field [146, 8]. Coloring/Recoloring techniques make it possible to identify sets of non-connecting grains that can share a GLS. In the case of a GG simulation, topological events (mostly grain vanishing) occur, which often leads non-connecting phases to become adjacent. This is why dynamic re-coloring methods [8, 146], which transfer grains between GLS functions, are used to ensure that the GLS functions remain valid. With this technique, it is possible to limit, independently of the total number of grains, the number of GLS functions (and thus the number of diffusion equations to solve) to a few tens in 3D and less than 10 in 2D, which is computationally affordable.

An additional consideration is that, after the resolution of the partial differential equations (PDE) describing the behavior of the polycrystal at high temperatures, the LS fields may become distorted, losing its metric properties inherent to distance functions [154], used in the formulation of the FE problem. Hence, it is necessary to *reinitialize* the LS functions, i.e. to maintain them as signed distance functions. Many methods have been proposed in the literature to perform this task, for instance, fast marching methods on regular grids [155, 156, 157] or solving a Hamilton-Jacobi equation [135] or optimized convective-diffusive formulations [154] both using unstructured finite element meshes. Other methods (such as the one used in the present work in chapter 2) prefer to recompute the distance to the iso-zero level-set in a purely geometric manner with a parallel-efficient algorithm based on spatial partitioning trees [158].

1.3.6 Summary and discussion

Monte Carlo methods were the first FF numerical framework used to simulate microstructural evolutions, they use a very simple and generic concept, able to be implemented in a straightforward way, both in a sequential and parallel context. However, their stochastic component and their definition of simulated time step (the Monte Carlo step) make them dependent on a time-scaling strategy. This problem can be avoided by the use of Cellular Automata models as they are based on a thermodynamic basis, hence relying on a physical space and time step, although they need a slightly more complex implementation in a parallel context, as CA models rely on the information of the vicinity of cells. MC and CA models, present however a weakness concerning the discretization of domains, as a homogeneous static discretization must be set at the beginning of the simulation, hence being very difficult to optimize particular situations (e.g. the size of inserted nuclei is mostly defined by the size of one cell). This "static" discretization via pixels/voxels also gives particular difficulties for problems subjected to very large deformation, such as those intended to simulate DRX in the context

of industrial TMTs.

One of the most important results of Vertex models (the Kawasaki's kind) is that multiple points arrange themselves in order to fulfill their equilibrium condition [16, 15], contrary to Vertex models (the Frost's kind) that impose their equilibrium without any thermodynamic consideration [102]. Thus, all Vertex models should give very similar (and very accurate [25]) results except for the initial, very quick, transient state obtained when a multiple junction is formed. Moreover, although the computational performance of vertex models should be essentially higher than models using a Front-Capturing approach, the implementation of Vertex models becomes very complex, especially in a 3D context, as all topological changes on the microstructure need to be addressed "manually" by algorithms intended to do so. Additionally, another weakness of the Vertex model is the impossibility to take into account intragranular phenomena, such as the ones studied in [159].

The MPF model uses a diffuse interpretation of boundaries, meaning they are defined within a domain volume (in 3D or surface in 2D) by the field variables. Furthermore, the definition of the free energy density function f (see Eq. 1.15) is not straightforward (both to derive and to give a physical meaning to each term) which have to lead to different definitions of this functional in the literature, yet giving similar results for a given phenomenon. Additionally, the minimization of the free energy can lead to poor computational performance because of the co-dependent formulation of the evolution of field variables (i.e. the minimization of free energy for all field variables must be performed simultaneously) and the definition of interfaces (diffuse interfaces) which is, in general, questionable as it is not representative of the physical "thickness" of grain boundaries. These difficulties of the MPF model can be solved by the use of LS methods based on distance functions, as they are able to define sharp interfaces and rely on the transport equation (Eq. 1.25) to make evolve the domains, using a predefined boundary velocity. However, the kinetics formulation of the LS approach can make it difficult to take into account certain properties of the GB [143, 23]. MPF and LS models still depend on the availability of a numerical solver, hence their implementation may become a difficult task if such tools are not available. Nonetheless, their inherent capabilities to handle topological events make them very attractive models for the simulation of microstructural evolutions.

The principal weakness of the MPF and LS models based on a FE resolution remains their computational cost, which is commonly high for simulations involving millions of FE elements (a normal number of elements in the context of GG and ReX for a representative test case). An alternative to the FE resolution is to use a FFT resolution which has been shown to obtain better performances [152, 153, 123], however, limiting their study to a no-deformation context.

A particular point of interest is the one concerning the modeling of DRX in context of very large deformations, for which the only works addressing this situation, are those using a LS-FE framework with unstructured FE meshes [7]. This is a major advantage of the LS-FE model over all other cited models as for simulations intending to characterize real TMTs, and in general, for the metal forming industry, this aspect is of prime importance.

As mentioned in the introduction of this thesis, the procedure having the greatest impact in the performance of the FE-LS framework is the remeshing [141], which is, a priori, necessary for two reasons: The first is to (hypothetically and ironically) reduce the computational cost of the whole numerical framework, by maintaining a refined mesh in the regions near GBs (i.e. refined within a given thickness centered at the GBs), as these regions are the only ones that have a relevant evolution in terms of LS field data [160], while the bulk of the grains is remeshed with a much coarser mesh. It could be a very dangerous reasoning as, for domains being substructured with thousands of subdomains (grains), it is possible that the refined thickness becomes of the same magnitude as the grain size, hence refining a significant percent of the total domain (if not all the domain), and thus spending time in the remeshing step where a static mesh (refined everywhere) could conceivably produce similar results, mainly in 3D. The second and most relevant reason for which remeshing is necessary, is precisely that in the context of very large deformations, the mesh needs to be updated so its quality does not become too poor. The next chapter will be dedicated to giving a brief introduction to metric-based remeshing, then a more quantitative point of view regarding the use of remeshing strategies for the LS-FE model will be given, along with the introduction of a new body-fitted remeshing algorithm, specially developed for multidomain systems using a LS or a MPF description.

Finally, only two models use directly the definition of a velocity for the GBM as mentioned in section 1.1: the Level-Set model and the Frost's Vertex model. All other models are based on either stochastic or thermodynamic formulations, still reproducing similar results. The models presented in this thesis will directly use the velocity equations 1.3 and 1.5 as an approximation for the movement of grain boundaries in context of GG and ReX while thermodynamic formulations will be avoided.

Résumé en Français du Chapitre 1

Ce chapitre détaille les bases scientifiques pour les travaux développés et expliqués par la suite sur ce manuscrit de thèse. Il est principalement dédié à l'introduction des notions de recristallisation et de croissance de grains. Les principales méthodes à champ complet (ainsi que leur historique) utilisées/développées dans ce contexte sont rappelées.

Les avantages et faiblesses de ces différentes méthodes existantes sont mises en évidence, montrant ainsi les forts avantages de l'approche LS en très grande déformation, la facilité d'implémentation des modèles Monte Carlo et Cellular Automata et le faible coût numérique des modèles Front-Tracking.

Chapter 2

Discussion of the FE-LS methodology

As discussed in chapter 1, the LS method is a powerful approach to model dynamic interfaces in the context of large deformations. The LS method has been applied to the simulation of microstructural evolutions as Grain Growth and Recrystallization at the mesoscale [7]. Interfaces between grains are implicitly described in an Eulerian framework, as the zero-isovalue of the LS fields and their evolution is governed by a transport formulation in the form of partial differential equations. The LS approach circumvents the notoriously difficult problem of generating interface-conforming meshes for geometries subjected to large displacements and changes in the topology of the domains.

Generally, to maintain high accuracy when using the LS method, moving interfaces are captured by a locally refined FE mesh with the help of mesh adaptation techniques. In a microstructural problem, the large number of interfaces and the fine mesh required in their vicinity make the mesh adaptation process very costly in terms of CPU-time particularly in 3D [8].

In the first part of this chapter, some of the concepts regarding the remeshing of numerical domains will be given, mainly in the context of anisotropic mesh adaptation and body-fitted remeshing. We will see that one of the methods to model microstructural evolutions (the so-called Level-Set method) can be coupled with a remeshing technique to obtain body-fitted meshes of polycrystal domains.

Then, a new mesh adaptation strategy is developed and applied to LS-FE simulations in the context of GG. It maintains the benefits of the classical Eulerian LS framework, while enforcing at all times the conformity of the FE mesh to implicit interfaces by means of local remeshing operations. Special treatments for vacuum regions have been adopted and will be presented within the generalization of a previous adaptation algorithm presented in [161]. A detailed study concerning the source of errors will be presented and compared for different remeshing

strategies including the presented one. Finally, we will illustrate how the new method decreases the requirement in mesh density while maintaining the accuracy at the interfaces, and a discussion regarding the best remeshing strategy to model LS-FE in this context will be given.

This chapter has been partially published in [19].

2.1 Introduction

Because most virtual polycrystalline microstructure generation tools are based on the concept of Voronoï cells or Laguerre Voronoï cells [55, 56, 57], meshing virtual microstructures does not raise any major challenge. The generation of an interface-conforming (i.e. body-fitted) mesh usually consists in discretizing cells facets, and then the volume within each cell [162]. However, for real polycrystals, observed using Scanning Electron Microscopy (SEM) or 3D X-ray imaging techniques [53, 54], a Voronoï/Laguerre-Voronoï space partitioning is, of course, not accessible. While there has been much research on real 3D microstructures meshing methods [163, 164], the generalization of these methods to massively multiphase materials such as polycrystals is not straightforward [165, 166]. The main challenge is linked to multiple junctions, namely interfaces between more than two grains where obtaining both high mesh quality and fidelity with respect to experimental data can be complex.

Once a mesh has been generated, modeling large plastic strains and subsequent microstructure evolutions such as ReX, GG or solid/solid phase transformations (SSPT) in a FE framework is also very challenging. As a consequence, many researchers have chosen to avoid the use of meshes where grain boundaries are explicitly meshed (i.e. with a conformal mesh), and instead, use implicit interface approaches (also called Eulerian approaches) such as the LS [134] or the MPF method [111] as described in chapter 1. While results using explicit interface methods are restricted to limited deformations, implicit interface methods have given access to the modeling of a wider range of thermomechanical phenomena. For instance, the LS method has been successfully used to simulate SRX or DRX in context of large deformations [154, 146, 6, 7]. However, the absence of a conformal mesh at grain boundaries typically seems to require a finer discretization [140] which could be a difficulty in terms of numerical cost mainly in 3D [8]. Thus, there is an interest for alternative methods with similar capabilities and robustness as the LS method, but based on explicit and reasonable interface meshing/remeshing.

In this chapter, a new methodology, based on a previous work originally applied to mechanical fracture problems [161], is proposed to generate conforming FE meshes using as input a LS or MPF data set. First, a brief introduction to the concepts of metric-based remeshing will be given along with the concept of the body-fitted mesh regeneration algorithm proposed in [161]. Then, based on these works, a new mesh adaption technique will be proposed to handle large deformations and displacements of GB interfaces. The abilities to generate and adapt conformal FE meshes thanks to the intermediate use of LS functions are particularly useful for the modeling of topological events such as grain disappearance during GG. An alternative method based on the full reconstruction of the computational mesh at each time step is presented in [142]. Here, all pro-

posed algorithms are based only on local mesh modifications, and not on full mesh reconstructions. Of course, the new remeshing algorithm will be used in the context of GG mechanism using a LS-FE framework and will be compared to a classical front capturing remeshing technique using the same numerical framework [5, 167, 168, 169]. A detailed description of the potential sources of errors during a simulation for each approach is presented, tested and compared, using a specific test case featuring an analytical solution when subjected to capillarity effects, namely the circle shrinkage case. Then, comparisons involving accuracy for more demanding test cases are given, and finally, comparisons of CPU time are performed for a large 2D polycrystal involving 10000 initial grains.

2.2 Remeshing strategies in context of FE modeling of multiphase systems

When using an Eulerian approach, the mesh is generally adapted periodically through time in the regions near to the interfaces [5, 167, 7, 170] and this process leads to high CPU times. One solution is to combine the benefits of Lagrangian and Eulerian methods: Arbitrary Lagrangian Eulerian (ALE) methods. Normally, on an ALE framework, the interfaces are described in a Lagrangian way (body-fitted interfaces) and the inner and outer nodes (those not belonging to the interface) behave in a Lagrangian-Eulerian way (see [171] for a good example in the context of large deformation of interfaces). However, a standard ALE method doesn't allow topological changes on the mesh and this can be an issue: the mesh can get tangled depending on the update technique used to move each vertex [172].

In the context of the DIGIMU project, tens of thousands of interfaces need to be handled. In the case of an implicit formulation of these (using LS or MPF methods), the number of mesh elements required to treat them is very high and the number of operations performed to update the mesh or the remeshing during interface migration can give as a result a prohibitive CPU time, mainly in a three-dimensional context.

Instead of using a pure ALE method or a very refined mesh on an Eulerian formulation, a possible solution could be the use of body-fitted re-meshing algorithms to make an explicit mesh over the implicitly-defined moving interface. In this section, the work of some authors in the field of LS methods coupled with body-fitted mesh generation algorithms will be discussed([173], [174]) along with some basic concepts used on metric-based remeshing strategies and the computation of mesh quality.

2.2.1 Mesh quality

The quality of the mesh is the factor that defines whether or not a mesh (or a subset of it) is acceptable. Normally, the quality of an element k is given by a value $Q(k) \in [0, 1]$ where $Q(k) = 1$ represents a perfect element (equilateral triangle in 2D and a regular tetrahedron in 3D in the context of Euclidean space) and $Q(k) = 0$ a fully degenerated one. $Q(k)$ is invariant to rotation, translation and change of orientation.

Authors in the literature have different approaches to define an expression for $Q(k)$ because not all of them have the same needs in terms of shape and size. Here we will only consider some examples used in the context of a mesh made by simplexes:

$$Q(k) = c_0 \frac{\rho(k)}{D_k}, \qquad \text{used on [174]} \tag{2.1}$$

$$Q(k) = c_0 \frac{|k|}{h^d}, \qquad \text{used on [173, 175]} \tag{2.2}$$

where for an element k, D_k is the diameter of its circumscribed circle/sphere, $\rho(k)$ its in-radius, $|k|$ represents its volume and finally, h the average length of its edges. d represents the space dimension of the calculation and c_0 is a constant used to scale the values of $Q(k)$ in the range of $[0, 1]$. Note that each property of k can be calculated on the Euclidean space (represented by a canonical Euclidean metric field) or in any other metric space M defined by a positive definite symmetric tensor (as we will see for anisotropic mesh adaptation).

The value of $Q(k)$ can be coupled with other expressions in order to have multiple criteria for one element, for example in [173]:

$$Q(k) = \min(c_0 \frac{|k|_M}{h_M^d}, h_M^d, \frac{1}{h_M^d}), \tag{2.3}$$

where the extra terms take into account the size of the element. Here the subscript M means that all quantities are calculated on the metric space defined by M.

2.2.2 Metric-based Remeshing

The procedure to enhance mesh quality taking into account the existing topology of an unstructured mesh can be done using a remeshing approach. When the criteria of the quality is measured using a scaled and/or rotated euclidean metric (this is for $M = c_{s(x)}\boldsymbol{R}_{(x)}$, where $c_{s(x)}$ is a size factor and $\boldsymbol{R}_{(x)}$ is a rotation matrix), the remeshing procedure will try to obtain elements in the shape of equilateral triangles (in 2D) or regular tetrahedra (in 3D) with edges of the size of $c_s(x)$. Moreover, if the metric space takes a more general form (e.g. $M = \boldsymbol{R}_{(x)}\boldsymbol{T}_{(x)}\boldsymbol{S}_{(x)}$, with $\boldsymbol{S}_{(x)}$ a scaling matrix and $\boldsymbol{T}_{(x)}$ a shearing matrix), the remeshing algorithm will try to approach the forms of more irregular triangles (or tetrahedra).

A given algorithm (the remeshing algorithm) will perform topological changes on the connectivity, insertion/deletion of nodes and/or displacements in the position of the nodes of the existing mesh, in order to obtain these shapes. A remeshing procedure can be established by the following steps [173]:

- iteratively applying *remeshing operators* on a subset of the mesh Ψ_o, obtaining a new possible replacing subset Ψ_i for each iteration.

- Classify each possible replacing subset Ψ_i based on its quality $Q(\Psi_i)$, for example using

$$Q(\Psi_i) = \min_{k \in \Psi_i}(Q(k)), \qquad \text{used on [173, 175]} \tag{2.4}$$

  and take the best subset Ψ_m.

- Replace Ψ_o by the best configuration Ψ_m if the quality of Ψ_m is greater than quality of Ψ_o.

.

A good example of metric-based remeshing strategy is the one proposed by Gruau et al. in [175], here the authors used Eq. 2.3 to compute the quality of a simplex k with vertex $S_0, ...S_d$, using the following definitions:

$$|k|_{M(k)} = |k|\sqrt{\det(M(k))}, \tag{2.5}$$

$$c_0 = \frac{d!}{\sqrt{d+1}} 2^{d/2}, \tag{2.6}$$

$$h_{M(k)} = \left(\frac{2}{d(d+1)} \sum_{0 \leq i < j \leq d} (S_j - S_i)^T M(k) (S_j - S_i) \right)^{1/2}, \tag{2.7}$$

where the desired metric of simplex k, $M(k)$, is obtained as the average of the desired metric at vertex $S_0, ...S_d$:

$$M(k) = \frac{1}{d+1} \sum_{i=0}^{d} M(S_i). \tag{2.8}$$

The algorithm proposed by Gruau et al. then iterates on the nodes and the edges of the mesh, one by one, by proposing new configurations for a *patch*[1] of elements, (see Figure 2.1(a-b) and 2.2(a-b)). The new configurations will be proposed by a reconnection operator named *star-connection*, developed by the authors (see Figure 2.1(c-h) and 2.2(c-e)).

For each one of the new configurations, the quality will be computed. Finally, the algorithm will choose the one having the best quality for its worst element as denoted by Equation 2.4.

Other good examples of anisotropic remeshing strategies can be found in [176, 177, 178, 179, 180, 181, 182, 183] for metric-based remeshing. One of the differences between these approaches and the one in [175] is that instead of using

[1] a patch of elements in this context is defined as the mesh subset Ψ_o^j obtained by the **union** of elements containing the node j or the **intersection** of the elements containing the nodes in edge j

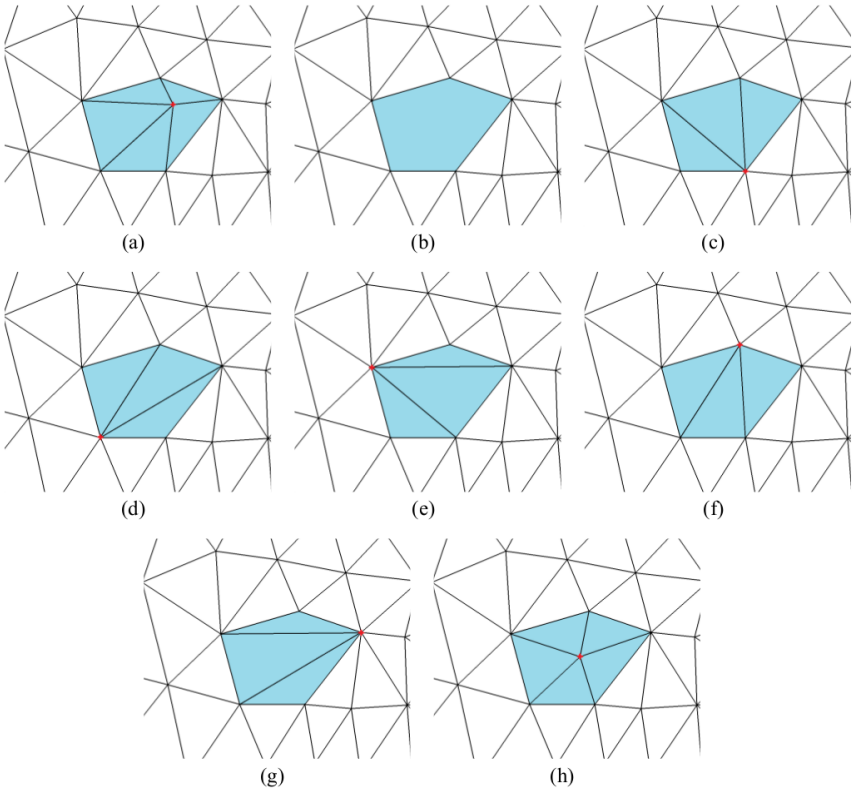


Figure 2.1: 2D Examples of the operations made by the *star-connection* operator applied on a node (patch defined by the **union** of elements containing the node): a) initial configuration, b) region to optimize, (c-h) candidates for the best configuration, Source: [173].

the *star-connection* algorithm, the authors perform topological changes in the mesh by the use of local remeshing operators, such as: vertex-smoothing, edge-swapping, edge-collapse and edge-splitting. These remeshing operators will be defined and used in chapter 3.

2.2.3 Body-Fitted remeshing of implicit defined interfaces

An extension of the work of Gruau et al. was proposed by Shakoor et al. in [161, 173]. The main new feature being the possibility to create a body-fitted mesh of an implicitly defined interface and to make anisotropic mesh adaptation at the same time.

The *interface fitting algorithm* presented in [173] can be decomposed in two steps: fitting the interface by cutting the edges and adapting the mesh near to the interface to obtain good-quality elements.

Initially, the algorithm will find each edge crossed by the interface. This is done by evaluating the LS function and by storing the edges whose nodes change

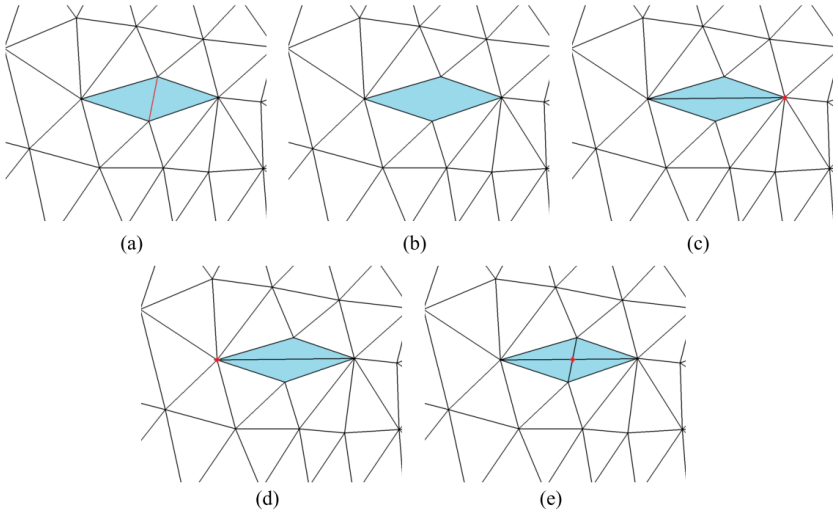


Figure 2.2: 2D Examples of the operations made by the *star-connection* operator applied on an edge (patch defined by the **intersection** of elements containing the nodes of the edge): a) initial configuration, b) region to optimize, (c-e) candidates for the best configuration, Source: [173].

sign (this means that the zero-value of the LS function lies on that edge). The next step is to split the edge (i, j) taking into account the position of the zero value of the LS function. This can be made by adding a node on the coordinates of the intersection S^{ij} [173]:

$$S^{ij} = S^i - (S^i - S^j)\frac{\phi(S^i)}{\phi(S^i) - \phi(S^j)}, \tag{2.9}$$

where S^i and $\phi(S^i)$ corresponds to the position and the value of the LS function of the node i. The inserted node S^{ij} needs to be fixed, and it can not be removed from the mesh by the adaptation process nor moved by a vertex smoothing algorithm. Yet, some re-meshing operators need to be allowed near the interface, otherwise, the resulting quality of the split elements would be very poor. In fact, once the new nodes are placed, the *star-reconnection* algorithm (see section 2.2.2) will reconstruct the mesh in the vicinity of the interface. Of course, the resulting body-fitted mesh on the interface is not allowed to be altered, thus two consecutive nodes having $\varphi = 0$ must remain connected after this last step.

In practice, these constraints are very restrictive and they can lead to a very aggressive remeshing on the interface. To solve this, a new idea is introduced: *relaxation* on the interface. Basically, the remeshing algorithm will allow to perform a reconnection or to delete a node of the interface (those with $\varphi = 0$) if the relative change of the volume occupied by each phase is lower than a factor ϵ_Q. However, relaxation is only necessary if the algorithm for remeshing is not efficient enough to obtain good qualities with a few iterations of the *star-connection*

algorithm on the nodes near the interface.

The authors improved the *star-connection* algorithm to increase its robustness, leading to a very rarely need for relaxation on the volume preservation. The changes lie in the addition of a new possibility called wide gather. If the target was an edge as in Fig. 2.2 and no modification was performed (because none of the configurations lead to a better quality), the operation is triggered. Basically, the *patch* is extended to the elements which contain the nodes of the edge as shown in Fig. 2.3a, this will allow a wider range of operations on the patch including *edge-collapse* (Fig. 2.3i).

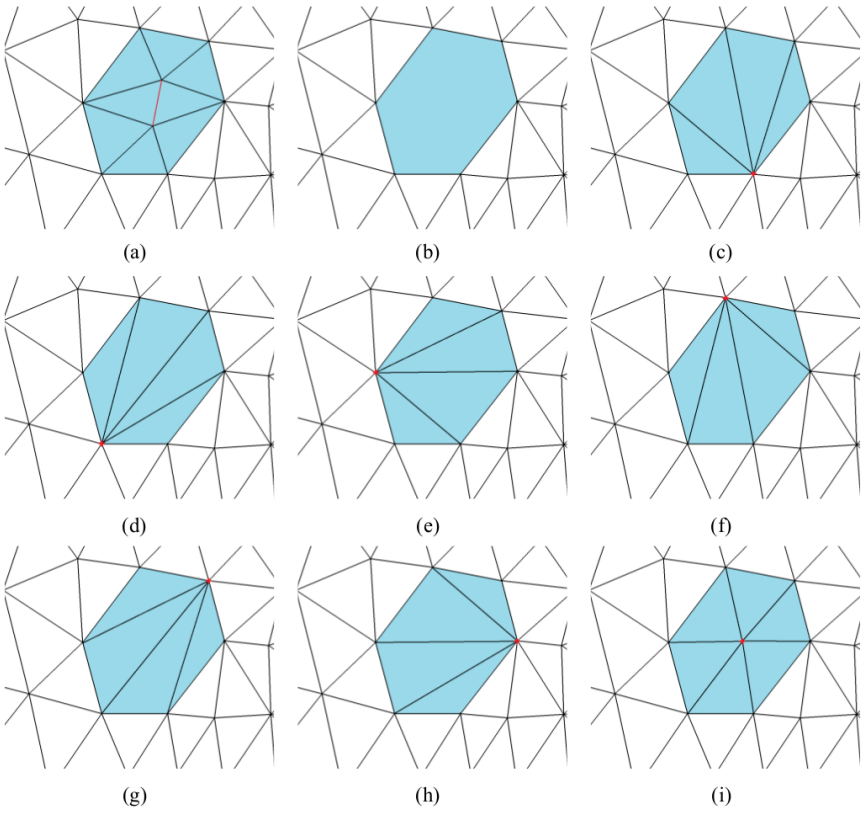


Figure 2.3: Example of a patch at the wide gathered neighboring of an edge: (a) initial patch, (b) external faces extraction, (c-i) candidates. Source: [173]

Applications of this remeshing strategy to void coalescence can be found on [184].

The algorithm described in this section has clear advantages in terms of precision at the definition of the interface (as one can define a real discontinuity in the properties at the interface). However, some disadvantages are also present. During the mesh adaptation process to obtain good-quality elements at the interface, the concatenated interpolations on the mesh could lead to numerical diffusion. Finally, even if the improved *star-connection* algorithm is very robust,

all possible combinations for the reconnection need to be tested on each node, and this may lead to disadvantages in terms of computational performance. A numerical study, aimed to quantify the profits of using a body-fitted mesh on the cases where currently an anisotropically-adapted Eulerian mesh is used, will be the purpose of section 2.5, where special attention will be given to the treatment of triple junctions and the modeling of topological events.

Alternatively, a similar work aiming to obtain a body-fitted mesh of implicitly defined interfaces, was presented in [174], where the authors, in a similar manner as in [181, 182, 183], considered the separated definition of remeshing operators (vertex-smoothing, edge-swapping, edge-collapse and edge-splitting) in order to fit and adapt the mesh.

Finally, I would like to highlight the work of Pascal Frey et al. [174], on the open source tool called MMG3D, which as said before uses almost the same principle as in [161]. This work however will not be used in this manuscript, as the methodology in [161] is already available and implemented in our computational environment.

2.3 A new fitting and joining algorithm for the remeshing of multiphase systems in a FE context

Simulations involving GG and the appearance of new grains (nucleation), are particularly complex at the topological level, because it is physically normal that domains appear and disappear constantly. A body-fitted remesher in this context would need to be robust enough to take into account these topological changes and efficient enough to treat the hundreds of thousands of interfaces present during the simulation.

Currently, the meshes used for these simulations in DIGIMU are not body-fitted. In fact, anisotropic mesh adaptation using metric-based remeshing is used in order to create a very refined mesh in the regions near to the grain interfaces. This is necessary if one intends to capture a continuous variation of properties with the help of LS functions and to obtain a highly accurate solution. However, the CPU-time used for the remeshing is too high, and limits the potential of the developed models. Another example lies in the context of the apparition of new grains. Currently in 2D, the mesher predicts the emplacement of the new nuclei and predisposes a very refined mesh on the regions where these nuclei are going to appear. However, in 3D, this process leads to prohibitive CPU-times and it has been found that it is more efficient to keep a fine isotropic mesh right from the beginning of the simulation than to apply a current remeshing strategy to treat these topological events [51].

Moreover, when generating the LS fields based on Voronoï cells or Laguerre-Voronoï cells over an implicit mesh, vacuum regions will appear because the mesh can not ensure that some of the edges (or facets in 3D) will not be crossed by the zero-isovalue of the LS functions. In other words, for a LS function defined as a signed distance function to the interface:

$$\begin{cases} \phi(x) = -\mathrm{d}(x,\Gamma) & x \in \Omega_- \\ \phi(x) = +\mathrm{d}(x,\Gamma) & x \in \Omega_+ \\ \phi(x) = 0 & x \in \Gamma \end{cases}$$

where Ω_- and Ω_+ define the exterior and the interior of grains and $\mathrm{d}(x,\Gamma)$ is the Euclidean distance between a point and the GB, and in the context of FE-LS polycrystal modeling where multiple LS functions $(\phi_i)_{i=1\ldots N}$ are used[2], the set $\Omega_\emptyset = \{x \in \Omega, \max_{i=1\ldots N} \phi_i(x) < 0\}$ will not be empty, given the non-presence of

[2] N is not necessarily the number of grains, as coloring techniques can be used to gather different grains into the same LS function and dynamically adapt this coloring throughout the simulation [146], see section 1.3.5.

nodes in the regions defined by $\Omega_\emptyset$

The presence of non-physical vacuum regions at the multiple junctions with the LS method is well known. Eq. 1.24, proposed in [137], is classically used [140, 167, 146, 158, 168, 169] to treat it.

As stated above, the set $\Omega_\emptyset$ does not contain mesh nodes, and if Eq. (1.24) is applied between two LS functions as in Fig. 1.5(a), $\Omega_\emptyset$ is empty (see Fig. 1.5(b)). However, when applying the same procedure on three LS as in Fig. 1.5(c), it can be seen that this treatment does not totally remove vacuum regions (see Fig. 1.5(d)). Another technique is proposed in [152, 145] to overcome this problem, but it will not be considered here.

2.3.1 Vaccum-less body-fitted remeshing of grain boundaries

In this section, a generalized version of the so-called interface fitting algorithm, introduced in section 2.2.3, is proposed to remove the remaining vacuum regions while constructing a conform FE mesh of the interface. This algorithm is based on purely topological mesh operations and can be easily extended to 3D. As the initial interface fitting algorithm consists simply in splitting mesh edges intersected by a LS function, and introducing this intersection as a new mesh node, it will not remove vacuum regions. Instead, entire mesh elements will be formed in these regions that do not belong to any grain. This is illustrated in Fig. 2.4(c) and (d).

In order to obtain a mesh fitted to the grain boundaries and without creating any vacuum region in the domain, the following general interface joining algorithm is proposed:

Algorithm 1 Fitting and Joining remeshing algorithm [19]

1: **for all** $T = d \dots 1$ **do**
2: **for all** T-simplices S of the mesh **do**
3: Compute all intersections between edges of S and any interface
4: Insert the barycenter of these intersections in the mesh by splitting S
5: Set the LS function associated to any interface that intersected edges of S to zero on the inserted node

where d is the number of spatial dimensions. We consider the hierarchical organization of simplicial meshes, i.e. each 3-simplex (tetrahedron) has 2-simplicial faces (triangles), which in turn have 1-simplicial edges (line segments).

A 2D example of the results obtained with this interface joining technique is shown in Fig. 2.5.

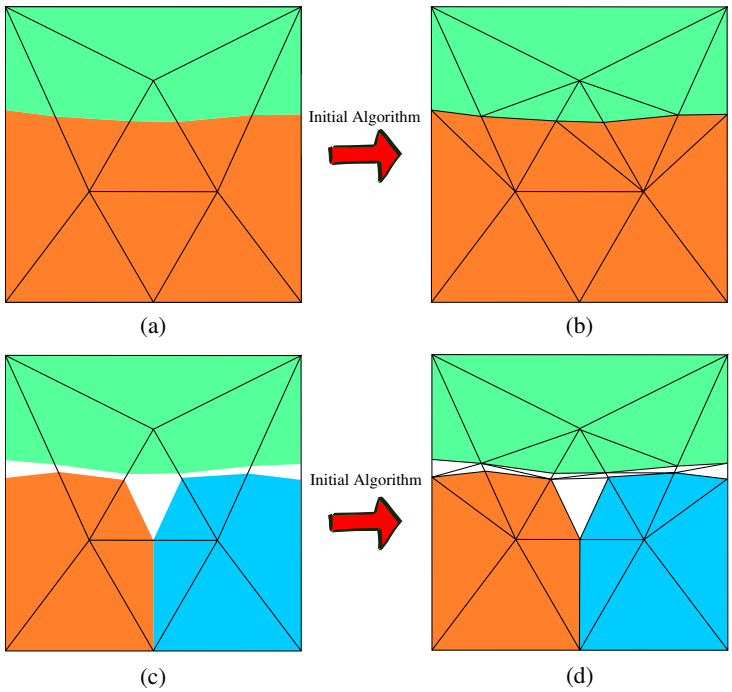


Figure 2.4: Example of FE discretization of the interfaces between two and three colored grains in 2D: (a) two implicit LS with no vacuum region, (c) three implicit LS with vacuum region and (c) and (d) results of the initial interface fitting algorithm on (a) and (c) respectively.

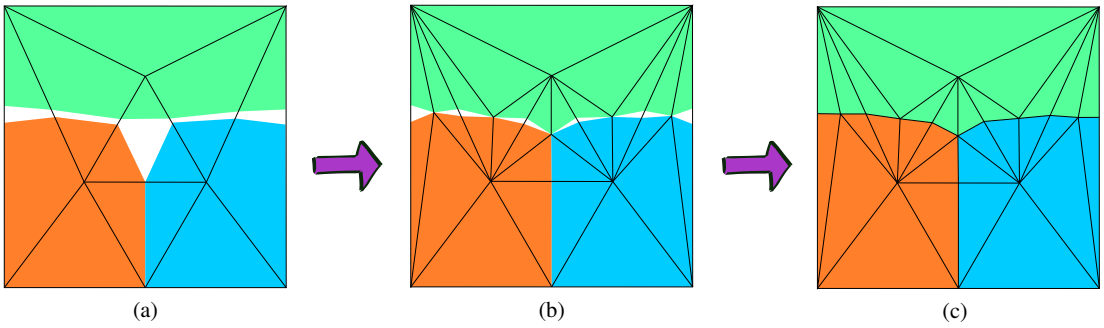


Figure 2.5: Solution of the three grains 2D problem: (a) result after using Eq. (1.24), (b) result after the first iteration of the joining algorithm, (c) result after the second and last iteration of the joining algorithm.

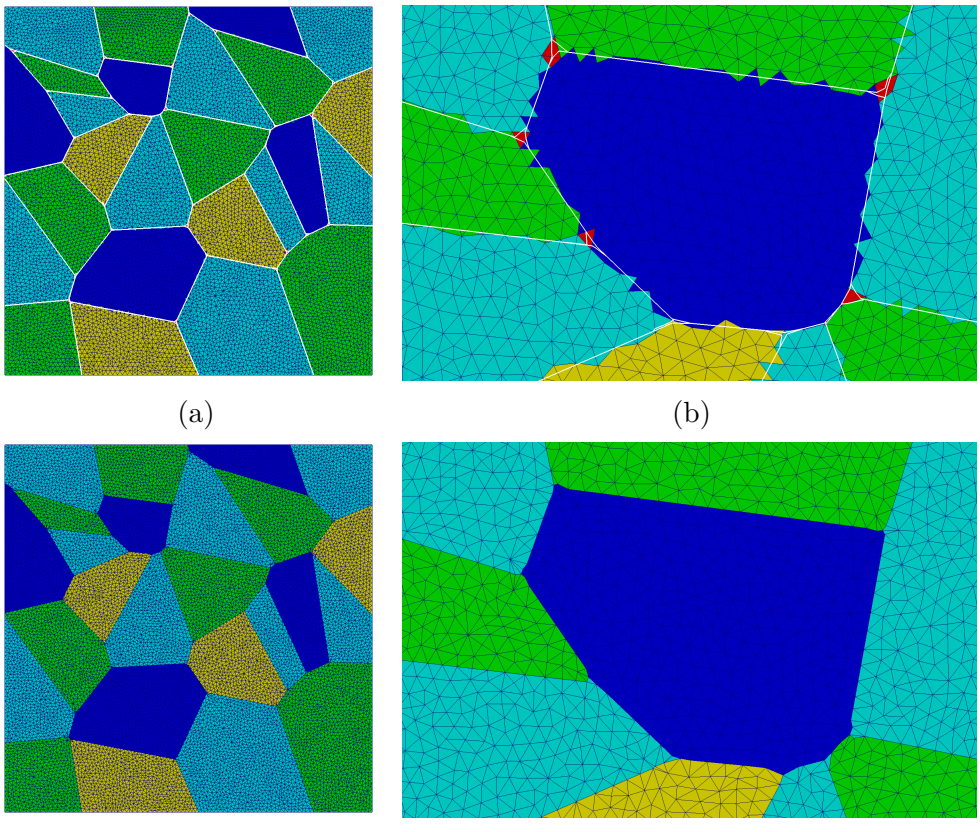


Figure 2.6: Example of a 2D polycrystal of 25 grains represented by 4 LS functions (blue, cyan, yellow, green). In (a) and (b), interfaces (white) are implicitly discretized and cross mesh elements, with elements containing vacuum regions (red) at some multiple junctions. In (c) and (d), interfaces are explicitly meshed using both interfaces joining and mesh adaption, so those vacuum regions are eliminated in the final mesh, with no significant deterioration of element shape.

This procedure prevents elements of the mesh that are intersected by interfaces to remain partly filled with vacuum. Elements that do not contain a junction (or at least three phases) are also treated by the algorithm, which avoids the creation of elements entirely contained in a vacuum. The output mesh is usually of poor quality, hence this interface joining process is generally followed by a mesh adaption step to restore a good element shape close to grain boundaries. This adaption may also follow some local mesh refinement criterion [185, 161]. An example of 2D result with a larger microstructure meshed using both the joining algorithm and mesh adaption is described in Fig. 2.6.

2.4 Grain growth modeling

2.4.1 Model and numerical method

Governing Equations

In first approximation, as mentioned in chapter 1. GG by capillarity can be described by a pure advective process. At the polycrystal scale, the velocity $\vec{v}$ at every point on the interfaces can be approximated by Eq. 1.3. The isotropy hypothesis remains here to consider M as only dependent on the temperature and γ as constant. In this context, the substitution of Eq. 1.3 into the transport equation 1.25 of each level-set function ϕ_i results in:

$$\frac{\partial \phi_i}{\partial t} - M\gamma\kappa\vec{n} \cdot \vec{\nabla}\phi_i = 0. \tag{2.10}$$

Most of the methods described hereafter can be extended to the context of anisotropic grain boundary energy γ or mobility M [143, 144], but this will not be considered here.

Eq. (2.10) is notoriously difficult to properly solve by usual numerical methods. The local curvature κ involves second derivatives of the level-set function ϕ_i, whose numerical estimate is very irregular, despite the large number of methods that have been proposed in the literature to obtain smooth and accurate approximations (see for instance [186, 187, 188]). Moreover, the computation of the curvature is often performed as a post-processing of the level-set function solution and not as an inherent part of the numerical scheme used to discretize Eq. (2.10). The time-explicit nature of this staggered approach implies the use of very small time steps because of restrictive stability conditions.

This is why it is preferred in this work to rewrite Eq. (2.10) by adding an additional assumption: the LS fields ϕ_i remain at all times signed distance functions ($|\vec{\nabla}\phi_i| = 1$) around their zero-isovalues during boundary migration. The resulting diffusive equation for the LS functions reads:

$$\frac{\partial \phi_i}{\partial t} - M\gamma\Delta\phi_i = 0. \tag{2.11}$$

Eq. (2.11) is in general much more stable than Eq. (2.10) and avoids the direct calculation of κ. It is solved by a standard linear Finite Element method in space combined with a backward Euler scheme in time. The implementation is fully parallel and has been shown to be perform efficiently on a large number of processors [154].

Interface treatment and Remeshing

Three different combinations of interface representation and mesh adaptation strategies are compared in the numerical tests performed in this chapter (see Sec. 2.5).

The first approach consists in using a Static Mesh (SM), disabling mesh adaptation so that the mesh remains the same all along the simulation. The interface is then implicitly represented by the level-set functions.

In the second approach, interfaces are still described by level-set functions only, but they are better captured by locally refining the mesh in their vicinity through Isotropic Mesh Adaptation (IMA). The local refinement makes it possible to better resolve the curvature of the grain boundaries.

The third approach consists in applying the New Fitting And Joining Algorithm (NFJA) described in Sec. 2.3 to track interfaces explicitly with a body-fitted mesh. With this technique, a mesh adaptation step is still required in order to improve the low mesh quality resulting from the fitting procedure and to accurately capture the interface curvature through local mesh refinement.

The mesh adaptation procedure involved in the IMA and NFJA approaches relies on local topological mesh operations, that are applied iteratively with the objective of improving a mixed criterion [189]. The criterion combines an evaluation of the local element quality and the conformance of the local edge length to a prescribed size field. Anisotropic meshes could have been generated in the vicinity of the interface through metric-based techniques [175, 190]. However, we have found highly anisotropic meshes to be of little interest in this specific context: as level-set functions are linear in the normal direction to interfaces, the main driver of the accuracy is the mesh size in the tangential direction, that determine the resolution of the interface curvature.

Considering a mesh size h_{int} at the interface, the mesh size field h is defined by:

$$h = \begin{cases} h_{\text{int}} & \text{if } |\phi| \leq 4h_{\text{int}} \\ h_{\text{int}} + \dfrac{7(|\phi| - 4h_{\text{int}})}{4} & \text{if } |\phi| \in [4h_{\text{int}}, 8h_{\text{int}}] \\ 8h_{\text{int}} & \text{if } |\phi| \geq 8h_{\text{int}} \end{cases} \tag{2.12}$$

This definition makes it possible to obtain a band of four refined elements on each side of the interface and a coarser (8 times larger) mesh in the bulk of the phase in a very similar way as in [191].

Additional tools

In this work, with the purpose of high performance, for all the presented cases, we will use the tools mentioned in section 1.3.5, corresponding to the Global Level-Set framework, Coloring/Recoloring techniques of [8, 146] and the Direct (Geometric) reinitialization protocol of [158].

General Algorithm

The general approach can be summarized as follows:

Algorithm 2 Grain growth and body-fitted remeshing algorithm [19]

```
Generate Initial State with a Coloring Method
while Time < FinalTime do
    for all LS Field φ do
        Solve PDEs with a FE method for φ
    for all LS Field φ do
        Apply Eq. 1.24 to φ
    for all LS Field φ do
        for all Grain G in φ do
            Transport G to another φ if necessary (Re-Coloring)
    for all LS Field φ do
        Reinitialize φ
    if Remeshing Active then
        if Body-Fitted Remesh then
            Remesh with the NFJA method
        if Interface Capturing Remesh then
            Remesh with the IMA method
```

2.4.2 Source of errors

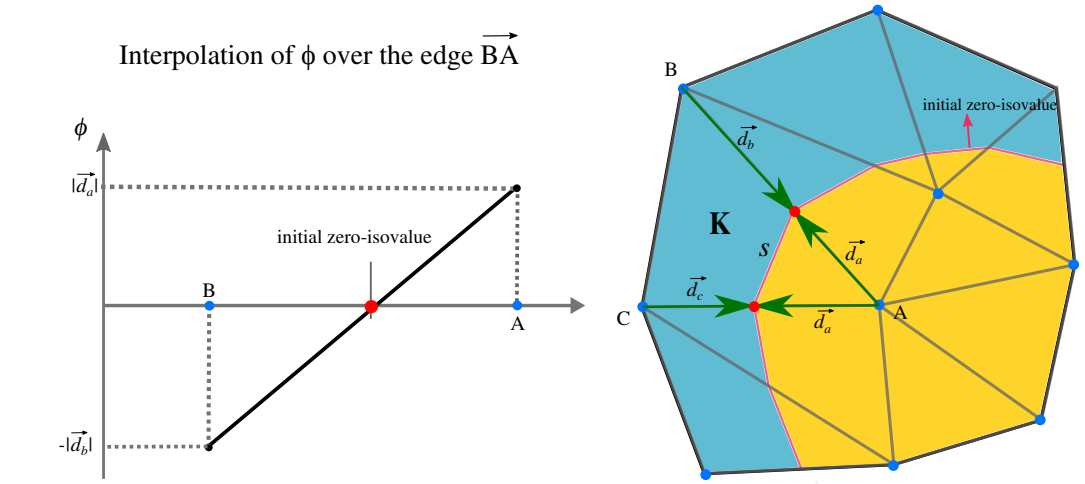
Each one of the numeric models has a different set of sources of errors that have been classified as follows: errors given by the direct reinitialization algorithm, errors given by the remeshing and transport process and finally, errors given by the resolution of the diffusion equation in the considered FE framework (P1, unstructured FE mesh). We will detail each one of the sources of errors excluding the ones obtained by the FE solution of the EDP which are considered as a function of the convergence parameter given to the FE solver and common to the different FE remeshing strategies proposed here.

Direct reinitialization errors

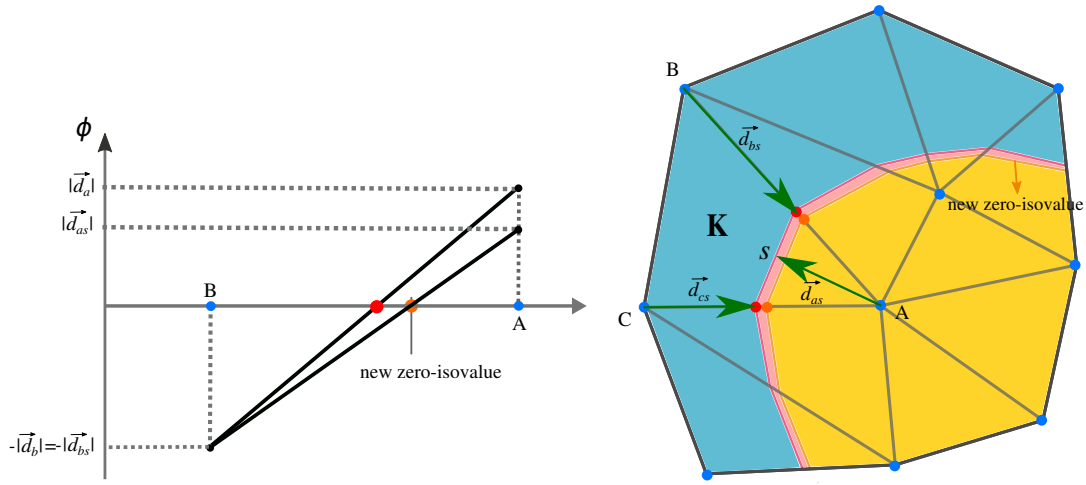
Using the direct reinitialization method proposed in [158] is a very fast way of reinitializing a LS field, however, some errors will be present when the FE discretization uses linear elements. Consider the configuration of figure 2.7(a), in this case, the zero-isovalue of the LS field (ϕ) is obtained by interpolating the LS values within the elements where a change of sign of the LS was found, a sequence of segments defining the interface is identified. Then, the reinitialization algorithm will recompute the distance of each one of the nodes to the nearest part of the nearest segment. Applying this procedure to node A, B and C of figure 2.7(a) will result in a shrinking of the concave phase: the initial LS values are given on A, B and C by the positive norm of vector $\vec{d_a}$ and the negative norm of vectors $\vec{d_b}$ and $\vec{d_c}$ ($|\vec{d_a}|$, -$|\vec{d_b}|$ and -$|\vec{d_c}|$). The zero-isovalue of the interpolation of these LS values within element K will be the segment s used to reinitialize nodes A, B and C. For nodes B and C, the value of the computed distance to segment S is the same as the initial LS values for these nodes ($|\vec{d_{bs}}| = |\vec{d_b}|$ and $|\vec{d_{cs}}| = |\vec{d_c}|$). However, the nearest distance of node A to the segment S is the norm of the vector $\vec{d_{as}}$ (which is orthogonal to S). As $|\vec{d_{as}}| < |\vec{d_a}|$, when interpolating the new reinitialized LS field, its zero-isovalue will be different from the initial one (see figure 2.7(b)). Normally, these errors increase when the curvature of the zero-isovalue increase, and they become null if the zero-isovalue is a straight line. A way to avoid this phenomena, is by using body-fitted interfaces like the ones obtained using the fitting and joining algorithm presented in this paper.

Remeshing and transport errors

It is well known that the process of transporting a numeric field from a mesh to another causes errors, mainly if the interpolation used for the transport has a low order. Figure 2.8 shows one example of loss of surface when a mesh using linear elements transports one LS field to another mesh. Figure 2.8(a) shows the initial mesh with its given zero-isovalue of the LS field. Each node of the new mesh (Fig. 2.8(b)) will compute its LS value from the interpolation of the LS field over the old mesh. Many case scenarios can occur. For instance, if one edge of the new mesh is completely inside of one of the elements of the old mesh (as in Fig. 2.9 see edge $\vec{zy}$), the old isovalues (including the zero-isovalue) over that edge will be transported to it as they intersect the same edge. In other words, the interpolation of the old LS to the nodes of that edge will cause that the isovalues computed in the old element along the edge and the new interpolation over the edge, coincide. However, this is not the case for all the edges of the new mesh, many of them will cross the edges of the old mesh (as in Fig. 2.9 see edge $\vec{xy}$). As the interpolation over each one of the nodes of those edges comes from different elements, the isovalues on those edges will not coincide with the interpolation of the LS of the old mesh along those edges, hence causing errors.



(a)



(b)

Figure 2.7: Reinitialization errors: a) initial configuration, interpolation of the initial LS field to its zero-isovalue, b) new interpolation after reinitializing compared to the initial one. Errors in the computation of the distance to the linear interpolation of the zero-isovalue of the LS produce the shrinkage of the concave phase.

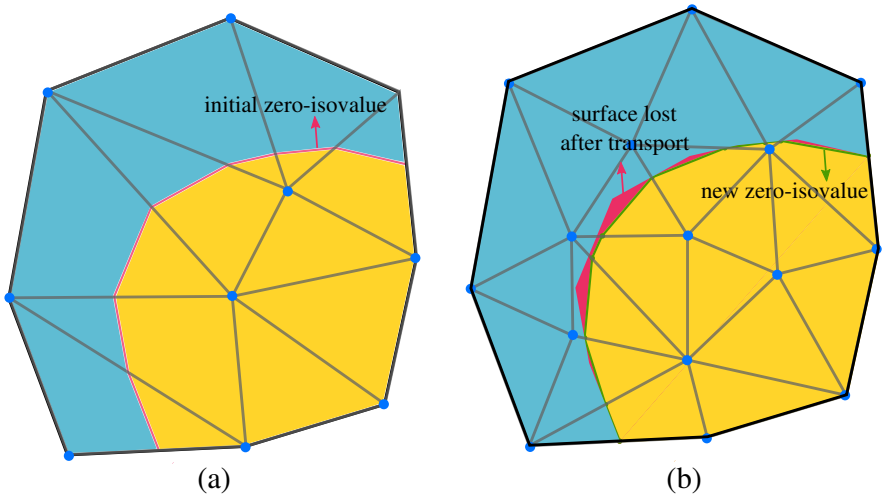


Figure 2.8: Transport Errors: a) Initial configuration, interpolation of the initial LS field to its zero-isovalue over the old mesh. b) New interpolation after remeshing and transporting the LS field to the new mesh, superposition of the phase before and after the remeshing/transport process, a loss of surface is identified.

As for the reinitialization errors, transport errors will cause the concave phase to shrink, producing a loss of surface. Once again, this phenomenon can be avoided if the nodes of the new mesh coincide with the zero-isovalue of the old mesh, which is exactly the way of producing new meshes by using the new fitting and joining algorithm.

Errors of the New Fitting and Joining Algorithm

One of the issues when using the new fitting and joining algorithm is that some of the elements at the interface end up with very poor quality (in terms of shape and size) when the zero-isovalue crosses the element too near to a node (see Fig. 2.10(a)). One way to avoid this problem is by pushing the interface to the node before applying the fitting algorithm to the patch of elements (see Fig. 2.10(b)). This procedure is triggered if the volume of one of the elements after fitting is smaller than a user defined value δ_v (the element colored in red in Fig. 2.10(a) left).

Once the fitting algorithm has finished, a mesh adaptation procedure begins in order to improve the quality of the mesh near to the fitted interface. Small changes on the volume of each phase are allowed under a percent value δ_p also user-defined. Take for instance the example showed in Fig. 2.11. An initial fitting process is done over the initial configuration (Fig. 2.11(top-left)) giving as a result the body-fitted mesh (Fig. 2.11(top-right)) containing potentially ill conditioned elements (see elements attached to nodes A, B and C), then, a first

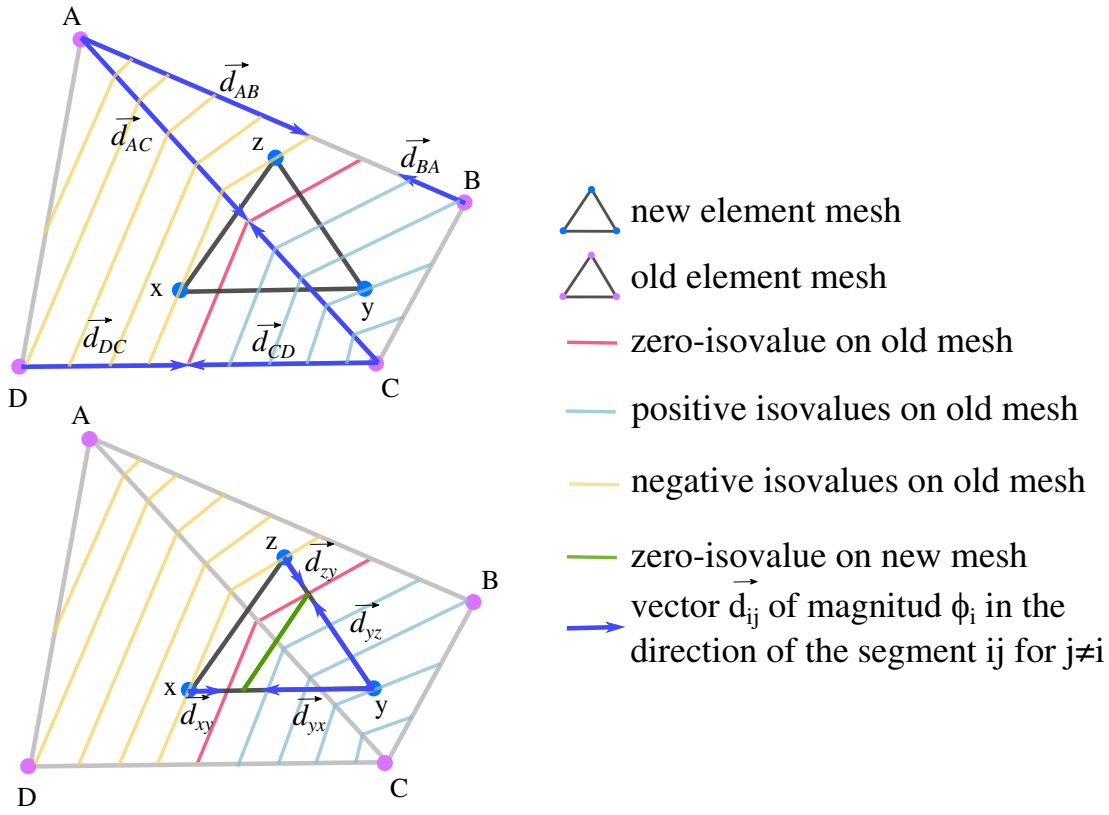


Figure 2.9: Transport example from two elements (ABC and ACD) to one element (xyz), a surface-loss after the transport is found on element xyz as the new linear interpolation crosses the edge $\overline{AC}$.

adaptation process will over node A, which is going to move the same node to a position where the quality of the whole element patch of node A is improved (Fig. 2.11(bottom-right)) till then, no change on either of the phases is registered. Then, a second adaptation attempt is done: the patch of elements surrounding node B in Fig. 2.11(bottom-right) contains two phases. If by applying a remeshing operation to this element patch, the quality is improved and the maximum volume change of the two phases (in this case, the yellow phase) does not go over the allowed user-defined percent (percent of the initial volume of the phase on the element patch) the operation is registered, else, the operation is discarded and no change on the patch is made. An additional constraint is added to the nodes of the boundary of the FE domain: when performing the adaptation step, the nodes of the boundary are not allowed to move, even if the movement increases the quality of the local patch. This condition enables to maintain the calculation domain boundaries during interfaces migration. Finally, point deletions are allowed in the same way as for all the other nodes (if the changes do not reduce or increase the volume of the local patch over the allowed user-defined percent).

These processes operations cause errors on the surface of the phases, producing

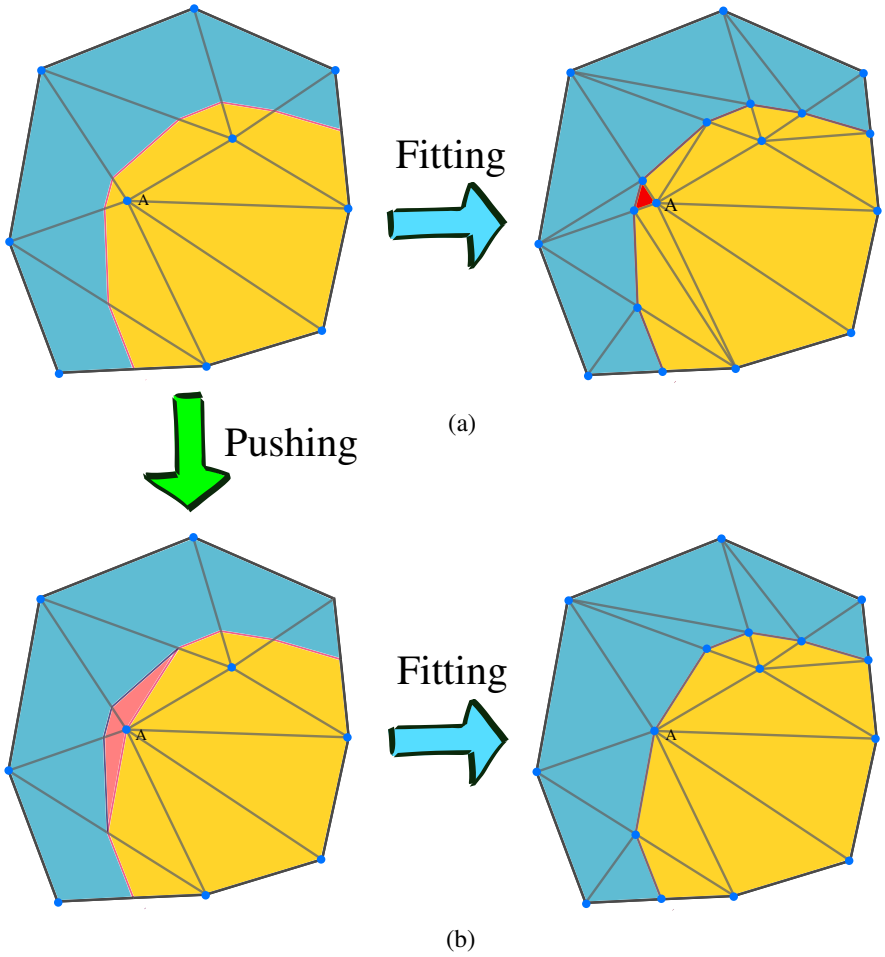


Figure 2.10: Fitting Relaxation: a) left: Initial configuration, right: fitting without relaxation, elements with a very small volume $< \delta_v$ are colored red. b) left: relaxation of the interfaces by pushing the boundaries to node A, the surface-loss of the yellow phase is colored pink, right: fitting after relaxation.

some of them to expand or to shrink. We will measure the effect of the relaxation with the help of the following test cases. The other sources of errors explained before will be studied and compared too.

2.5 Numerical results

2.5.1 Considered geometries

Equation 2.11 will be considered to obtain the evolution of the LS fields for several test cases, each one responding to different interesting topological situations: sphere shrinkage, T-Junction, square shrinkage and finally, a 2D Laguerre-Voronoi tessellation composed of 10000 grains. For each one of these geometrical configurations, the three remeshing approaches (SM, IMA, NFJA) will be com-

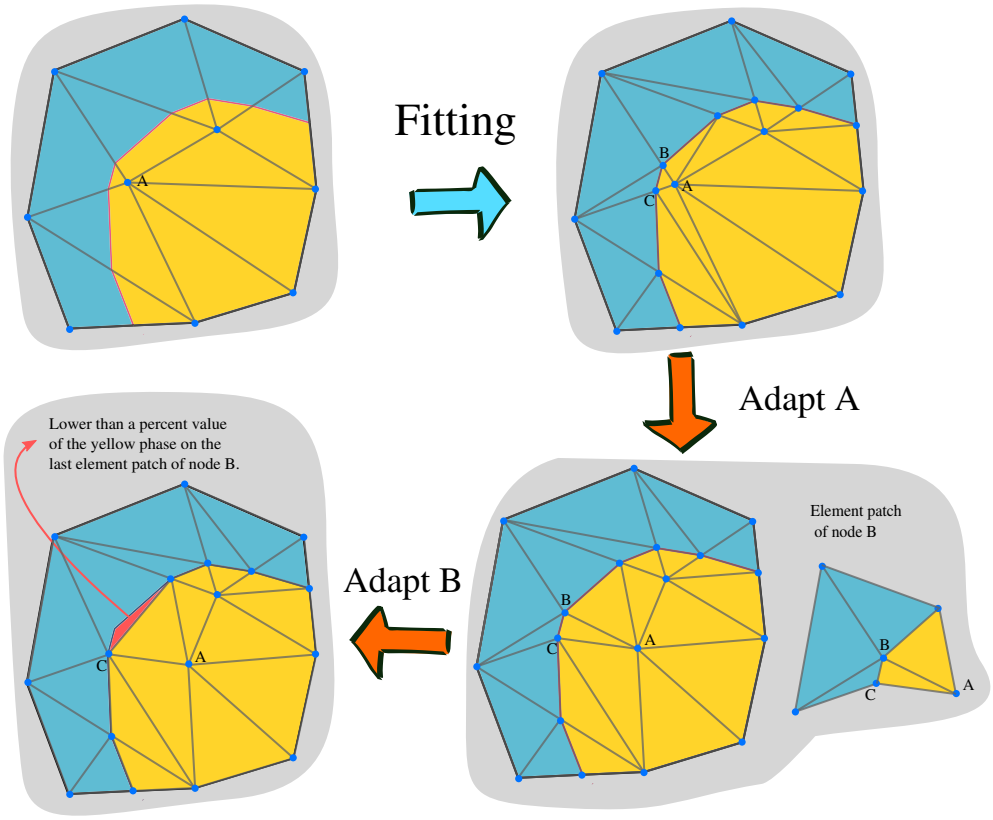


Figure 2.11: Adaptation Relaxation: (top-left): Initial Configuration, (top-right): Fitted Mesh, (bottom-right): adapted node A (vertex displacement), (bottom-left): adapted node B (edge $\overline{BC}$ collapsed on C)

pared.

2.5.2 Circle shrinkage

The circle shrinkage test case, being the most basic of all, enables to observe the response of the interface to the instantaneous curvature and so to observe the topological disappearance of the grain. This case allows to compare the shape evolution of the phase defined by the positive values of ϕ to the analytic solution; The evolution of the analytical circle radius r is simply given by the following differential equation:

$$\frac{dr}{dt} + M\gamma\frac{1}{r} = 0 \Longrightarrow r\left(t\right) = \sqrt{r_0^2 - 2M\gamma t}. \tag{2.13}$$

This equation can be rewritten in terms of the surface of the circle S, with a linear solution, as:

$$\frac{dS}{dt} + 2\pi M\gamma = 0 \Longrightarrow S\left(t\right) = S_0 - 2\pi M\gamma t. \tag{2.14}$$

As mentioned before, solving Eq. 2.11 in a FEM context implies some errors. Here we will quantify each source of errors using the surface S_ϕ of the phase ϕ describing the "circle". In reality, the phase ϕ can not define a perfect circle but the errors given by its real shape will be neglected.

We can establish the signed difference between the numeric surface computation S_ϕ and the analytic value S as the total error E^{**}:

$$E^{**}_{(\phi,h,t,\Delta t)} = S_{\phi(\phi,h,t,\Delta t)} - S_{(t)}, \tag{2.15}$$

where h is the mesh size at the interfaces and Δt is the time step. Eq 2.15 can be rewritten in order to obtain the error per increment E^{*}:

$$E^{**}_{(\phi,h,t,\Delta t)} - E^{**}_{(\phi,h,t-\Delta t,\Delta t)} = S_{\phi(\phi,h,t,\Delta t)} - S_{\phi(\phi,h,t-\Delta t,\Delta t)} + S_{(t-\Delta t)} - S_{(t)}, \tag{2.16}$$

$$E^{*}_{(\phi,h,t,\Delta t)} = \Delta S_{\phi(\phi,h,t,\Delta t)} - \Delta S_{(t,\Delta t)}. \tag{2.17}$$

Finally, we can express the relative error as:

$$E_{(\phi,h,t,\Delta t)} = \frac{\Delta S_{\phi(\phi,h,t,\Delta t)} - \Delta S_{(t,\Delta t)}}{\Delta S_{(t,\Delta t)}}. \tag{2.18}$$

The term $\Delta S_{\phi(\phi,h,t,\Delta t)}$ is actually the addition of the different contributions to the change of surface given by the different treatments done during a computational increment: solution of the EDP in the FE context, reinitialization and remeshing. The latter itself is also differentiated in the change of surface given by the processes of transport, fitting and adaption as explained in section 2.4.2. This can be summarized as follows:

$$\Delta S_{\phi(\phi,h,t,\Delta t)} = \Delta S_{solver(\phi,h,t,\Delta t)} + \Delta S_{reinit(\phi,h,t,\Delta t)} + \Delta S_{remesh(\phi,h,t,\Delta t)}, \tag{2.19}$$

$$\therefore E_{(\phi,h,t,\Delta t)} = \frac{\Delta S_{solver(\phi,h,t,\Delta t)} - \Delta S_{(t,\Delta t)}}{\Delta S_{(t,\Delta t)}} + \frac{\Delta S_{reinit(\phi,h,t,\Delta t)}}{\Delta S_{(t,\Delta t)}} + \frac{\Delta S_{remesh(\phi,h,t,\Delta t)}}{\Delta S_{(t,\Delta t)}}. \tag{2.20}$$

Finally, Expressions for the three principal sources of error can be established:

$$\xi_{solver(\phi,h,t,\Delta t)} = 100 \cdot \frac{\Delta S_{solver(\phi,h,t,\Delta t)} - \Delta S_{(t,\Delta t)}}{\Delta S_{(t,\Delta t)}}, \tag{2.21}$$

$$\xi_{reinit(\phi,h,t,\Delta t)} = 100 \cdot \frac{\Delta S_{reinit(\phi,h,t,\Delta t)}}{\Delta S_{(t,\Delta t)}}, \tag{2.22}$$

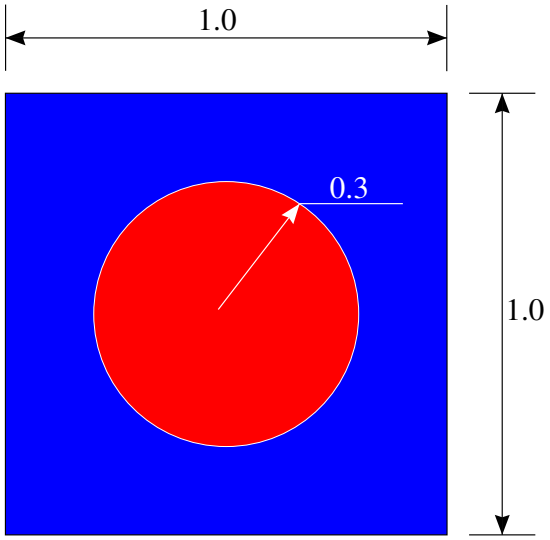


Figure 2.12: Circle shrinkage test case: initial state.

$$\xi_{remesh(\phi,h,t,\Delta t)} = 100 \cdot \frac{\Delta S_{remesh(\phi,h,t,\Delta t)}}{\Delta S_{(t,\Delta t)}}, \tag{2.23}$$

where the terms ξ_i represent the relative error given by the procedure i (solver, reinit and remesh). The dependence of ϕ for the different values of ξ_i describes the error produced by how far the phase ϕ is to represent a perfect circle of surface S_ϕ, in this study, this dependence will be neglected and we will focus on the evolution of the error as a parameter of the mesh size h, the time t and the time step Δt. Note that we have not used an absolute value to describe each error. Indeed, we want to observe if the numeric models are quicker or slower than the analytic solution: a negative (resp. positive) value of the error would mean that phase ϕ shrinks too fast (resp. low) during the considered step.

In the following, dimensionless simulations will be considered and the value of the reduced mobility $M \cdot \gamma$ will be assumed to be unitary.
The initial radius is set as $r_0 = 0.3$ (initial surface $S_0 \approx 0.2827$) and the circle is immersed in the center of a 1×1 square as illustrated in Fig. 2.12.

As stated before, three meshing configurations will be compared, the first one will use the interface fitting and joining algorithm to remesh every time step, the second one will use an interface capture meshing (a mesh refined only at the interfaces but not fitted) and the last one will use a static mesh with a uniform mesh size. Examples of each meshing approach are given in Fig. 2.13. Note that for the static mesh, all the domain must be refined in order to maintain the same level of accuracy as for the other cases during the interface migration.

Multiple runs with different mesh sizes h and time steps dt were made for

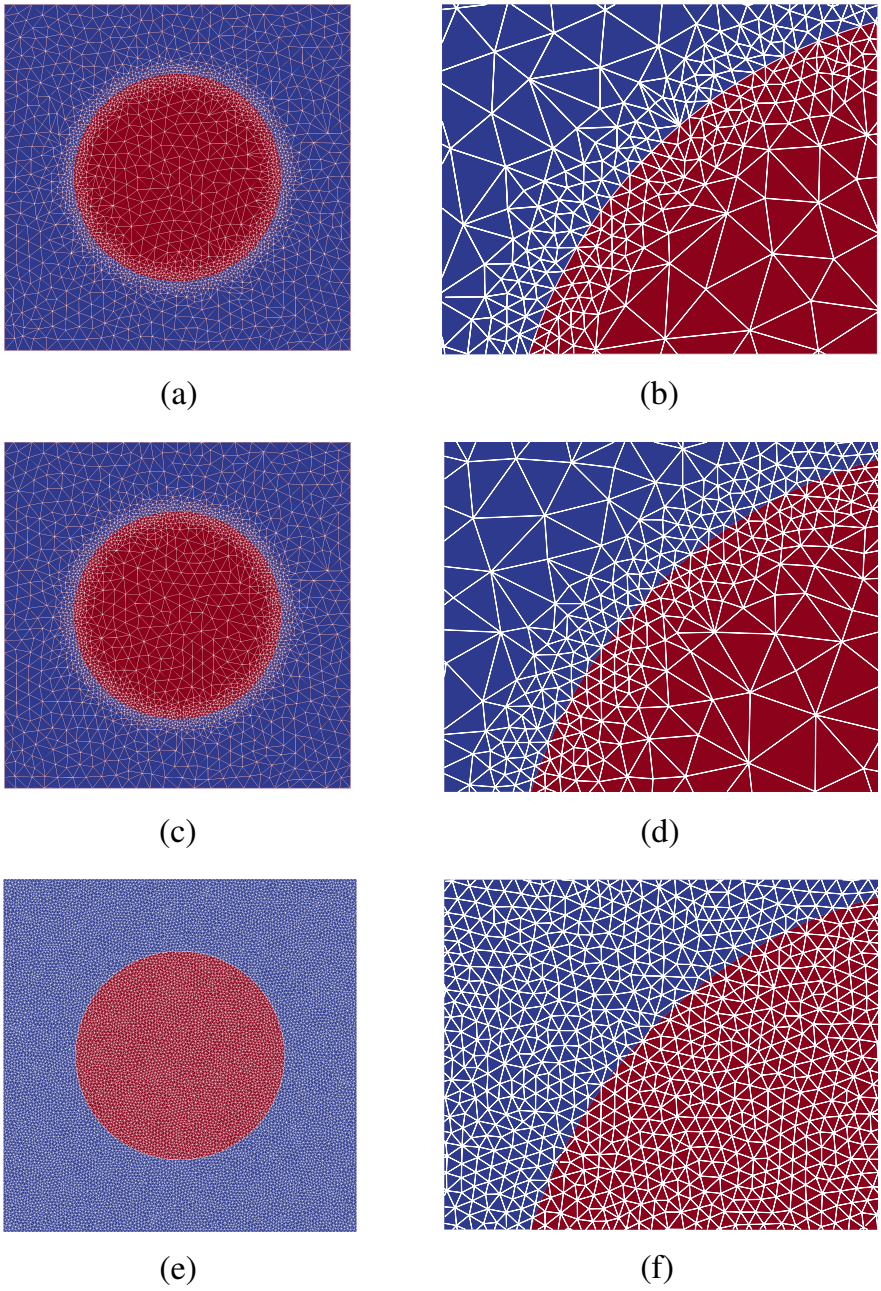


Figure 2.13: Meshes obtained for each one of the configurations for the circle shrinkage test at $t = 0$ and with a mesh size $h = 0.01$ at the interface. (a,b) Using the interface fitting and joining algorithm. (c,d) Using a classic interface capturing algorithm. (e,f) Static mesh.

each one of the configurations. Eq. 2.11 was solved with a standard diffusion FE solver with a precision of 10^{-10} for a P1 (linear) interpolation.

One example of the obtained results for the evolution of the surface for the configuration with a static mesh is summarized in Fig. 2.14, Ks and Kr are the median values of the surface change when solving the FEM problem and when reinitializing respectively. These values can be replaced in Equations 2.21 and 2.22 as ΔS_{solver} and ΔS_{reinit} to obtain the values of ξ_{solver} and ξ_{reinit}. Figure 2.15 shows the values of ξ_{solver} and ξ_{reinit} for different time steps and mesh sizes. Note that for this case $\xi_{transport}$, ξ_{fit} and ξ_{adapt} are equal to zero because there is no remeshing.

It is interesting to see that errors given by the reinitialization procedure are much more important than those obtained by using a FEM to solve Eq. 2.11 and that they tend to be bigger when the time step decreases. In fact, from our observations ΔS_{reinit} is not dependent of the time step dt used (it depends only on the mesh size h), however, a smaller dt means more increments to simulate the same time, and as the error from the reinit accumulates, the more increments the bigger the value of ξ_{reinit}.

A similar computation can be made for the IMA and the NFJA approaches. When using the IMA approach, the value $\xi_{transport} \neq 0$ because the remeshing makes the procedure of transport unavoidable. however, as there are no fitting, $\xi_{fit} = 0$ and $\xi_{adapt} = 0$. The values of ξ_{solver} and ξ_{reinit} are almost equal to those of the SM method displayed in Figures 2.15 as the only difference near to the interface between these two approaches is that a procedure of remeshing is done between time steps. Figure 2.16 shows the results for the values ξ_{solver}, ξ_{reinit} and $\xi_{transport}$ for different values of the mesh size h and the time step dt in context of the IMA strategy. The values of $\xi_{transport}$ were found to be very small compared to those of ξ_{reinit} and ξ_{solver} hence, they can be neglected in the following.

Finally, when using the NFJA approach, as explained in section 2.4.2, $\xi_{transport} = 0$ and $\xi_{reinit} = 0$. however, $\xi_{fit} \neq 0$ and $\xi_{adapt} \neq 0$ as a fitting procedure is used. The NFJA allows controlling the two variables δ_v and δ_p introduced in section 2.4.2, these variables control directly the values of ξ_{fit} and ξ_{adapt}. A very low value of δ_v will cause that the algorithm allows to get very small elements after fitting while the value ξ_{fit} tends to zero. In the same way, a very low value of δ_p will cause that even though a very poor quality of elements was found after fitting, the algorithm could not modify the interface in order to improve that quality, while the value ξ_{adapt} tends also to zero.

Figure 2.17 shows the values of ξ_{solver} and ξ_{adapt} for one example with $\delta_v = 10^{-10}$ and $\delta_p = 2 \cdot 10^{-2}$. For these values, ξ_{fit} is very low and can be neglected. An example of a mesh obtained with these values is showed too. In the same way,

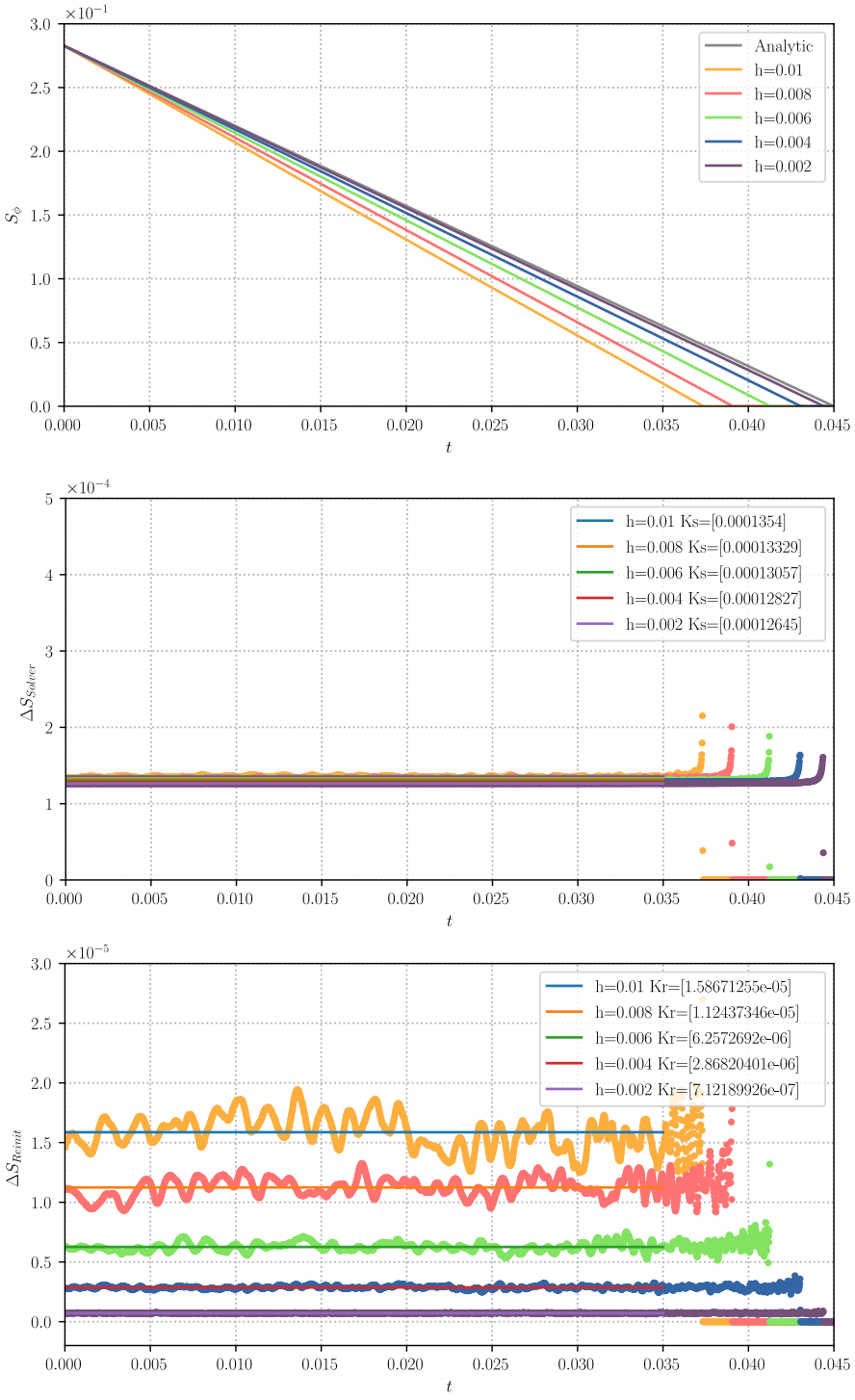


Figure 2.14: Results for the static mesh configuration using a $dt = 2 \cdot 10^{-5}$ and for different mesh sizes h. Top: Evolution of S_ϕ . Middle: evolution of ΔS_{solver}, Ks is the median value for each h. Bottom: evolution of ΔS_{reinit}, Kr is the median value for each h.

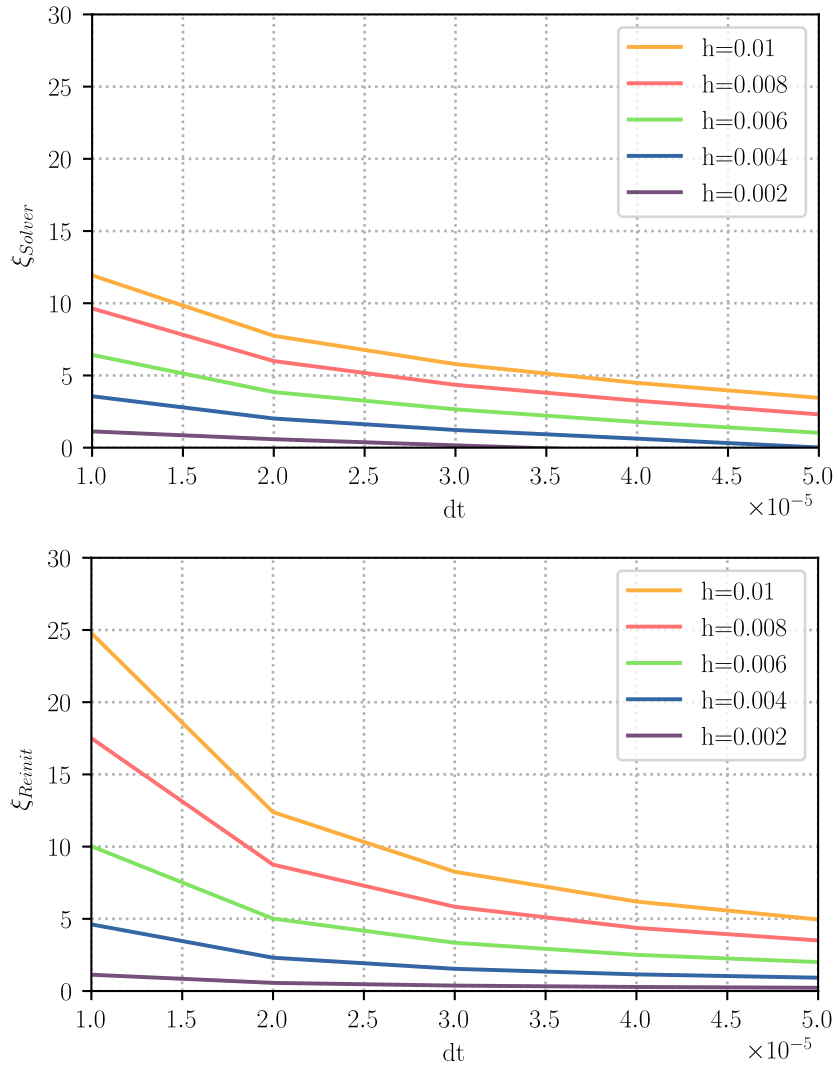


Figure 2.15: Results for the static mesh configuration. values of ξ_{solver} and ξ_{reinit} for different values of the time step and the mesh size.

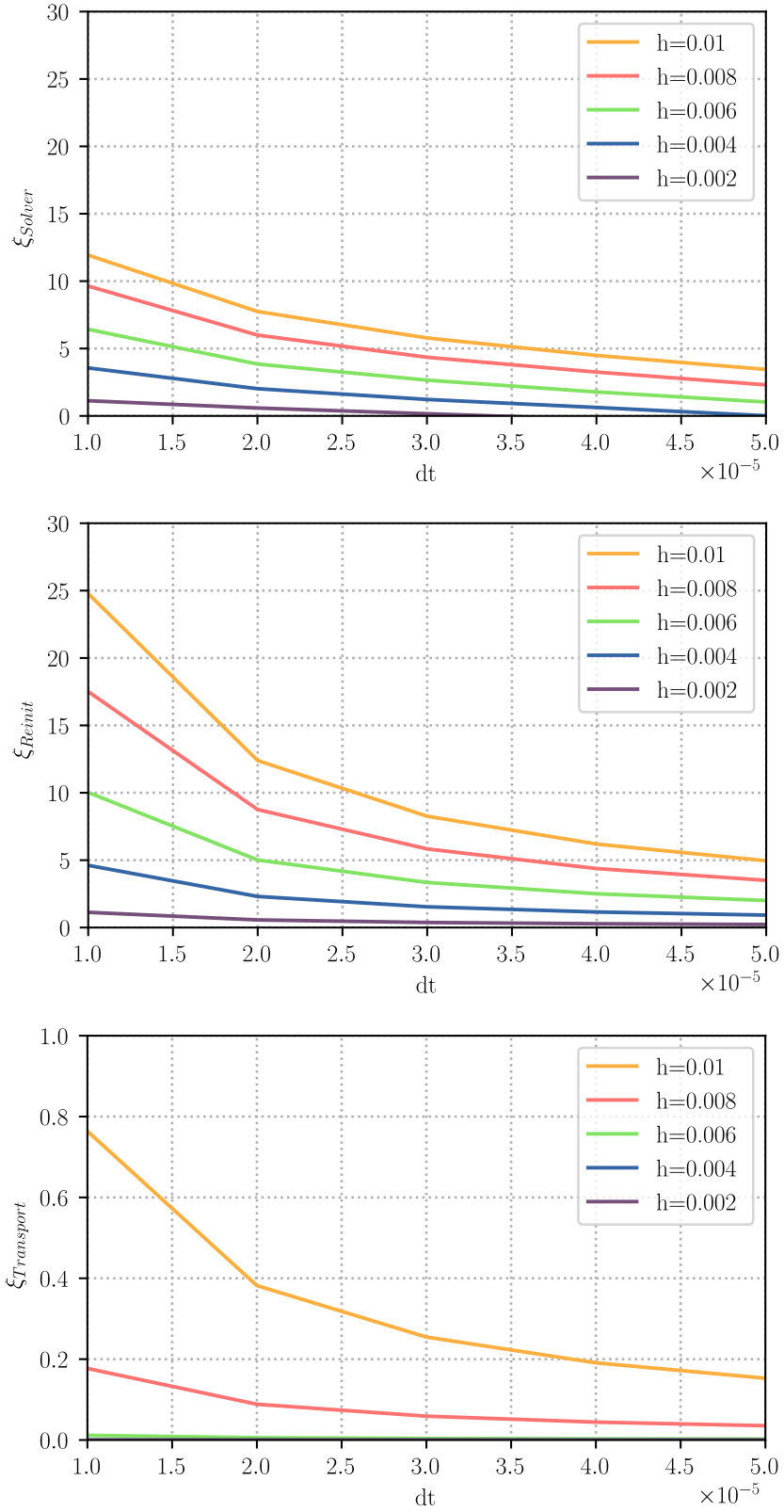


Figure 2.16: ξ_{solver}, ξ_{reinit} and $\xi_{transport}$ for different mesh sizes h and time steps dt for the IMA case

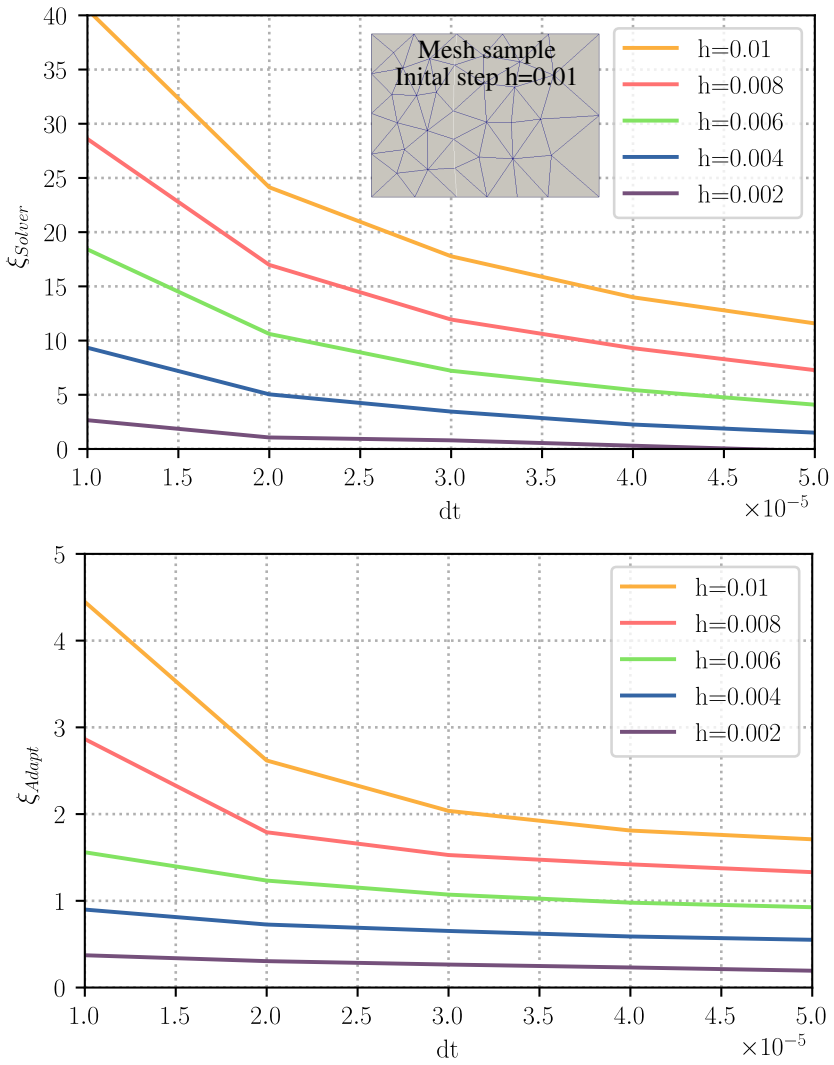


Figure 2.17: ξ_{solver} and ξ_{adapt} for different mesh sizes h and time steps dt for the NFJA case with $\delta_v = 10^{-10}$ and $\delta_p = 2 \cdot 10^{-2}$

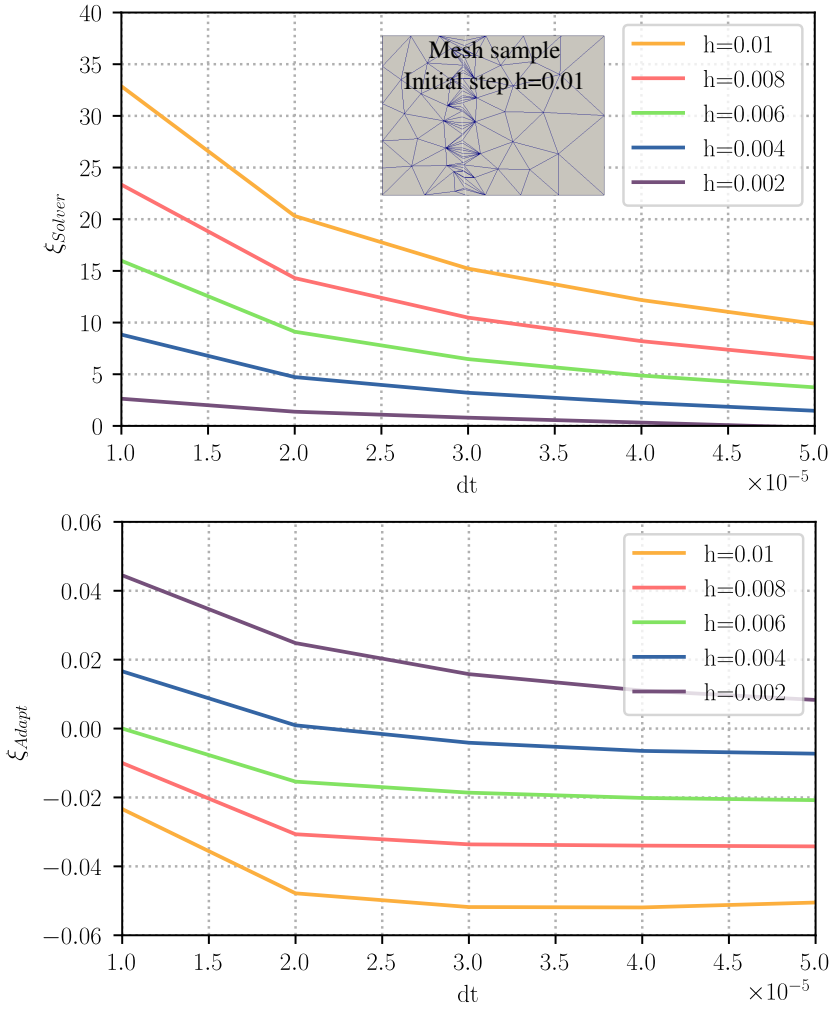


Figure 2.18: ξ_{solver} and ξ_{adapt} for different mesh sizes h and time steps dt for the NFJA case with $\delta_v = 10^{-10}$ and $\delta_p = 2 \cdot 10^{-3}$

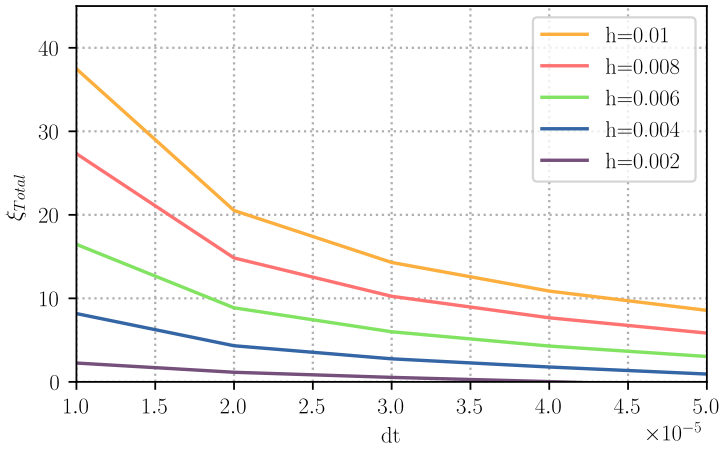
Figure 2.18 shows another example for the values of $\delta_v = 10^{-10}$ and $\delta_p = 2 \cdot 10^{-3}$. It is interesting to see that for the latter, ξ_{adapt} can be neglected and the remaining errors are given by the value of ξ_{solver} and additionally, ξ_{solver} is lower than the one from figure 2.17. Of course, the curvature of the interface is better preserved when δ_p is low but the mesh obtained with such values is nearly degenerated. Results show that the FEM solver is more sensitive to the small changes in the description of the surface than to the quality of the elements, and that it will be actually more accurate to maintain the interface as it is after fitting than to try to improve the mesh quality field with remeshing operations.

Figure 2.19 shows the value of $\xi_{total} = \sum \xi_i$ for the IMA method and for the NFJA with the two sets of constants used in Figures 2.17 and 2.18. The smallest ξ_{total} was found for the NFJA approach with the values $\delta_v = 10^{-10}$ and $\delta_p = 2 \cdot 10^{-3}$. Even though the ξ_{solver} was very high for the NFJA in comparison to the one given by the IMA, results show that with the right choice for δ_v and δ_p for the NFJA approach, one can be more accurate on the prediction of the evolution of interfaces when using Eq. 2.11 and a body-fitted mesh in a FEM context.

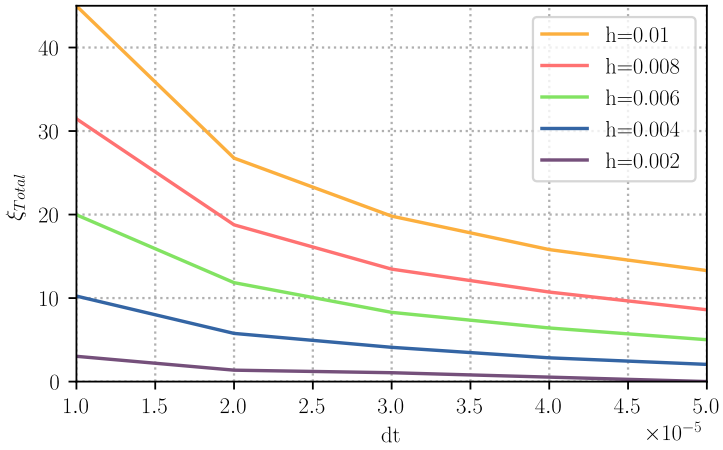
2.5.3 T-Junction case

The presence of triple points is needed in the description of 2D polycrystals, they represent the junction between 3 grains presenting different properties. The T-junction problem is an initially unstable configuration of three interfaces (for three grains at a 90°-90°-180° initial configuration) that converge to a 120°-120°-120° quasi steady-state equilibrium as the $M\gamma$ term is assumed here isotropic. The equilibrium of the triple point is given by the Herring's equation [192]. It will bring the system to a state where the surface energy is minimized and the three lines will arrange themselves in a stable 120°-120°-120° (Young's Equilibrium) configuration evolving after with a homogeneous velocity given by the migration of the curved interfaces and the Neumann boundary conditions [129] (see figure 2.20). Curved Interfaces around the junction will maintain its shape and make the triple point move at a constant velocity until they collapse with another multiple point within the microstructure.

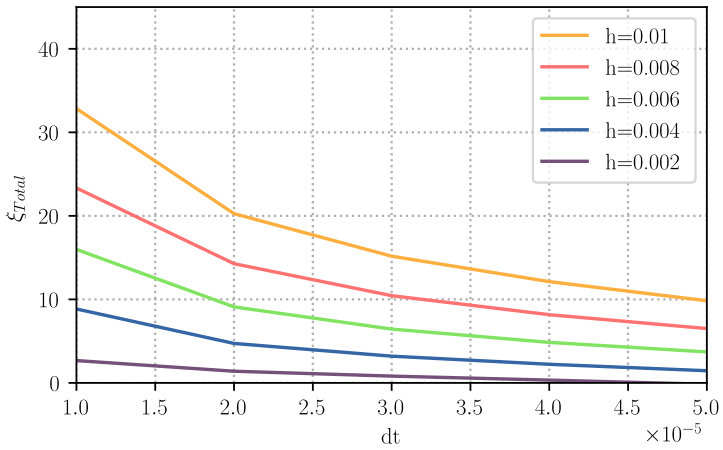
Comparing the results from the T-junction test and the square-shrinkage is much more difficult because there is no analytic solution for these problems concerning the way to reach the quasi steady-state. However, as illustrated in the precedent study over the circle-shrinkage test, convergence is obtained when the mesh size decreased. This result will be used to obtain a reference case for the T-junction problem. Several computations were turned with different static meshes. Figure 2.21 shows the interfaces $\phi_1 - \phi_2$ and $\phi_1 - \phi_3$ after t=0.35 for the T-Junction configuration. At this time, the quasi steady-state is ensured. As convergence is obtained when the mesh size decreases, the simulation with 500000 elements



(a)



(b)



(c)

Figure 2.19: $\xi_{total} = \sum \xi_i$ for different mesh sizes h and time steps dt (a) ξ_{total} for the IMA method. (b) ξ_{total} for the NFJA with $\delta_v = 10^{-10}$ and $\delta_p = 2 \cdot 10^{-2}$. (c) ξ_{total} for the NFJA with $\delta_v = 10^{-10}$ and $\delta_p = 2 \cdot 10^{-3}$.

(last one from figure 2.21) will be used as reference.

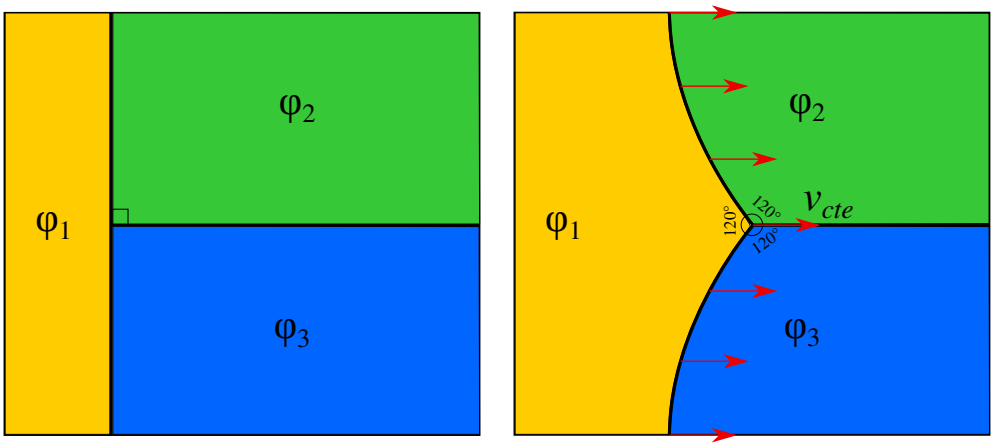


Figure 2.20: T-Junction Case. left: initial state, right: steady state.

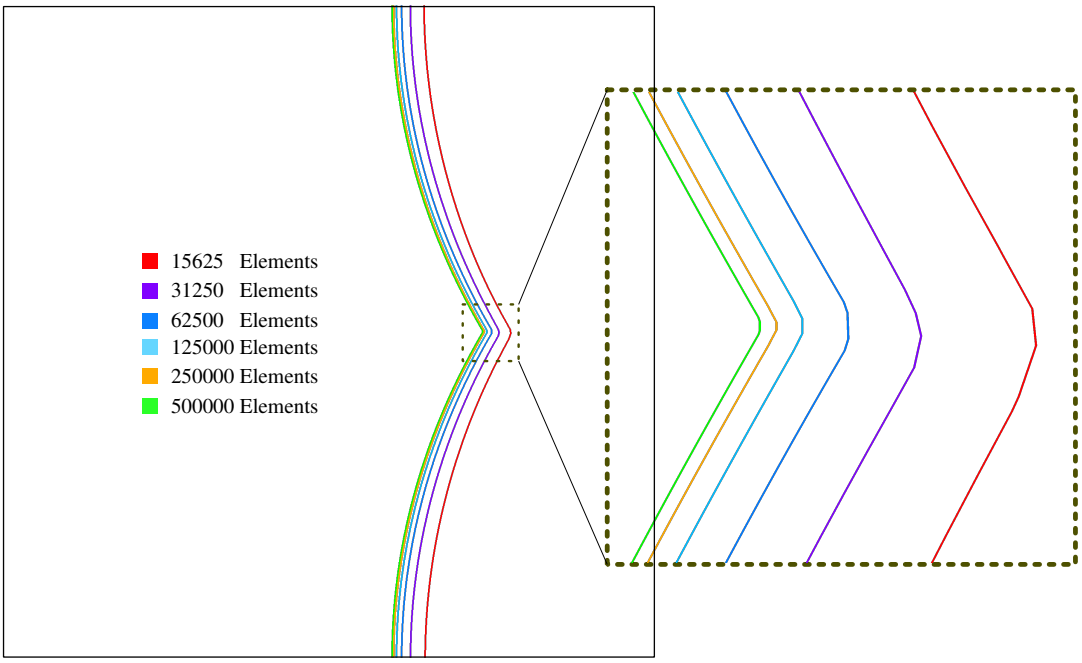


Figure 2.21: T-Junction Case: static Mesh, convergence analysis. Only the interfaces between $\phi_1 - \phi_2$ and $\phi_1 - \phi_3$ are shown (see figure 2.20 for the notation).

Figure 2.22 shows the ϕ_1 phase for the reference model and the geometric difference of the same phase obtained with the other models (the IMA and NFJA) at time t= 0.35. Compared to the reference model, the error in the area of the ϕ_1 phase is $\xi_{IMA} = 10.1\%$ and $\xi_{NFJA} = 15.4\%$. Hence, the IMA approach is clearly more accurate. An acceleration of the multiple point was found for the NFJA

model.

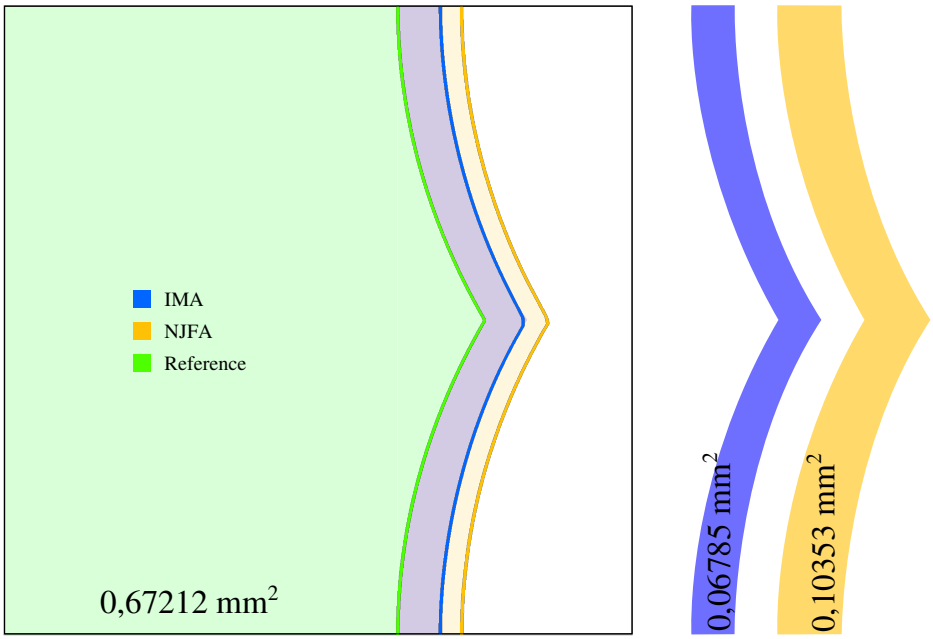


Figure 2.22: ϕ_1 phase for the reference model and the geometric difference of the same phase obtained with the other models (the IMA and NFJA) at time t= 0.35. values for the area of each section is given.

Finally, during the simulation campaign, some differences on the capabilities of the algorithms were observed. Some limits were observed when using the IMA method: the minimum number of refined elements at each side of the interface is around 4. This number of elements ensures that the numeric diffusion obtained at the remeshing step over the metric field is not too important, hence the remeshing success. Otherwise, as illustrated in Fig. 2.23 the remeshing may fail after some increments. On the other hand, the NFJA method is able to remesh successfully every time step, as some of the new nodes are fitted to the interface hence the metric field can not be numerically diffused there.

2.5.4 Square-Shrinkage case

The square shrinkage test makes it possible to observe the behavior of each model when four triple points converge to the same position. The instability of this configuration suggests that the 4 triple points should become 2 triple points and not 1 quadruple point. Figure 2.24 shows the normal behavior of this test case: initially there are 5 phases where the central phase ϕ_1 represents a perfect square; then, each one of the triple points reaches a quasi steady-state similar to the one obtained for the T-Junction problem. At this stage, each one of the interfaces of

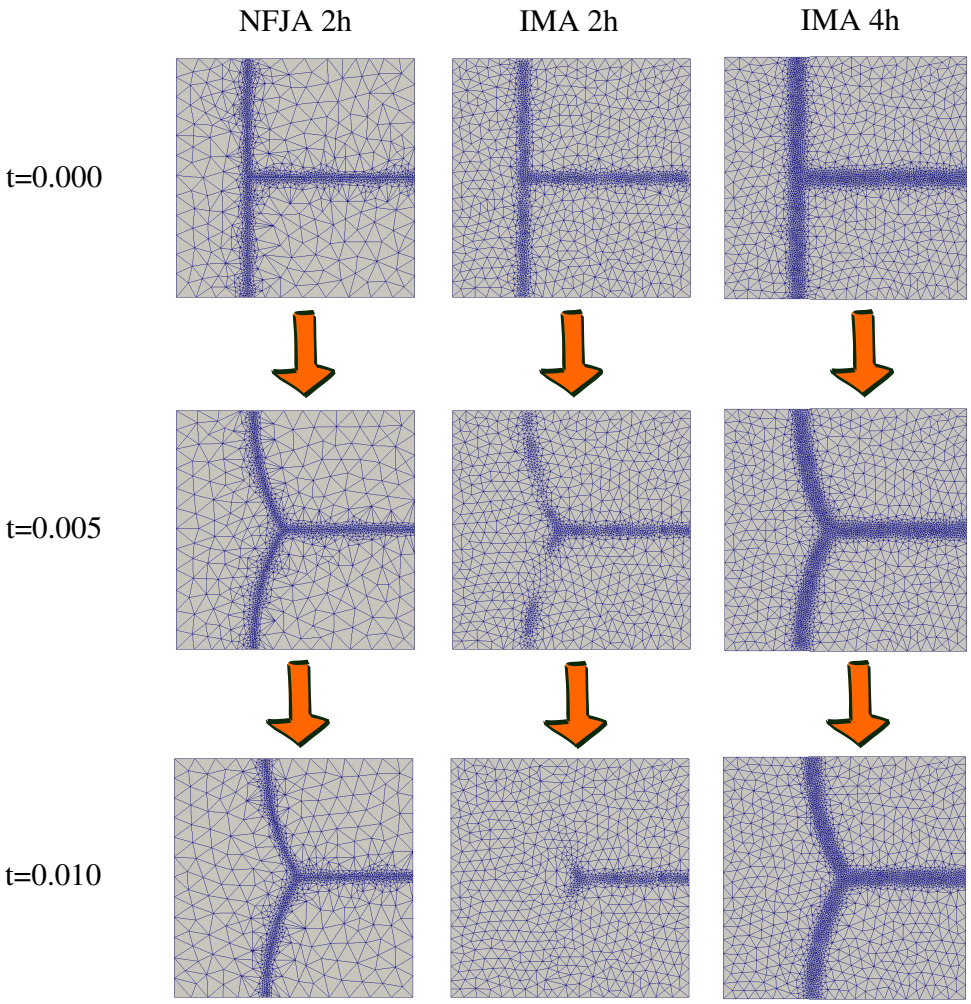


Figure 2.23: T-Junction Case. Numeric diffusion of the metric field used by the mesh adaptation algorithm for the IMA case. The needed thickness of refined elements of each side of the interface is around 4 (4 elements) for the IMA model and around 2 for the NFJA.

the phase represented by ϕ_1 is shrinking at a constant rate. Finally, ϕ_1 disappears and a new interface is created given the absolute numerical instability of the quadruple point. Here two possible configurations are possible for the creation of the new interface ($\phi_2 - \phi_4$ or $\phi_3 - \phi_5$). The choice of the created interface should be given by the difference in the surface energy of each one of the possibilities, the one with the lowest energy is the one that should be created, however, in our study isotropic grain boundary properties are considered, and either of the two decompositions is valid.

In the context of a FE level-set method, it is almost impossible to obtain a perfectly symmetric quadruple junction that will maintain its stability: a symmetric (four right angles) quadruple junction is a meta-stable state that will

decompose in a lower energy state, i.e. two triple junctions, with an infinitesimal perturbation δ. As mentioned in section 2.4, the LS method depends on the FE mesh discretization, convergence stop criterion of the FE solver, the interpolation degree and the approximations made such as the application of Eq. (1) or the reinitializing method used. Finally, by applying this method to the considered geometry (square shrinkage), the angles between the 4 interfaces at the moment when the quadruple point appears will be of $90 + \Theta$ where $\Theta >> \delta$, triggering its decomposition. In fact, It is actually highly probable that the quadruple point never really appears (two very close triple points appearing instead).

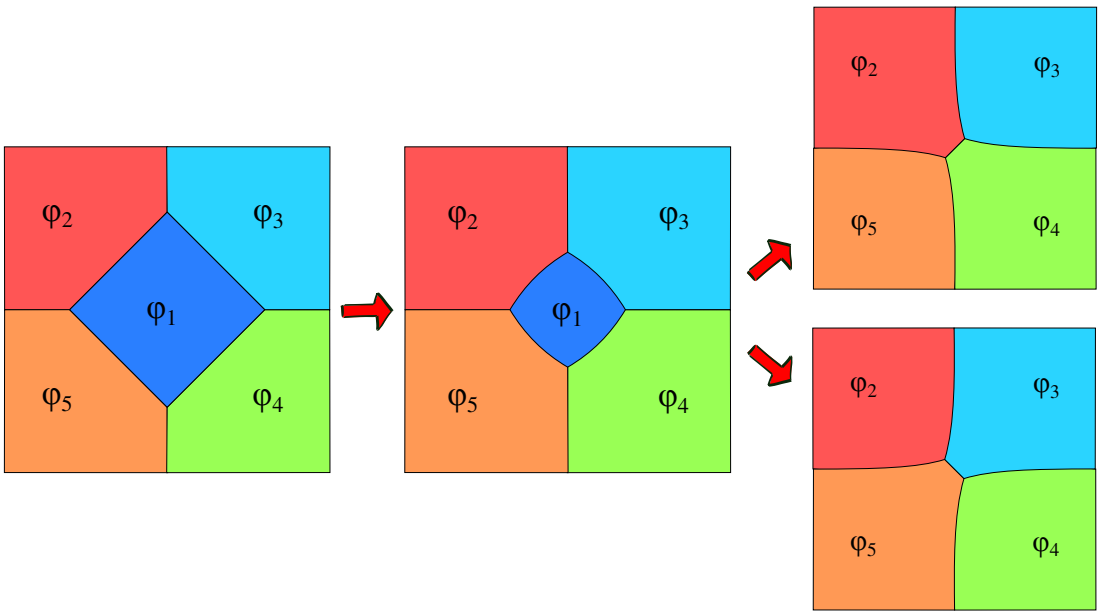


Figure 2.24: Square-Shrinkage. left: initial state, center: square shrinking, right: the square disappears and a new interface is made ($\phi_2 - \phi_4$ or $\phi_3 - \phi_5$).

Similarly to the T-Junction test case, a convergence analysis was made using static meshes in order to obtain the reference evolution for this configuration. Convergence was obtained after using a static mesh with 1 million elements. our reference will employ a static mesh with 2 million elements.

Figure 2.25 shows the comparison of the NFJA and IMA methods to the reference after t=0.05. The error in the area of the ϕ_1 phase was $Err_{NFJA} = 38.4\%$ and $Err_{IMA} = 16.3\%$. Once again an acceleration on the evolution of the NFJA was observed. Another difference from the reference model was that both NFJA and IMA methods created interface $\phi_3 - \phi_5$ while the reference model created $\phi_2 - \phi_4$ (See Figure 2.26).

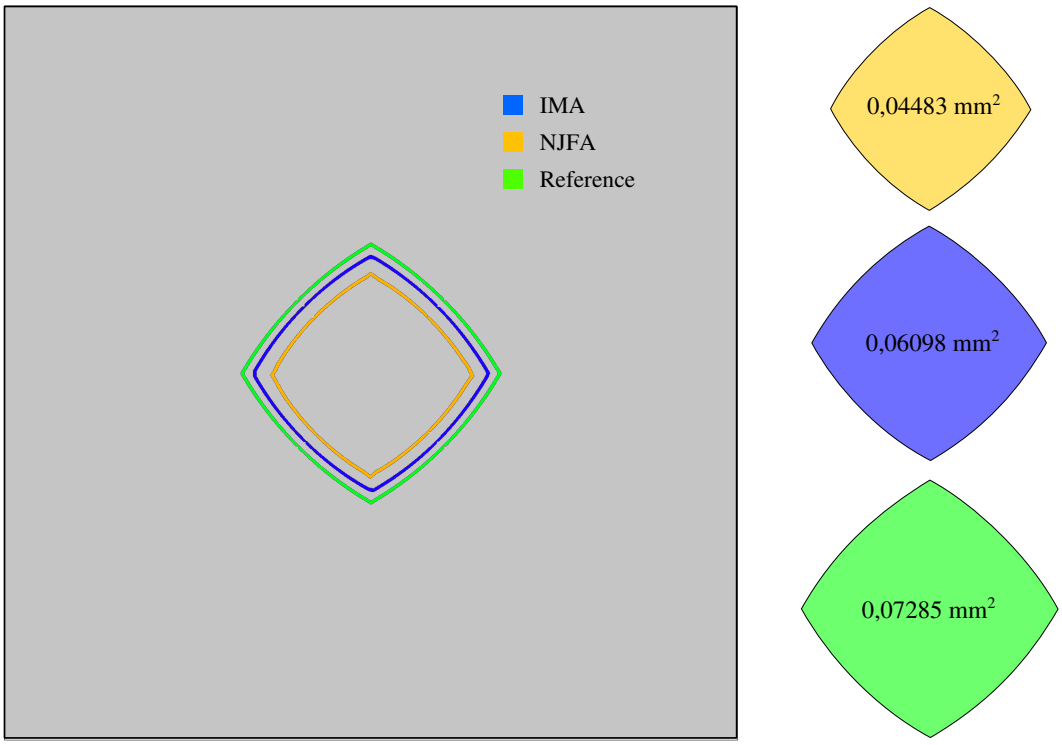


Figure 2.25: Square-Shrinkage. Comparison of the ϕ_1 phase at t=0.05 [s].

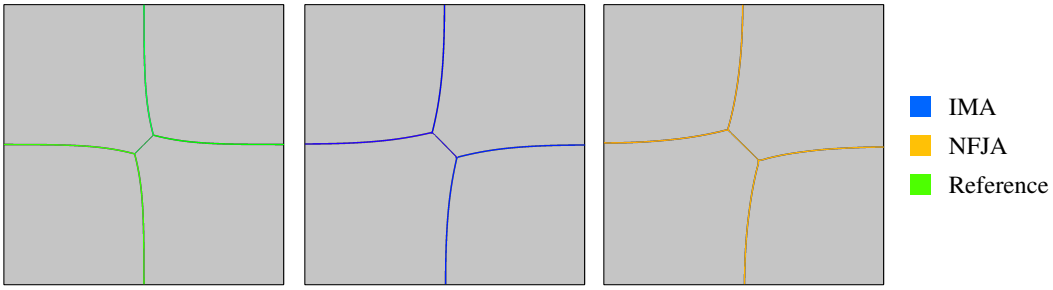


Figure 2.26: Square-Shrinkage. Comparison of the interfaces after the disappearance of the ϕ_1 phase. t=0.1.

2.5.5 2D-10000 grains case

Here, the New Fitting and Joining Algorithm is also compared with a more classic method of mesh adaptation where the interfaces are captured with a non-conform local refined mesh as detailed in section 2.4.1 and described as the Isotropic Mesh Adaptation (IMA) technique instead of tracked with a body-fitted mesh adaptation algorithm. Both cases will be compared to a reference case, which is the convergence of the evolution of the mean grain size (equivalent radius in number) when using a homogeneous refined static mesh. When a homogeneous static mesh is considered transport errors are not present. In addition, if the mesh

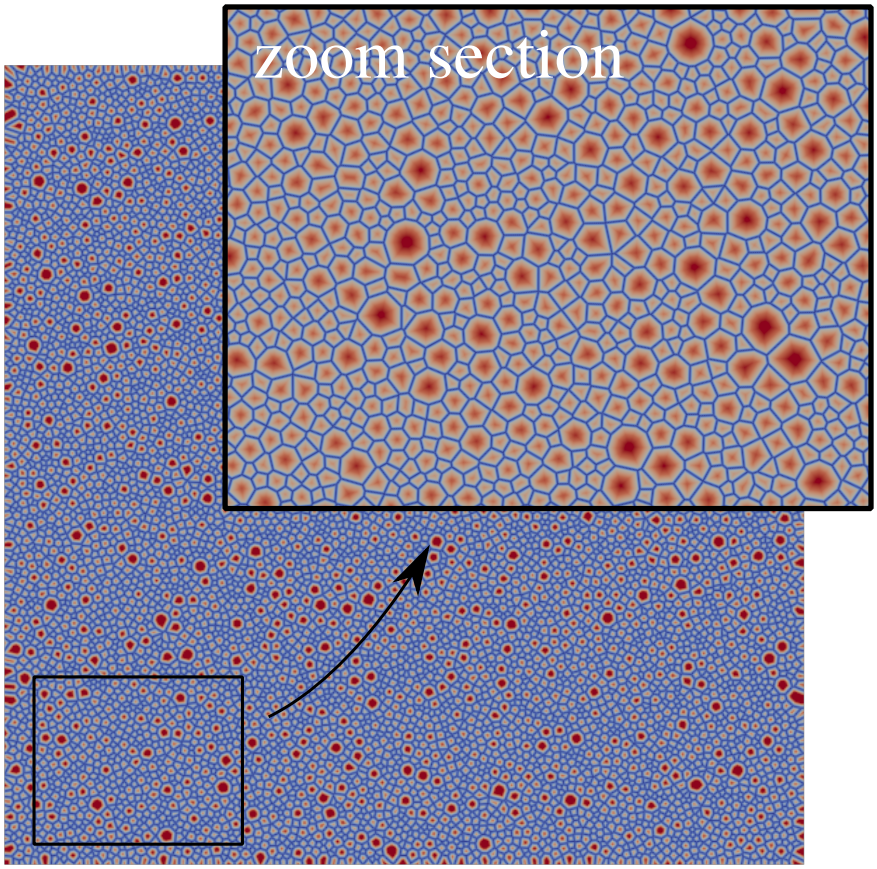


Figure 2.27: Initial state of the microstructure composed of 10000 grains build thanks to a Laguerre-Voronoï algorithm.

size is small, errors on the reinitialization procedure become less important as the distance functions are better described and finally, vacuum regions become smaller. Hence the case with the homogeneous refined static mesh will be treated as the better solution in terms of precision and errors of the two other cases will be computed thanks to its evolution.

The initial microstructure considered is composed of 10000 grains generated using the concept of Laguerre-Voronoi cells [55, 56, 57]. Fig. 2.27 illustrates the initial state for a square domain with surface $A = 10[mm^2]$ and a grain size lognormal distribution with a mean value of $m = 0.017[mm]$ and standard deviation $\sigma = 0.006[mm]$. The values for M and γ are chosen as representative of a 304L stainless steel at 1050° Celsius (with $M = M_0 * e^{-Q/RT}$ where M_0 is a constant $M_0 = 1.56e11[mm^4/Js]$, Q is the thermal activation energy $Q = 2.8 \cdot 10^5[J/mol]$, R is the ideal gas constant, T is the absolute temperature T=1323 [K] and $\gamma = 6 \cdot 10^{-7}[J/mm^2]$). The isothermal treatment is simulated during 3600 seconds.

Figure 2.28 describes the evolution of the mean grain size (calculated in num-

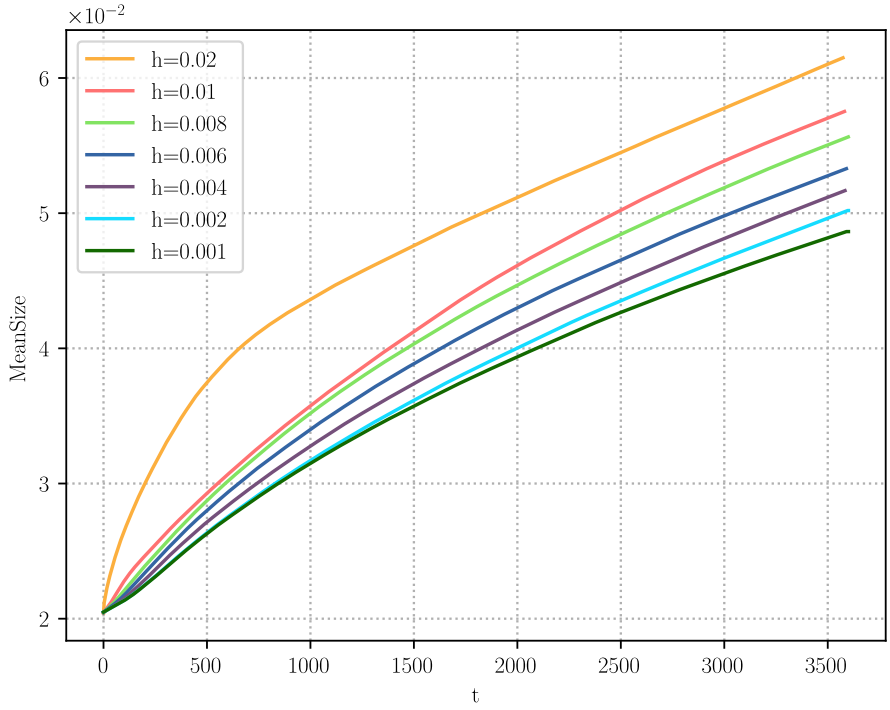


Figure 2.28: Convergence analysis when a homogeneous static mesh is considered: evolution of the mean grain size of the microstructure

ber) when different static meshes are considered. We assume that convergence is reached when the accumulated L_2 error in time remains lower than 5% at $3600s$. Hence, the simulation employing a mesh size $h = 0.001[mm]$ will be considered as the reference case in the following.

Figure 2.29 shows the evolution of each case, the one using the IMA technique for the capturing of the interfaces and the New Fitting and Joining Algorithm (NFJA). Note that two curves are listed for the NFJA method, one (the NFJA) corresponds to a simulation which remeshing is done by employing the same metric field as in the IMA case, the other (NFJA Improved) corresponds to another where the metric field had been improved as shown in Fig. 2.23.

Figure 2.30 shows the comparison of the grain size distributions weighted by surface of each model (IMA, NFJA and Static Mesh) and for the reference case with a mesh size of $h = 0.004$, for the times $t = 1800$ and $t = 3600$. Figure 2.31 shows the L2-Error over these grain size distributions for multiple mesh sizes $h = [0.01, 0.008, 0.006, 0.004, 0.002]$.

Finally, a summary of these results can be made in a single chart: Figure 2.32 shows the evolution of the $L2$ Error in function of the Total CPU-time at the end of the simulation for every case. These charts show that for a precision greater than 5% on the prediction of the mean grain size and greater than 10% on the

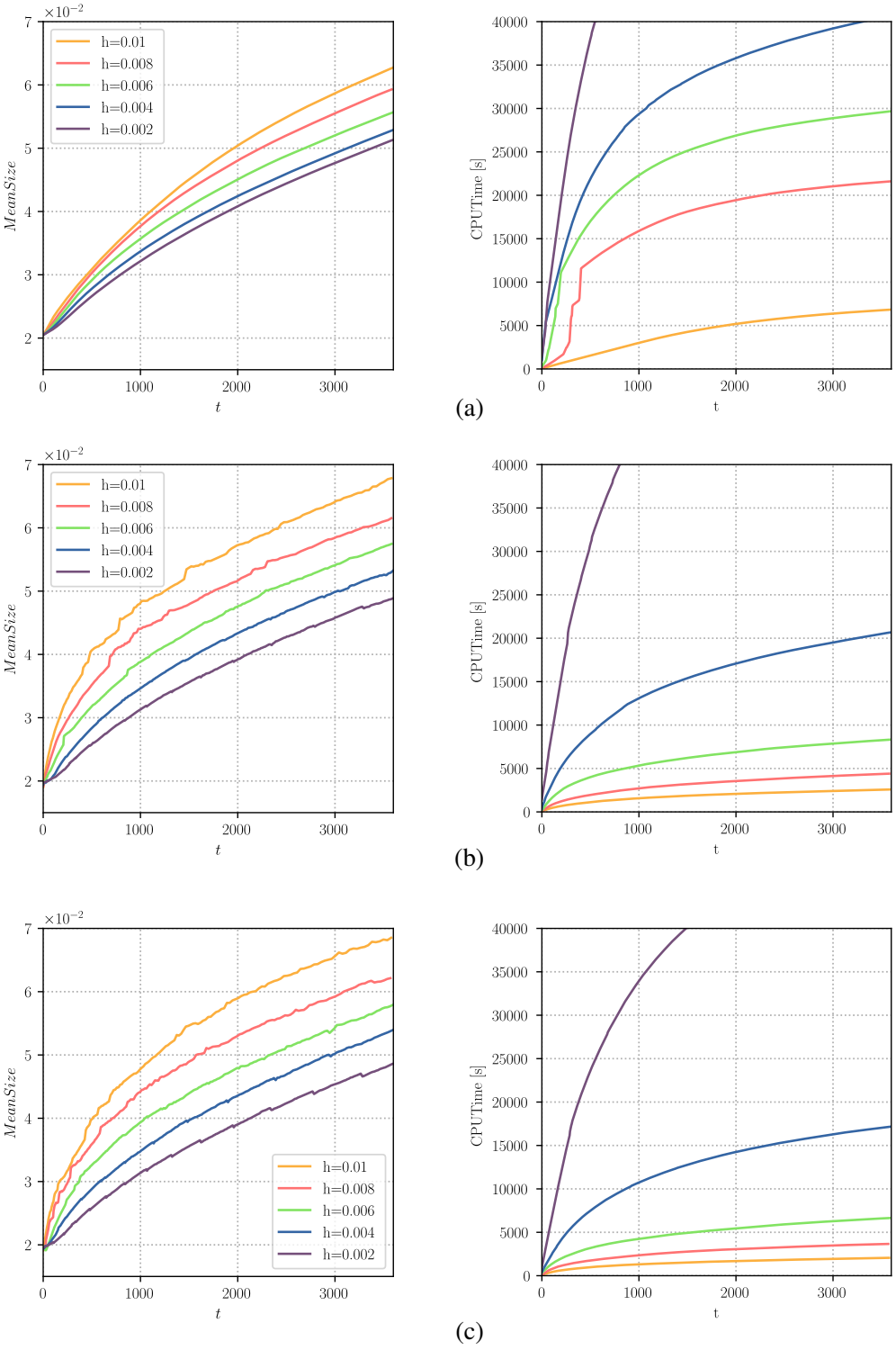
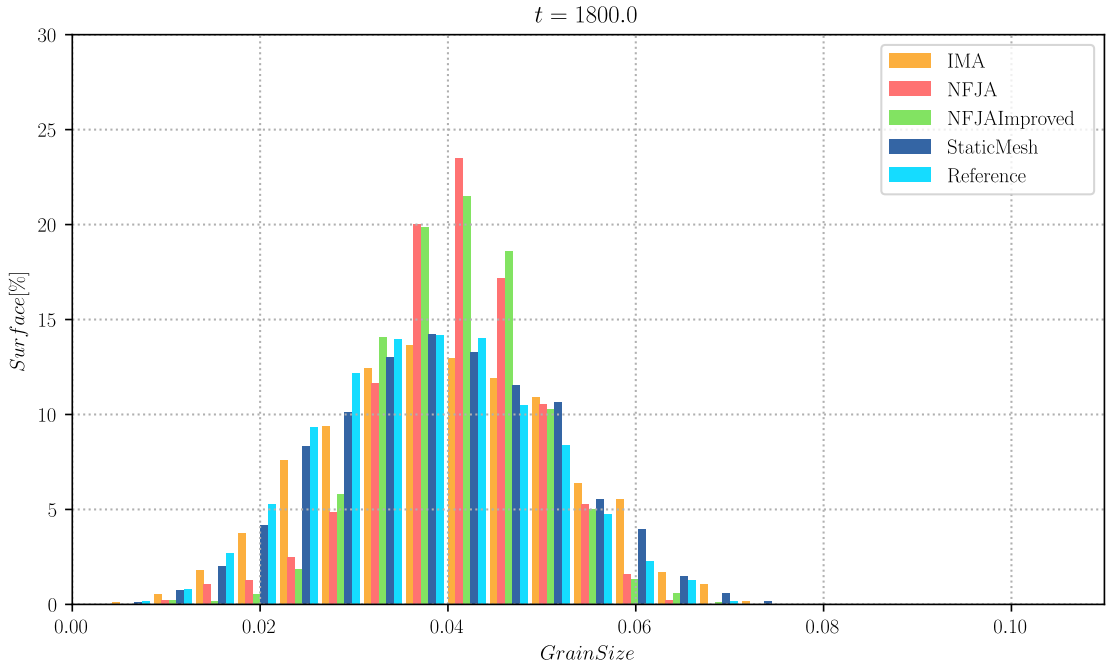
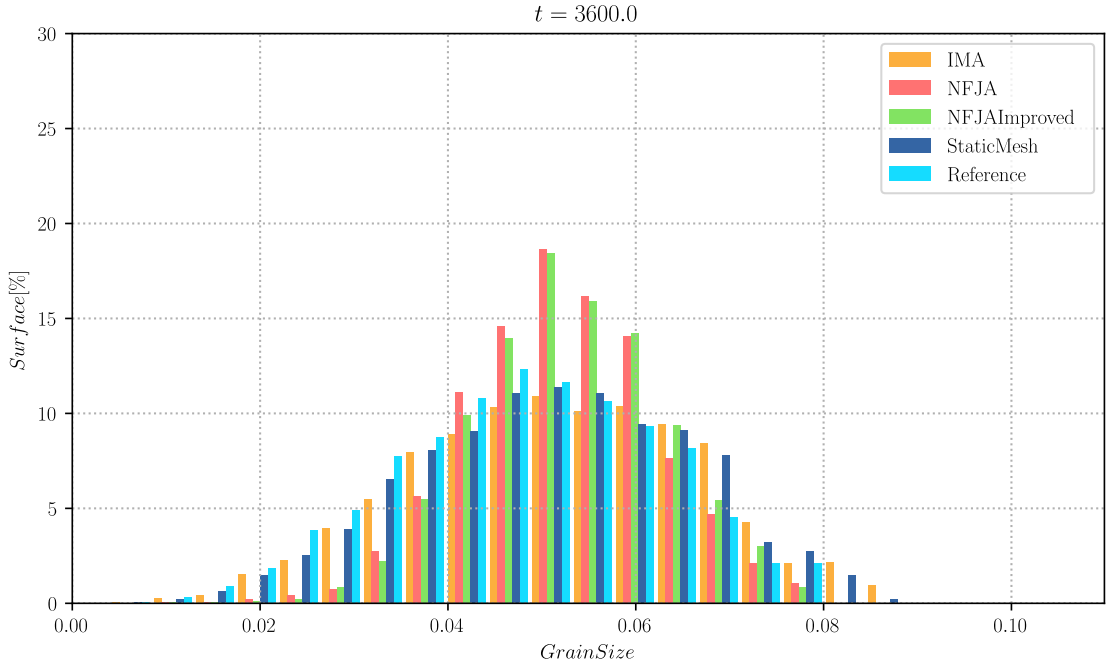


Figure 2.29: Results for each case: (a) using a classic meshing adaptation (IMA) technique, (b) using the New Fitting and Joining Algorithm (NFJA), (c) using the NFJA method with the improvements on the metric field explained in Fig. 2.23 and the reference curve. Left: evolution of the mean grain size and right: CPU-time of simulations, the reference case is not plotted.

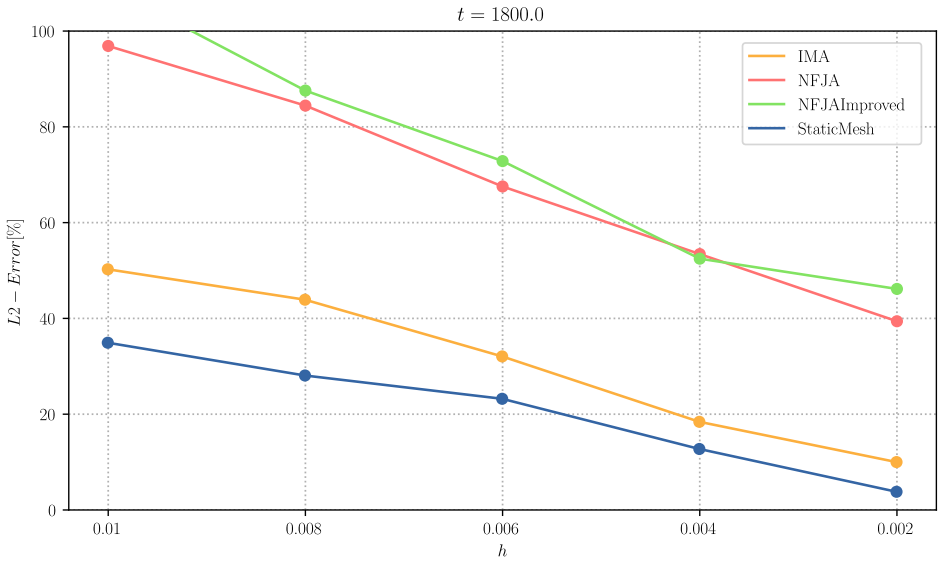


(a)

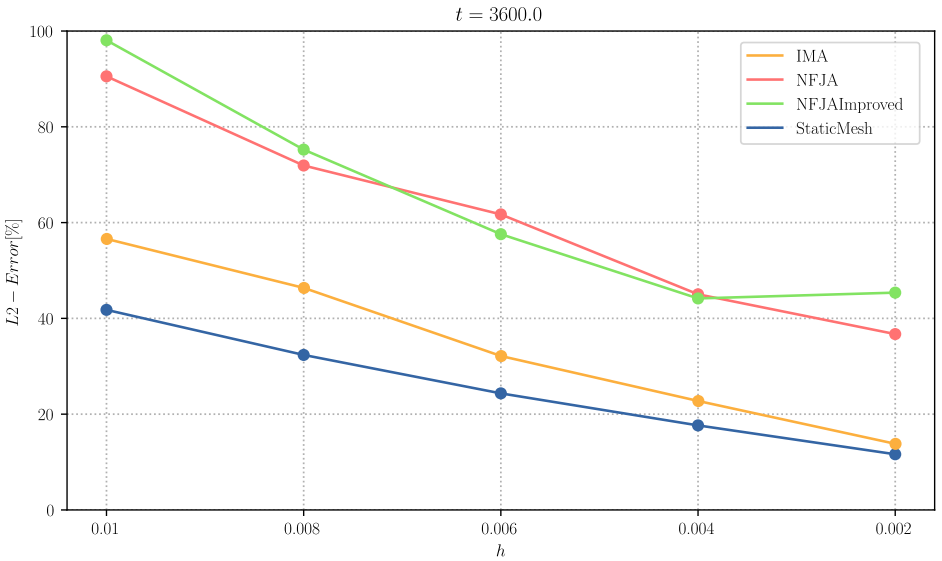


(b)

Figure 2.30: Grain size distributions weighted by surface for the different models and for a mesh size $h = 0.004$. (a) state at time $t = 1800$, (b) state at time $t = 3600$



(a)



(b)

Figure 2.31: L2-Error over the grain size distributions vs the mesh size h. (a) state at time $t = 1800$, (b) state at time $t = 3600$

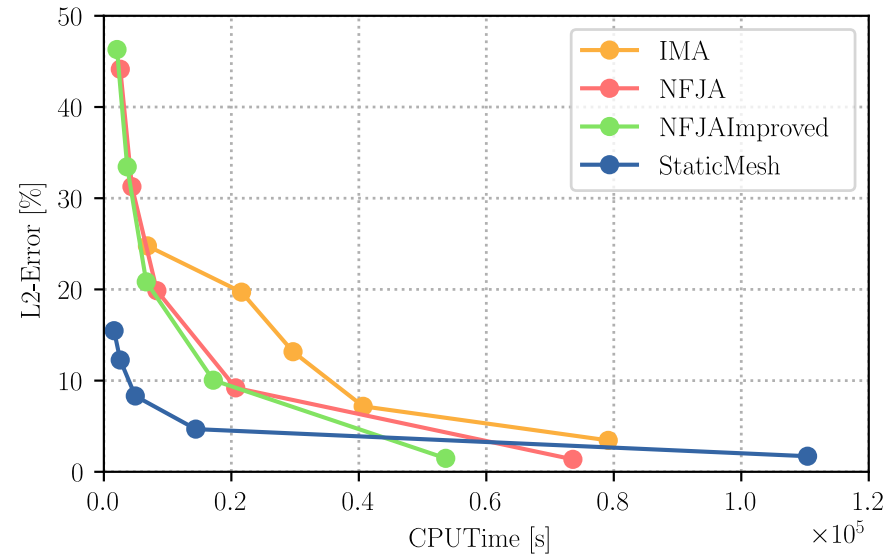
prediction of the grain size distributions the fastest method is the one using a static mesh, meaning that the procedure of remeshing regardless which one we use seems not to give any advantage in terms of CPU-time for the considered configuration.

2.6 Discussion and conclusions

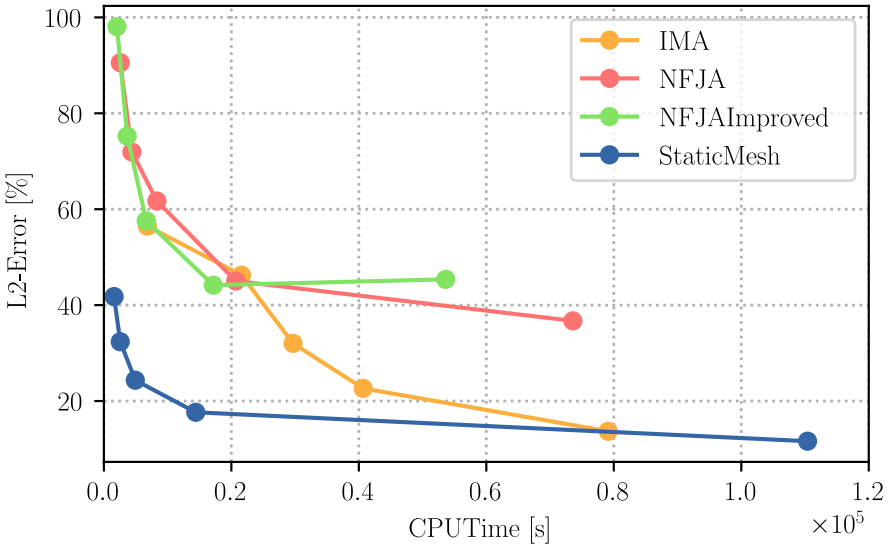
Figure 2.29 shows that the behavior of the CPU-time for each of the simulation is not linear, this is due to the fact that the number of grains is also changing during the simulation: the fewer grains there are, the smaller the zone needed to maintain with a refined mesh and the smaller the time of computation by increment.

Figure 2.32(a) and (b) illustrate that for IMA and the NFJA methods the accuracy range is wider than for the one using a static mesh. CPU-time for both cases (IMA and NFJA) are very near, however, they represent very different processes. The time needed to remesh with the NFJA for a fixed number of elements is higher because the fitting process adds an extra amount of computational work. On the other hand, some of our observations showed that the NFJA needed fewer mesh elements to properly define the interface even if the asked metric field was the same. Thus the proposed new front-tracking approach appears already as competitive compared to the existing LS front-capturing approach used in the state of the art in the context of unstructured FE mesh [7, 6, 143, 169]. However, when comparing the CPU-time obtained for the simulations using a static mesh, every test (except for the one with $h = 0.002$) turn out to have a lower CPU-time than for the corresponding IMA and NFJA approaches. This result in addition to the fact that the error ranges are close, take us to conclude that the remeshing process (IMA and NFJA) does not reduce the computation time of a 2D few-thousand multidomain simulation in context of the proposed recent algorithm (diffusive formulation, coloring/recoloring algorithms, optimized direct reinitialization and treatment at multiple junctions).

Figures 2.30, 2.31 and 2.32(b) show that in one hand the IMA case behaves better in terms of accuracy on the prediction of the grain size distributions contrary to 2.32(a) where the NFJA case predicts better the mean grain size. Of course, the grain size distribution gives a better description of the global state of the polycrystal hence we conclude that the IMA approach is more accurate for a fixed mesh size. On the other hand, if comparing the CPU-time in either 2.32(a) or (b) the computational cost needed for the IMA case is always higher than for the NFJA case, but the error given by the NFJA remains too important (even for the smallest mesh size h, last point from right to left) to consider it at the optimal candidate for a multidomain simulation. In fact, comparing the grain size distributions (see Figure 2.30) reveals that the kinetics of the NFJA method



(a)



(b)

Figure 2.32: $L2$ Error vs the Total CPU-time at the end of the simulation. Each point of the same curve represent a simulation with a different mesh size (from left to right: $h = [0.01, 0.008, 0.006, 0.004, 0.002]$) . (a) L2-Error over the mean grain size, (a) L2-Error over the grain size distributions

is clearly different from the one using an implicit description of the interfaces (IMA or Static Mesh) being faster on the evolution of the small grains and slower on the evolution of the big ones. This conclusion seems coherent with the results described for the T-junction and square-shrinkage cases where the kinetics of the triple junctions was systematically overestimated by the NJFA approach comparatively to others. If this weakness is automatically linked to the mesh topological operations realized at the multiple junctions, to solve is not straightforward and constitutes a perspective of this new approach.

Indeed, this approach remains clearly of interest as further improvements could be made with the use of the NFJA that could result in a suitable method to model multidomain problems with a diminution of the CPU-time and of the error that could not be possible to make by employing a more classical approach. Indeed, with this approach, geometrical data such as interface normal and curvatures can be computed directly from the body-fitted mesh using the position of the interface nodes only, instead of relying on the costly and inaccurate approximation of the LS field derivatives which could lead to the direct use of Eq. 2.10 in a stabilized framework, or even lead to the use of a Lagrangian scheme, in which grain boundary kinetics is simulated via the direct movement of the boundary nodes, and not through the resolution of a transport equation. These developments are in fact the topic of discussion in chapters 3, 4, 5 and 6 of the present manuscript.

Résumé en Français du Chapitre 2

Ce chapitre est dédié à l'introduction d'une nouvelle stratégie de remaillage explicite. Sur la première partie de ce chapitre, deux stratégies de remaillage appliquées aux interfaces sont détaillées: maillage implicite et maillage explicite des interfaces. Un maillage implicite requiert une discrétisation assez dense aux environs des interfaces pour atteindre une représentation précise des discontinuités physiques, alors qu'un maillage explicite n'a pas de besoin spécifique pour bien définir les interfaces des grains sur une microstructure. Dans ce sens, une méthode a été proposé par Shakoor et al. permettant d'obtenir une discrétisation explicite des interfaces sur domaines initialement discrétisés implicitement à l'aide de l'approche LS. En principe, l'utilisation de cette méthode pourrait réduire la densité des maillages utilisés par la méthode LS sans réduire la précision sur la définition des interfaces, et donc, autoriser une meilleure performance des simulations des évolutions microstructurales. Elle n'est cependant pas directement applicable au contexte des jonctions multiples.

Ainsi, une extension de cette méthodologie a été développée et est présentée. Différents cas tests sont présentés pour évaluer les différentes stratégies disponibles: maillages implicites adaptés avec des éléments isotropes (IMA) aux interfaces, maillages explicites statiques raffinés sur tout le domaine (sans remaillage aux interfaces) (SM), et nouveaux maillages explicites (NFJA) avec différentes configurations de densité.

Les résultats montrent qu'en 2D la méthode SM présente les meilleurs résultats en matière de performance et de précision, suivi par la méthode NFJA. Toutefois, ces tests ont été réalisés sur un environnement favorable à la méthode implicite et plusieurs améliorations pourraient être réalisées dans le contexte du modèle NFJA: le calcul des propriétés géométriques aux interfaces pourrait être effectué sur les interfaces uniquement, au lieu d'utiliser des gradients des fonctions LS. Ceci pourrait conduire à l'utilisation de l'équation de transport (Eq. 2.1) directement, ou même, le développement d'un modèle Lagrangien des mouvements des interfaces, où le mouvement des noeuds "fittés" du maillage représenterait les mouvements des joints de grains.

Chapter 3

A novel highly efficient Lagrangian model

In this chapter, a new method for the simulation of evolving multi-domains problems will be presented. The method is inspired by the front-tracking approaches where only the interfaces between domains are discretized. The presented method maintains the discretization of the interior of the domains using an evolving triangular mesh (valid for a FE study) and treats the topological events such as the disappearance of domains or the creation of interfaces by means of selective local remeshing operations. Geometric properties are only computed on the interfaces using local spline reconstructions and interfaces are moved using a Lagrangian approach ensuring at all times the validity of the mesh.

Accuracy and computational cost of this method will be evaluated in the context of microstructural evolutions, specifically for the GG mechanism and it will be compared to a more classical front capturing approach based on a LS description already detailed in chapters 1 and 2.

This chapter has been published in [20].

3.1 Introduction

As introduced in chapter 1, Multiple numerical models have been developed to simulate multidomain problems. In the context of the FE Method, the LS method is a powerful tool capable of accurately handle large interface deformations and topological events with relative ease. The FE-LS numerical framework has been successfully applied to predict the microstructural evolution of a polycrystalline material subjected to thermal or thermomechanical processes, where phenomenons as GG [5] and ReX could appear [6, 7, 142, 145]. However, the time needed to perform such simulations could go from some minutes for a domain with hundreds of grains in 2D, to many days (even weeks) for a simulation involving hundreds of thousands of grains in 3D.

In section 1.3.5 it was discussed that some optimizations could be applied to the FE-LS method with the objective of reducing its computational cost (hence the CPU-Time) without reducing its accurateness [8, 19]. However, these optimizations have a certain limit imposed by the nature of the FE method (discretization of the domain, interpolation degree and resolution) and the LS method (reinitialization and treatment of vacuum regions).

Another numerical approach to simulate multidomain problems is the Voronoi implicit interface method (VIIM) introduced in [193] and further discussed in [194] which uses a single unsigned level-set and a Voronoi approach to solve the inconsistencies at the interface produced after affecting the level set field with a given velocity (by solving the necessary equations of the related phenomena), over a fixed Eulerian mesh.

Alternatively to the FE-LS method, models based on the Lagrangian displacement of interfaces can be used as in the context of vertex approaches [14, 15, 16, 17, 18] or front-tracking methodology [9, 10, 11, 12, 13]. These methods explicitly describe the interfaces in terms of vertices. Each increment, the velocity of the interfaces is computed and applied. The application of these strategies is less demanding than for the FE-LS method, because the dimension of the discretization and the computation is reduced by one dimension (only interfaces are discretized, while the grain interior is not). Nonetheless, difficulties of these approaches remain the complexity of handling all the possible topological events, such as the disappearance and appearance of new interfaces and domains or the contact between crossing interfaces. Another issue of this method is that it is not possible to take into account in the computations, physical mechanisms occurring inside the grains as there is not a discretization inside of them.

In this paper, a new method for the simulation of evolving microstructures by curvature flow is presented. The method is inspired by the vertex approaches in the sense that geometrical properties are only computed at the interfaces and

the migration of the grain boundaries is defined thanks to a Lagrangian scheme. However, The presented method maintains the discretization of the bulk of the grains using an evolving triangular mesh, hence the discretization will remain valid for a FE computation if needed (like in context of crystal plasticity or dynamic recrystallization computations). The treatment of the topological events such as the disappearance of domains or the creation of interfaces can be handled by means of selective local remeshing operations performed on the triangular mesh.

The proposed numerical approach: TOpological REmeshing in lAgrangian framework for Large interface MOTION (ToRealMotion, hereafter TRM) will be compared to the FE-LS approach using a classical front capturing LS-FE framework [5, 167, 168, 169]. These comparisons will be based on specific test cases for 2D-GG, using the same approach and results presented in chapter 2. Given the application of the new model in this chapter, the term *domain* will reference an individual grain in a microstructure, however, the new TRM approach can be extended to any massively multi-domain problem immersed in a FE mesh.

3.2 Numerical method

3.2.1 Data structure: points, lines and surfaces

In the present numerical method, multiple data structures have to be defined. These data structures are organized following the geometric entity that they represent. This way, three classes of geometric entities are defined: Points, Lines and Surfaces. Each one of these classes of entities contains its own data structure with a set of nodes and elements (elements only in the case of Surfaces) that are used for its discretization on the triangular mesh.

Inspired by the classification of low-degree manifolds [195], a degree is attributed to each node of the FE mesh. This degree depends on the class of the geometrical entity that they locally represent: Nodes representing a point (P-Nodes) have a degree equal to 0, Nodes representing a line (L-Nodes) have a degree equal to 1 and Nodes representing a surface (S-Nodes) have a degree of 2.

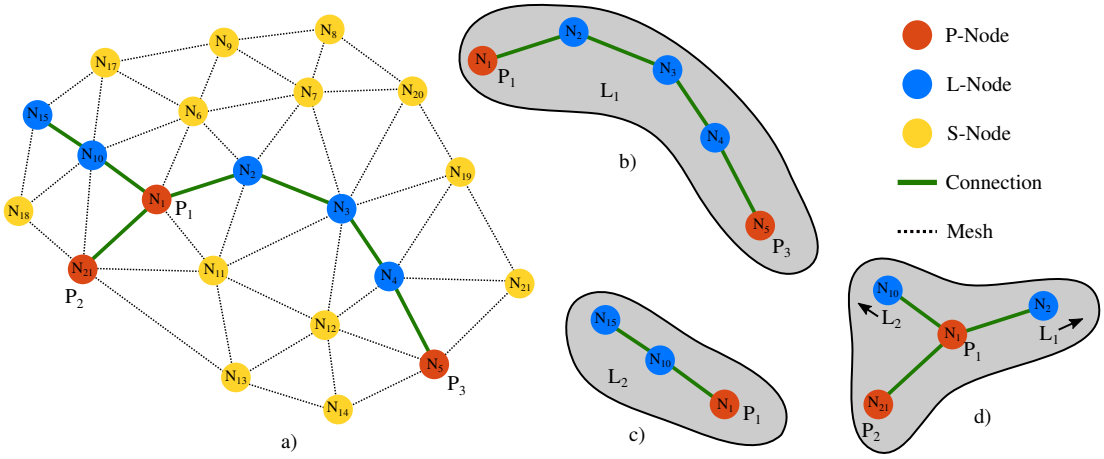


Figure 3.1: discrete geometric connectivity immersed in a triangular mesh. a) Mesh containing 21 Nodes: 3 P-Nodes (red), 5 L-Nodes (blue) and 13 S-Nodes (yellow). b) structure of line L_1, c) structure of line L_2, d) structure of point P_1.

The simplest data structure is given for a point, this geometric entity is represented by one P-Node and a set of local geometric connections (at least 3 connections). These connections refer to the current linking of this point to other points or other lines as shown in Fig. 3.1 (a) and (d), here the point P_1 is connected to the point P_2 and to two lines L_1 and L_2 containing L-Nodes N_2 and N_{10} respectively. Note that in Fig. 3.1(a) even if point P_2 is directly connected to L-Node N_{10} by the mesh, this does not mean that there exists a geometric connection between them, otherwise N_{10} could not be a L-Node but a P-Node because it would have 3 geometric connections.

Node	N_1	
Connections		
Type	Entity	Node
Point Connection	P_2	N_{21}
Line Connection	L_1	N_2
Line Connection	L_2	N_{10}

Table 3.1: example of the data structure of point P_1 from figure 3.1.

The data structure of geometric lines is a little more complex, a normal structure of a line consists of a set of L-Nodes (containing at least 1 L-Node) and 2 optional limit points. Additionally, we have chosen to maintain an order on each line, beginning with an initial point, then the ordered set of L-Nodes and ending with the final point. This choice was made to ease the approximation of the lines by natural splines as it will be explained further. Figure 3.1(b) and (c) illustrate two examples of lines: lines L_1 and L_2, where L_1 is a typical line with two points and 3 L-Nodes and L_2 a line with only one point and 2 L-Nodes, each line can be ordered either way, L_1 from point P_1 to P_3 or inversely and L_2 from P_1 to L-Node N_{15} or inversely. Moreover, note that the connection P_1 to P_2 do not represent a line because there is not a L-Node between them. While $P - P$ Connections represent an interface as well as geometric lines, they will be treated differently in the presented model. Tables 3.1 and 3.2 summarize an example of the data structure of point P_1 and line L_1 from Fig. 3.1.

Initial point	P_1
Node set	
local index	Node
0	N_2
1	N_3
2	N_4
Final point	P_3

Table 3.2: example of the data structure of line L_1 from Fig. 3.1.

Finally, the data set of geometric surfaces contain a non empty set of elements, a set of S-Nodes, a set of limit lines and a set of limit points. Here, the order has been chosen not to be relevant, thus all sets are unordered. Figure 3.2(a) shows the same discretization of Fig. 3.1(a) with additional element information and Figures 3.2(b-d) show its decomposition in surfaces S_1, S_2 and S_3. Table 3.3 presents an example of the data structure of Surface S_2.

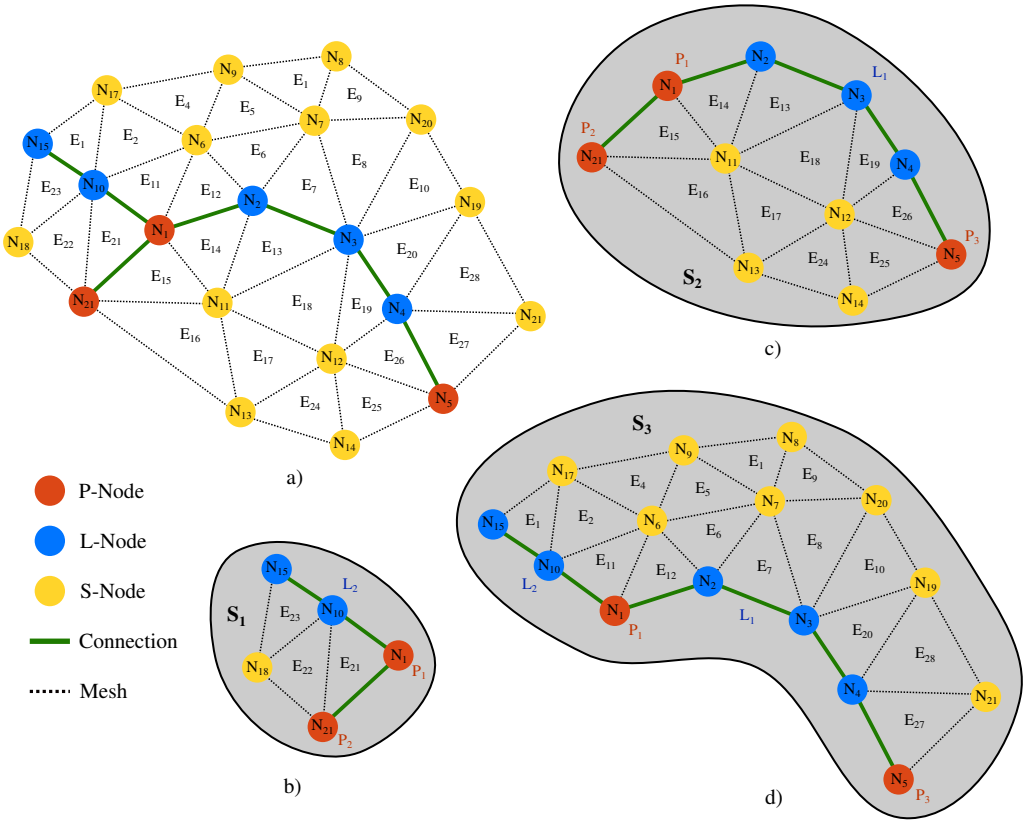


Figure 3.2: discrete geometric connectivity immersed in a triangular mesh. a) Mesh of Fig 3.1(a) with additional element information b) structure of surface S_1, c) structure of surface S_2, d) structure of surface S_3.

Element set
E_{15}
E_{13}
E_{16}
E_{14}
E_{17}
E_{18}
E_{24}
E_{19}
E_{25}
E_{26}

Node set
N_{11}
N_{13}
N_{12}
N_{14}

Point set
P_1
P_2
P_3

Line set
L_1

Table 3.3: example of the data structure of surface S_2 from Fig. 3.2.

3.2.2 Preprocessor: LS-TRM interface and geometric reconstruction

Considering the importance and the capabilities of the LS method, we have considered it useful to develop an interface between this method and the presented

model. This interface will treat any initial state described using one or multiple LS fields to be used as an entry state on the TRM model. The interfaces will produce a body-fitted mesh and a set of nodal data representing the degree of each node as explained at the beginning of section 3.2.1. Moreover, the body-fitted mesh and the nodal degree data will be used on the reconstruction of the geometric entities presented also in section 3.2.1. The idea behind this reconstruction is to allow the code to optimize operations such as the computation of geometrical properties of the interface, which only needs to be performed on line entities for the presented 2D model.

Body fitted mesh

Normally, when using the LS method, interfaces are described implicitly as the zero iso-value of the considered LS fields [146, 6], each field is interpolated on the FE mesh allowing to identify each grain as a sequence of implicit segments (usually different from the edges of the mesh), as a consequence, the mesh used to define the LS fields is not directly suitable for the TRM model. First, the LS data set must be used to obtain a body fitted mesh, where the interpolated interfacial segments are also represented by the edges of the mesh. This particular purpose can be addressed by the use of the remeshing technique presented in chapter 2, where the works introduced in [184, 161] lead to the development of a new remeshing procedure, able to produce vacuum-less body-fitted meshes, via a *fitting and joining* algorithm (see chapter 2).

Figure 3.3 illustrates the behavior of the algorithm when performed on a typical multiple junction of three domains. Figure 3.3(b) clearly shows how all the created nodes lie at the interface between two domains except for one, which is at the multiple point between the three domains. This reconstruction is here purely geometrical as its aim is to close vacuum regions around triple junctions and not to give its exact position inside an element, as we have considered that the accuracy of the definition of a multiple point should be defined by the element size and not by its undefined location over one single element. Furthermore, one could give weights to each intersection in order to place the triple point in a more energetically stable position if necessary. However, for our TRM model this is not relevant as this procedure only serves as a part of the pre-processor step.

Up to this point, the fitted mesh is sufficient for the TRM model, however, the mesh is optionally adapted using local remeshing operations with the intention of increasing its quality (which is very poor at the boundaries after the fitting see 3.3(b)), of course, these adaptations are done without disrupting the fitting of the mesh. Fig. 3.3(c) shows the resulting body-fitted mesh after the adaptation procedure.

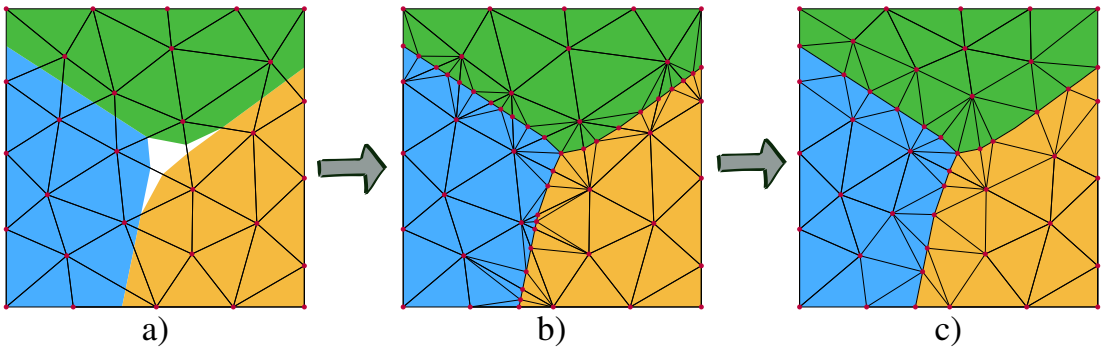


Figure 3.3: Example of behaviour of the *fitting and joining* procedure presented in [19] when performed over a triangular mesh with 3 LS fields interpolated linearly, each color represents a phase. a) initial state, interfaces are implicitly described, b) state after applying the *fitting and joining* algorithm, nodes appear at the interface between 2 phases and at the multiple junction. Vacuum regions disappear and all elements belong to one phase. c) Resulting mesh after the mesh adaptation procedure introduced in chapter 2.

Nodal geometric tagging

Using a LS approach and a body-fitted mesh makes it possible to obtain the surface nodes by looping over all the domain and extracting all the nodes for which at least one of the LS fields is zero. However, this information is not sufficient. Triple points need to be identified before one attempts to reconstruct the analog data structure presented in section 3.2.1.

Once the body fitting mesh is obtained, a nodal tagging is performed. Each node will be tagged regarding its geometric degree as explained at the beginning of section 3.2.1. This is achieved by iterating over all the LS fields stored on a node, if one of its LS fields is positive, this means that we are at the interior of a phase and a degree equals to 2 (S-Node) must be attributed to that node. Then, if two of the LS fields are equal to zero and the rest are negative, that node lies on a line and a degree equals to 1 (L-Node) is given to it. Finally, if more than 2 level set fields are equal to 0 this node is a multiple point, hence obtaining at the end a degree of 0 (P-Node). Fig. 3.4 shows an example of nodal geometric tagging over the body fitted mesh obtained in Fig. 3.3.

Once the tagging is complete, the level set information is not necessary any further, the interface is complete and the new data set (body-fitted mesh and tagging) is ready to be sent to the TRM model for the geometric reconstruction.

Point reconstruction

The point reconstruction is always the first step of the geometrical reconstruction process. Points are the entities that hold lines together and that allow the com-

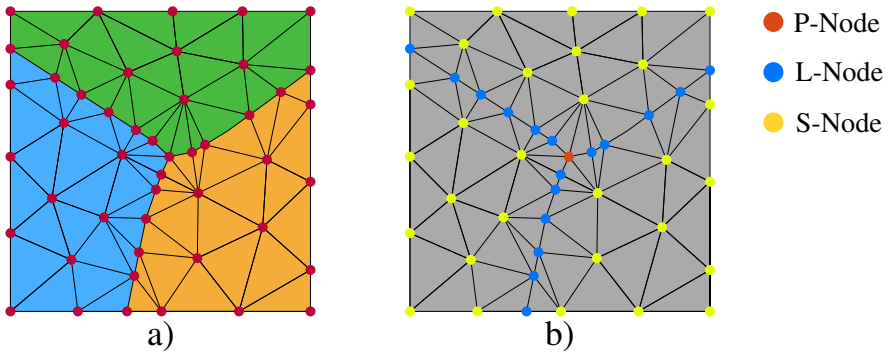


Figure 3.4: Example of the nodal geometric tagging performed on a body fitted mesh. a) considered topology and b) resulting nodal tagging.

putation of the *apparent geometrical properties* of the interface at P-Nodes. Each node tagged as a P-Node will initiate the creation of a point entity, this node is automatically attributed to the new created point. Concerning the connections of the point (explained in section 3.2.1) As there are no lines nor points created yet, connections will not be attributed on the creation process.

Once all points are created, point connections can be created. This is done by iterating over all points and querying if their corresponding P-Node N_0 is connected to any P-Node N_1 on the triangular mesh, if it is true, the point of P-Node N_0 will add N_1 as a connection.

Lines reconstruction

Once all points are reconstructed, the line reconstruction is performed, section 3.2.1 shows that line entities maintain an ordered L-Node sequence. This order is generated by performing a recursive algorithm on the tagged L-Nodes of the mesh.

If a node is connected to a multiple point in 2D, the latter will be considered as the end or the beginning of the line (depending on the stage of the recursive algorithm).

The reconstruction procedure initiates by defining a queue of L-Nodes to be attributed, this queue corresponds at this stage as all tagged L-Nodes. The first L-Node of the queue is considered as the initial node for the recursive algorithm which will search for adjacent L-Nodes that still appear in the queue. The procedure is repeated recursively on the adjacent nodes till one of the neighbors is tagged as a P-Node (which has a corresponding point already created). Each time a node is treated, it is erased from the queue of L-Nodes available. Moreover, the points found will be stored as members of the lines and they will be taken into account for the calculations of the geometrical properties of the interface.

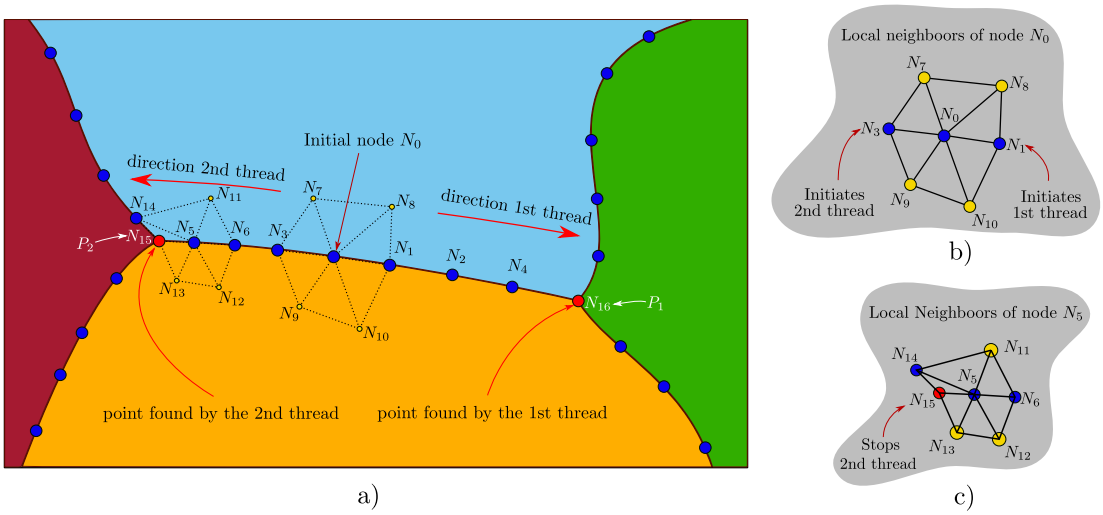


Figure 3.5: Example of construction of a line in 2D. a) L-Node N_0 is chosen as the initial node of the recursive algorithm, each treated L-Node launches the procedure on each adjacent L-Node not treated until it founds an adjacent P-Node. Only some relevant elements are plotted. b) Local node patch of Node N_0, when treating N_0, N_1 initiates a recursive thread while N_3 launches another. c) Local node patch of node N_5, the second thread launched by N_3 is stopped by the presence of P-Node N_{15}.

For the example shown in Fig. 3.5(a), two lists of nodes are created by two corresponding recursive threads, each thread is launched by the presence of a L-Node in the local neighborhood of node N_0 (see Fig. 3.5(b)) that still appear on the queue. The recursion stops once one of the neighbor nodes is a P-Node as in Fig. 3.5(c), where the coupled point P_1 of the P-Node N_4 is added to the list as a final point. Two lists are then created, the first is composed by the L-Nodes N_1, N_2, N_4 and point P_1 and the second by N_3, N_6, N_5 and point P_2. The line will then be defined as the concatenation of the inverse first list, the initial node N_0 and the second list of L-Nodes, its initial point will be defined as the point found by the first thread while its final point as the point found by the second thread.

By performing this algorithm, each L-Node of the mesh is identified and affected to a specific line having in account its relative position to other nodes on the same line. This information will be used to choose a patch of nodes that will serve as the support for an approximation/interpolation of the interface.

Once all lines are created, the missing connectivity of the point entities can be created. This is done by iterating over all lines, creating a connection between its initial/final point and the current line. Note that this is a dual link: lines store the points that are attached to it and points store to which lines it is attached,

this strategy enables to perform quick connection searches on both ends (points and lines).

Surface reconstruction

Surface reconstruction was developed similarly to the line reconstruction algorithm. Here, a queue containing all S-Nodes from the mesh is initially created, then a recursive algorithm identifies S-Nodes and elements belonging to the same domain while storing limit lines and limit points found.

Take for example the state presented in Fig. 3.6, a set of S-Nodes (yellow) and a set of L-Nodes (blue) has been tagged and one line L_1 (green) has been identified by the process presented in section 3.2.2. Here two domains separated by L_1 need to be identified. Initially, the first S-Node of the queue is chosen to initiate the recursive algorithm, this initial S-Node is inserted to a new (empty) Surface S_1 along with its element patch giving end to the first iteration. The next iteration computes a new set G_{newSN} of S-Nodes to be added to S_1, this set of S-Nodes is composed by the set of nodes appearing on the element set of S_1 ($Elements(S_1)$), minus the set of nodes already inserted in S_1 ($Nodes(S_1)$):

$$ELTs = Elements(S_1) \tag{3.1}$$

where function $Elements(S_1)$ extract the elements present in surface S_1[1].

$$G_{NE} = \left\{ \; Nodes(Elt_i) \quad \forall \quad Elt_i \in Elts \; \right\} \tag{3.2}$$

$$G_{SNE} = \left\{ \; N_i / N_i \in G_{NE} \quad with \quad Type(N_i) = SNode \; \right\} \tag{3.3}$$

where function $Nodes(Elt_i)$ extract the nodes belonging to Elt_i[2], $Type(N_i)$ extracts the type of the node N_i (P-Node, L-Node or S-Node), G_{NE} are the nodes used to form the set of elements $ELTs$, and G_{SNE} are the S-Nodes present in G_{NE} extracted with the expression $Type(N_i) = SNode$.

$$G_{newSN} = G_{SNE} \setminus Nodes(S_1) \tag{3.4}$$

the nodes in G_{newSN} are colored in white in each frame of Figure 3.6. Once these nodes are computed, they are inserted in Surface S_1 and extracted from the queue.

[1]Function $Elements(Entity)$ extracts the Elements present in $Entity$, where $Entity$ can be a surface or a node. When extracting the elements from a node N, the result will be the set of elements surrounding N.

[2]Function $Nodes(Entity)$ extracts the nodes present in $Entity$, where $Entity$ can be a point, a line, a surface or an element.

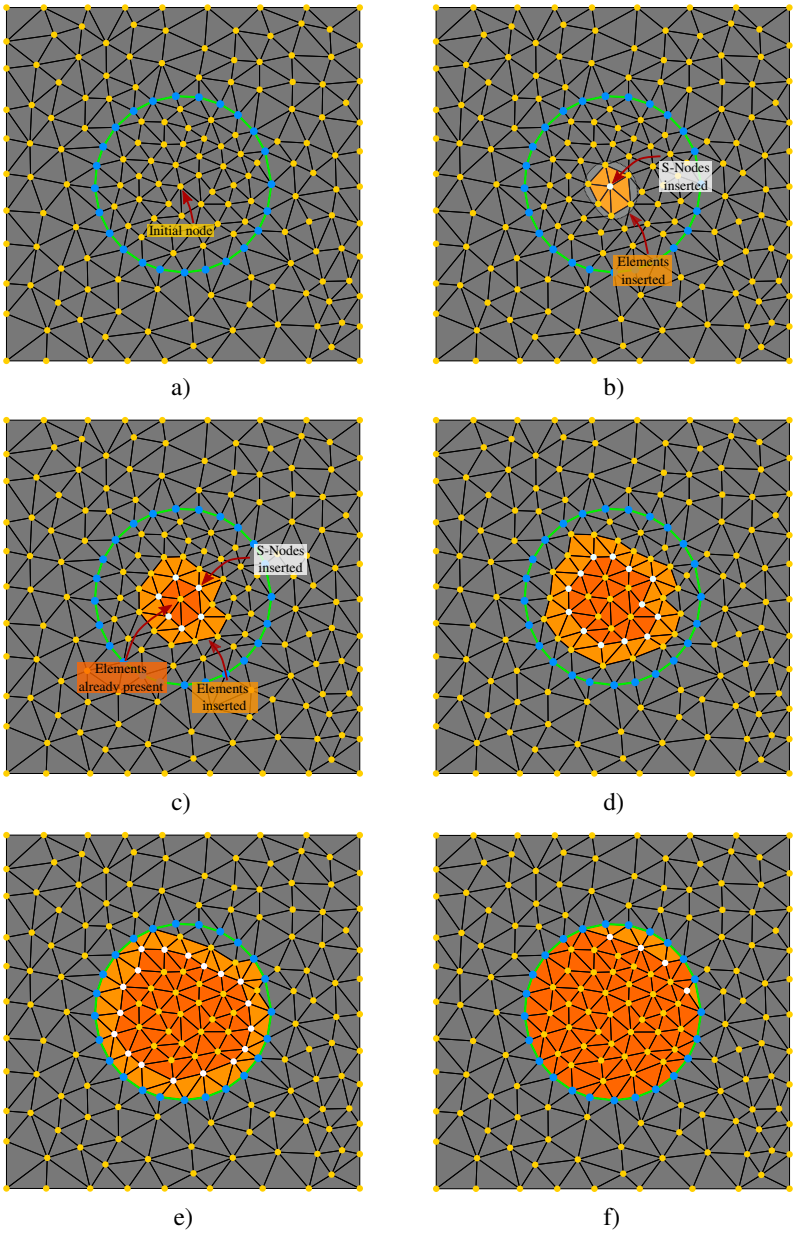


Figure 3.6: Example of construction of a surface in 2D, L-Nodes - blue, S-Nodes - yellow, Line L_1 - green, elements in the queue - gray, elements attributed to Surface S_1 - orange. a) an Initial S-Node is chosen, b) the L-Node is inserted in Surface S_1 along with its element patch, c) and d) the new S-Nodes of the inserted elements are also inserted to S_1. Each new S-Node inserts its unattributed elements (those that do not belong to any surface) to S_1. e) the first element having a node belonging to line L_1 inserts it to the limit lines of S_1. f) the recursion stops as there are not new S-Nodes to insert in the next iteration.

The set of nodes filtered G_{other} from G_{NE} in Eq. 3.3 ($G_{other} = G_{NE} \setminus G_{SNE}$) are the nodes having a $Type(N_i) = PNode$ or $Type(N_i) = LNode$, these nodes

are used to add their coupled geometric entity (line or point) to S_1 as a limit entity as explained in section 3.2.1.

The new set of S-Nodes G_{newSN} enables also to compute the new elements to be inserted in S_1: the set of elements G_{Elt} of their element patch that do not belong already to S_1:

$$G_{Elt} = \left\{ \; Elements(N_i) \quad \forall \quad N_i \in G_{newSN} \; \right\} \setminus Elements(S_1). \tag{3.5}$$

Finally, the recursion stops once the set G_{newSN} is void, meaning that there are no more S-Nodes to treat for that surface/phase as in Fig. 3.6(f). Note that the queue of S-Nodes is still not empty, hence the recursion restarts with the first node of the queue (which necessary belongs to a new surface S_2). This process is repeated till the queue is empty.

Once all surfaces, lines and points are created, the preprocessing is complete.

3.2.3 Computation of geometrical properties: curvature and normal

One of the objectives of the presented TRM method is to allow a more efficient way to calculate some geometrical properties by looping only over the nodes belonging to the interface.

With the basis of the data structure presented in section 3.2.1 and the preprocessing procedure presented in section 3.2.2, it is possible to approximate each geometric line to the analog parametric curve given by its ordered list of nodes. The choice of the mathematical approach for the approximation will be given with the aim of the desired information (curvature and normal). Having this in mind, one approach has been taken into account: Natural Parametric Splines (NPS) [196] which allow capturing the local variations of normal and curvature with a piece-wise polynomial of third degree.

Natural Parametric Splines

A spline is defined as a function constructed piece-wise which is totally constrained by applying certain conditions to the external limits of each sub-region. The choice of the conditions will change the behavior of the parametric interpolation; in our case, we will limit our study to natural splines: a fitted surface for which the continuity of the first and the second derivative is assured at every node in addition to the conditions for the most outer nodes where the second derivative is imposed to zero. In our algorithm, we have chosen to compute the splines using each node of a given line as the geometrical support.

The main difference of the NPS approach with other methods (for example a moving least square (MLS) approach as in [197, 198]) is that the curvature and the normal are calculated locally without over smoothing the interface, allowing to capture the small variations of curvature from node to node with a relative high precision. However, the solution over time for a velocity proportional to the curvature could be unstable for high values of time step dt, but if dt is sufficiently low, the interface should find a position that minimizes the surface energy at all points (see Figure 3.7).

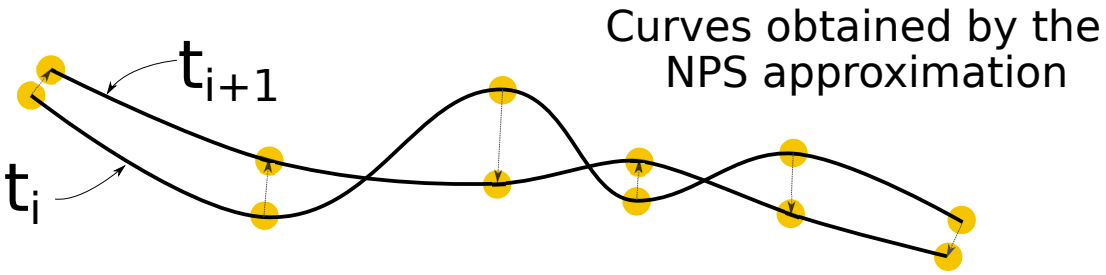


Figure 3.7: Parametric piece-wise curve of a NPS approximation. Local curvatures are captured; The interface is capable to correct itself over time into a smooth interface if dt is low.

The stability of the TRM model using natural parametric splines will be discussed in section 3.4.2.

3.2.4 Selective remeshing operations: preserving topology

One of the main features of the presented TRM model is its remeshing procedure. Section 3.2.1 presented the data structure employed by the TRM model, this data structure needs to be maintained at all times to ensure all geometric computations over the defined geometric entities. Of course, when a remeshing procedure is performed, the mesh is changed and the sets defining each geometric entity have to evolve to new valid sets. Therefore, the remeshing procedure must take into account the local data structure of the nodes and elements involved in each remeshing operation.

Local remeshing operations are generally employed by mesh adaptation techniques in order to improve the local overall mesh quality Q of an element patch computed over different criteria (shape, volume, mesh size...). Generally, the element patch is replaced by another element patch with better quality. Currently, there are two ways of implementing a remeshing procedure: the first implements a so called *star-connection* algorithm, which performs several attempts of reconnecting a local subset of elements until its minimum quality criteria is achieved [184]. The second uses the separate definition of local remeshing operations (vertex smoothing, node collapsing, edge splitting, edge swapping....) to be

performed on the mesh, depending on its local requirements (edge size, element shape...) [199, 200] allowing to have more control over the operations performed on the mesh but penalizing the variety of solutions to a given triangulation and in general, being more complex in terms of programming. We have opted for the latter because our main interest is to have control over the local topology at a given point, hence, the purpose of this section is to introduce some remeshing operations along with their restrictions to some of the local geometries defined on the triangular grid. Here the notion of mesh quality Q will be computed as a factor of the shape and the size of the elements using the same approach as in [184], however, it has to be noted that the selective remeshing procedure is not only driven by the local overall mesh quality Q of an element patch but also by the local topological degree of the nodes of the patch.

Selective node collapsing

Node collapsing is one of the most used algorithms when attempting to coarsen a mesh. Even when using the *star-connection* algorithm, the majority of iterations performed are equivalent to node collapses within an element patch. A Node collapse is performed when two connected nodes are below a certain distance δ_c the one from the other. The value of δ_c that triggers a node collapse is usually a percentage of the local mesh size desired. Figure 3.8 illustrates the mechanism of a node collapse performed to node N_4 over node N_1 (N_1 collapses N_4), here elements E_{11} and E_{12} are deleted.

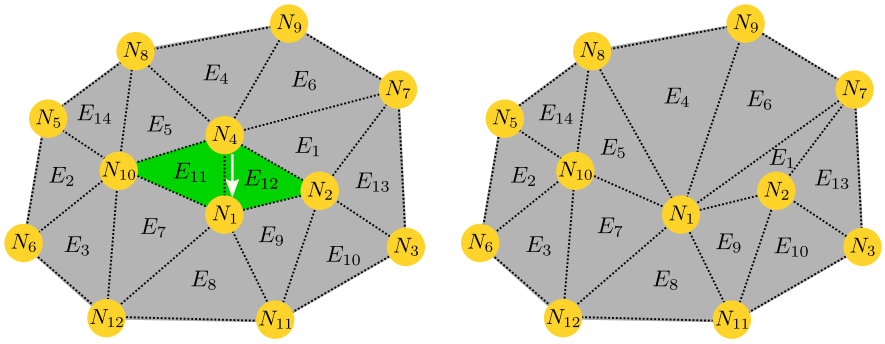


Figure 3.8: Example of node collapsing. a) initial state, elements to disappear are colored green b) state after collapsing

In our TRM model, node collapse should be avoided when the operation can not determinate to which geometric entity the remaining node will belong. To do this, the operation is performed based on the topological degree hierarchy T explained in section 3.2.1:

$$\begin{cases} T(N) = 0 & \forall \quad N/Type(N) = PNode \\ T(N) = 1 & \forall \quad N/Type(N) = LNode \\ T(N) = 2 & \forall \quad N/Type(N) = SNode \end{cases}$$

where N is a given node and $T(N)$ is the topological degree of N. Node collapse is then allowed to be performed to N_j over N_i (N_i collapses N_j) if $T(N_j) \geq T(N_i)$, meaning that P-Nodes can collapse all type of nodes, L-Nodes can only collapse S-Nodes and L-Nodes, and S-Nodes can only collapse S-Nodes. In all cases, the remaining node maintains its own degree and its own coupled geometric entity.

Additionally, two more conditions have to be met when collapsing L-Nodes: two connected L-Nodes can collapse if they belong to the same geometric line (see figure 3.9) and if they are consecutive within the line (see figure 3.10). Similarly, if a L-Node N_j connected to a P-Node N_i, N_i can collapse N_j if the coupled point of N_i appears as a limit point on the coupled line of N_j and if they are consecutive within the line.

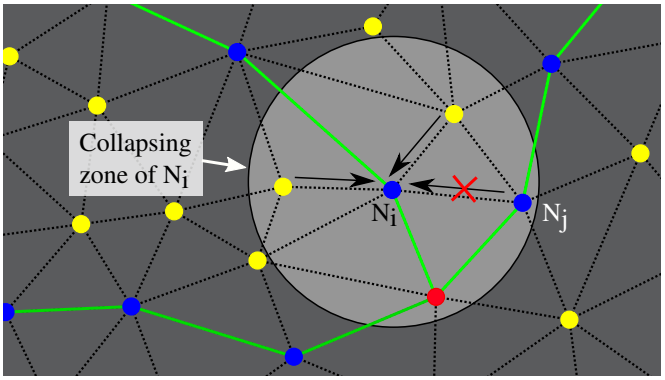


Figure 3.9: Node Collapsing. Some nodes are within the collapsing zone of N_i: Two S-Nodes (yellow) will collapse, one L-Node (blue) cannot collapse and one P-Node (red) cannot collapse. N_i cannot collapse P-Nodes (topological degree), N_i can collapse L-Nodes but N_j does not belong to the same line.

Selective edge splitting

Contrary to node collapsing, edge splitting is often used on the procedure of refining a mesh, it is performed by the insertion of a new node, the creation of some new elements and the re-connection of the element patch involved after the insertion. Similarly to the node collapse procedure, edge splitting is often driven by the size of the edges, splitting all edges having a distance greater than a predefined parameter δ_s. Figure 3.11 shows one example of an edge splitting performed in the edge $\overline{N_2N_{10}}$, the element patch involved before and after the

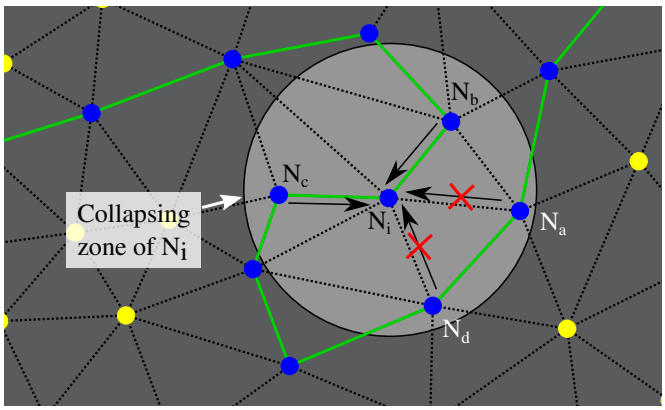


Figure 3.10: Node Collapsing. Some nodes are within the collapsing zone of N_i: four L-Nodes (blue) N_a, N_b, N_c and N_d. Only Nodes N_b and N_c can collapse as they are consecutive to N_i within the same line.

splitting is colored green.

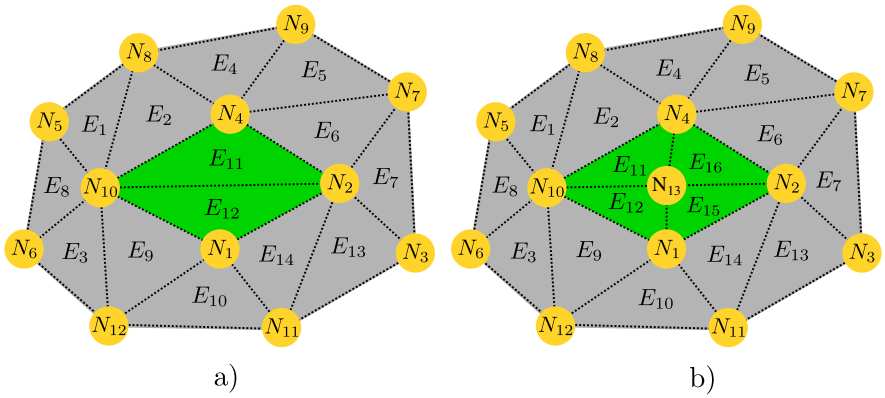
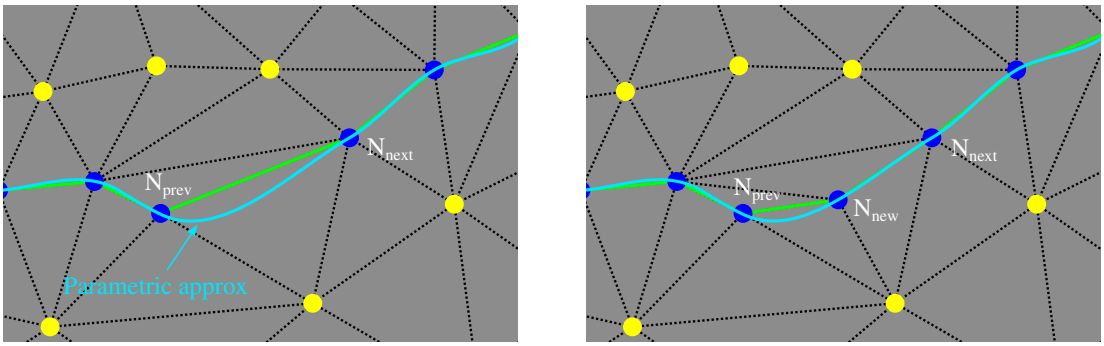


Figure 3.11: Example of edge splitting. a) initial state, the edge $\overline{N_2N_{10}}$ will be split, elements involved in the procedure are colored green b) state after splitting.

Edge splitting is allowed for every edge on the mesh on our TRM model but having certain conditions: in order to maintain a given coherence on the description of the geometric entities defined on the mesh, the inserted node has to present a coherent type and to be added to the right geometric entity. Three possible cases are considered:

- when splitting a connected P-P edge (an edge with two P-Nodes as vertices where each point appear on the list of connections of the other point as presented in section 3.2.1) the inserted node N_{new} will be of type $Type(N_{new}) = LNode$ and a line must be created, the line will have the two points from the two P-Nodes as limit points and one L-Node (N_{new}) on its structure. N_{new} will be inserted at the center of the edge.

- When splitting a P-L edge (an edge with one P-Node and one L-Nodes as vertices) or a L-L edge (an edge with two L-Nodes as vertices) that are connected[3] on one geometric line, the inserted node N_{new} will be of type $Type(N_{new}) = LNode$ and will be inserted at its corresponding position on the line data structure. Additionally, N_{new} will be inserted at the midpoint[4] of the parametric approximation of the line between its previous and next nodes (L-Node or P-Node) on the line. An example of this case is shown in Fig. 3.12
- For every other case, the inserted node N_{new} will be of type $Type(N_{new}) = SNode$ and will be inserted on the corresponding geometric surface at the center of the edge.



a) b)

Figure 3.12: Example of edge splitting of a L-L edge within a geometric line, the connections within the geometric line are displayed green, the parametric approximation of the line is displayed in cyan. a) initial state, the edge $\overline{N_{prev}N_{next}}$ will be split, b) state after splitting, Node N_{new} is placed at the midpoint of the parametric approximation of the line between nodes N_{prev} and N_{next}

Selective edge swapping

Edge swapping is another remeshing operation that changes the connectivity of the mesh but contrary to node collapsing and edge splitting, it does not insert nor deletes a node. This operation is very useful when the mesh is subjected to large displacements or shearing, as it allows to unblock very distorted meshes [172, 171].

In the presented TRM model, edge swapping is forbidden wherever it destroys any boundary connection (This is for example if two L-Nodes belong to the same line and are consecutive).

[3]Connected nodes do not mean that they share an element on the finite element mesh but that they belong to the same line and are consecutive within the line, see Figures 3.9 and 3.10 for examples of non-connected nodes

[4]Here the *midpoint* corresponds to the scalar mean value of the parametric variable between the next and previous node.

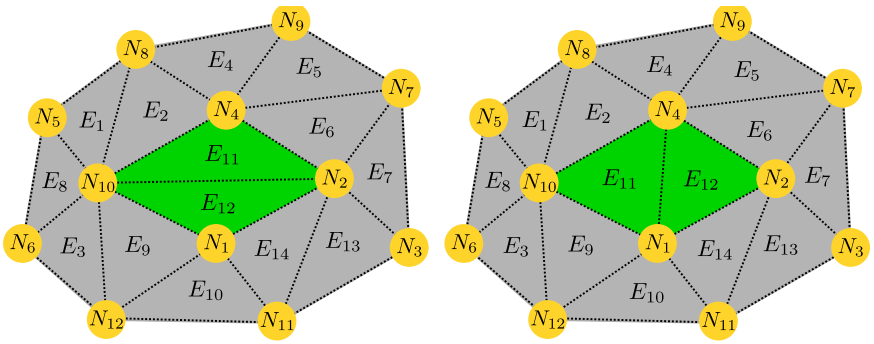


Figure 3.13: Example of edge swapping. a) initial state, the edge $\overline{N_2N_{10}}$ will swap, elements involved in the procedure are colored green b) state after swapping

Selective vertex smoothing

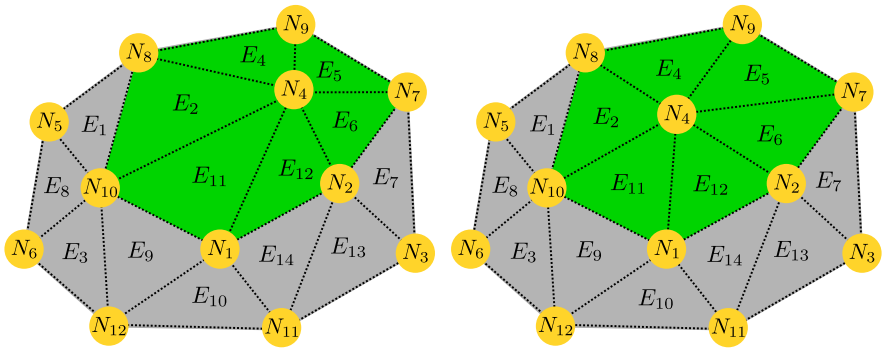


Figure 3.14: Example of vertex smoothing. a) initial state, the node N_4 will be smoothed, elements involved in the procedure are colored green b) state after smoothing

Vertex smoothing is the most simple remeshing operator, it consists in moving nodes to a location where the quality Q of the local patch is improved (usually taken as the patch barycenter). Figure 3.14 shows one example of a vertex smoothing operation performed over node N_4, the local patch involved in the operation is colored green. This example illustrates how this simple operation can dramatically improve the mesh quality shape of the elements involved.

Of course, vertex smoothing in our procedure is not allowed on nodes N_i with a degree $T(N_i) \leq 2$ as this procedure would change the position of points and lines, reducing the precision of the upcoming geometric computations. A variant of the vertex smoothing operator called *selective vertex gliding* will be introduced further, to be performed on L-Nodes allowing to have a better discretization on domain boundaries.

Selective vertex gliding

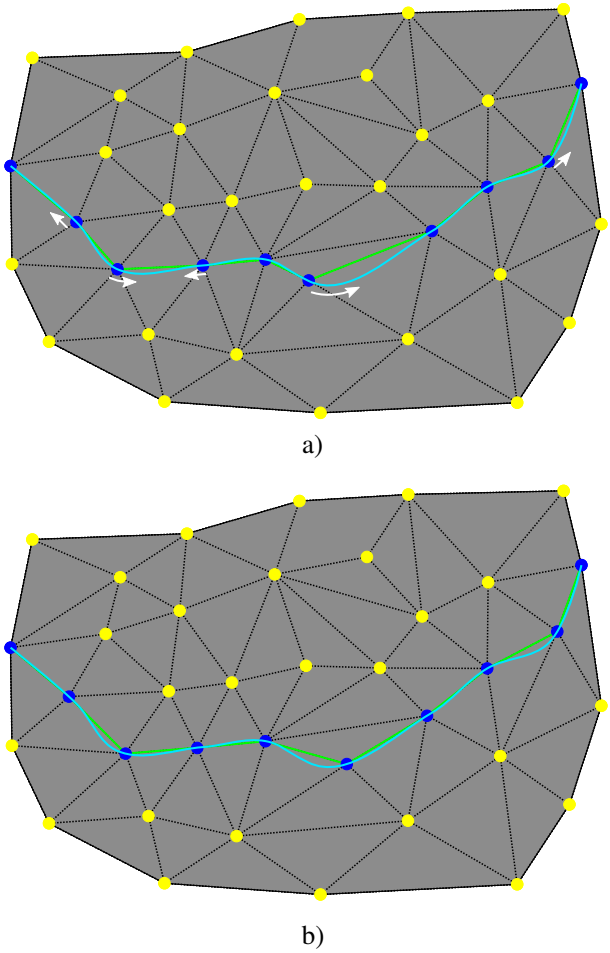


Figure 3.15: Example of vertex gliding when performed on some of the L-Nodes (blue) of a given line using its parametric approximation (cyan), a) initial state and upcoming gliding (white arrows), b) final state after gliding.

Here a new operator is introduced in order to improve the discretization of geometric lines. This operator "glides" L-Nodes over the parametric approximation of the line to the *midpoint* between its previous and next nodes (L-Node or P-Node) on the line. The intention behind this operator is to allow nodes to be equidistant on a line, reducing instabilities and ensuring the same precision over each parametric segment. The choice of gliding each node using the midpoint, instead of redistributing all nodes over the line at the same time enables to reduce the possible flipping of some of the elements of the boundary in which case the operation is discarded.

Figure 3.15 shows an example of vertex gliding when performed on some of the L-Nodes (blue) of a given line using its parametric approximation (cyan).

3.2.5 Lagrangian movement

Once a velocity has been computed on the mesh, whether it is issued from a geometric property obtained with the help of the interface approximation explained in section 3.2.3 or not, each node N_i of the mesh is allowed to be moved to a new position $\vec{r_i}$ in a Lagrangian way using the velocity vector field $\vec{v}$ and a given time step dt.

$$\vec{r_i} = \vec{r_i^0} + \vec{v_i} \cdot dt. \tag{3.6}$$

where $\vec{r_i^0}$ is the position of the node N_i before its displacement.

Mesh conformity (in a FE sense) must be ensured at all times in our TRM model, this is why it is necessary to check every movement to avoid flipping on the concerned element patch. Figure 3.16 shows one example of element flipping on Element E_4 when node N_4 is moved to a location outside its element patch.

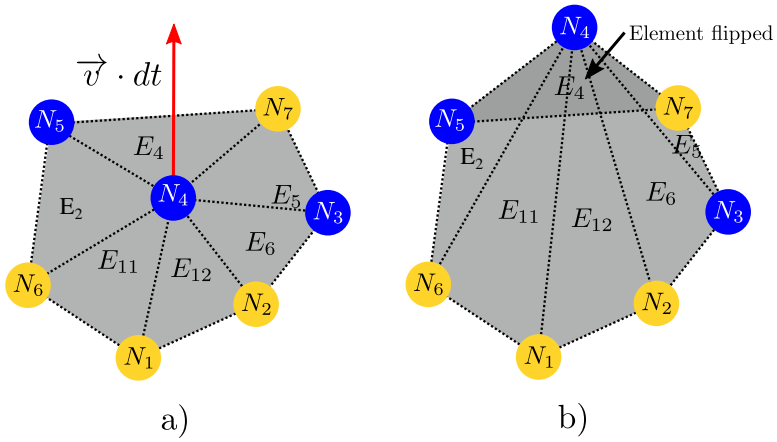


Figure 3.16: Example of element flipping, a) initial state, the displacement vector of N_4 $\vec{v} \cdot dt$ lies outside the element patch. b) state after updating the position of N_4, element E_4 has been flipped.

Even though this behavior is very rarely encountered by the proposed TRM model with standard parameters, a very big dt or very small mesh sizes could cause a more regular appearance of this mechanism. When a flipping is encountered by the TRM model, it responds by iteratively halving the magnitude $|\vec{v}|$ and checking if the movement still produces a flipping. When a valid configuration is obtained the movement is saved and the algorithm continues to the next node. Even if this procedure changes the behavior of the Lagrangian movement, it allows not to damage the mesh while moving. Moreover, as the corrections are made locally and for very few displacements, precision should not be highly influenced.

3.3 Grain growth modeling with the TRM model

As mentioned in chapter 1, the simulation of microstructural evolutions are given by the addition of complex and different mechanisms as GG [5, 167, 201, 168, 169], ReX [5, 154, 140, 146, 6, 7, 142] or ZP [98, 10, 11, 12]. Here we will consider only GG in order to test and compare the TRM model to other more classical approaches.

During grain growth the velocity $\vec{v}$ at every point on the interfaces can be approximated by Eq. 1.5. In this chapter, isotropic conditions will be considered: the mobility M is only dependent on the temperature and the GB energy γ is constant.

In this section, two numerical models will be taken into account for the modeling of GG by capillarity using Eq. 1.5, the first will use the implicit description and evolution of grain boundaries by the resolution of PDEs over LS fields in an FE context [140, 167, 168], and the other using the presented TRM model composed by the presented formalisms: the data structure and preprocessor of sections 3.2.1 and 3.2.2, the approach for the computation of geometric properties of section 3.2.3, the remeshing strategy of section 3.2.4 and the Lagrangian description and evolution of grain boundaries of section 3.2.5.

3.3.1 Topologic remeshing and Lagrangian movement approach

Here the main procedure for the modeling of GG using the TRM model will be presented, nonetheless, once again, the presented algorithm can be modified in order to simulate other different multi-domain problems by changing the velocity equation used in this context. Multiple topological changes on the polycrystal structure (grain disappearance and quadruple point dissociation) usually encountered on GG will be taken into account on the modeling algorithm.

Velocity computation and interface migration

After obtaining a valid data structure for the whole polycrystal topology using the procedure explained in section 3.2.2 the model is ready to initiate the simulation of the migration of GBs. Initially, the algorithm performs the calculation of the local curvature κ and the local normal vector $\vec{n}$ with the help of the approximation by splines explained in section 3.2.3. These geometrical values can be used on Eq. 1.5 to obtain a value for the local velocity vector $\vec{v}$ for every node of the interface, i.e. for every L-Node in our structure.

Note that the velocity at multiple junctions (P-Nodes) can not be deduced from Eq. 1.5 as the curvature and normal at these points can not be mathemati-

cally computed. An alternative approach is needed: **Model II** of [14], where the product $\kappa\vec{n}$ is obtained from an approximation of the free energy equation of the whole system in a vertex context.

Boundary conditions are also modeled through the use of **Model II**. We do so by changing the status of every L-Node found in the domain boundary to a P-Node. Then, every P-Node of the boundary creates a "virtual" connection (a connection to take into account in **Model II**) to the two adjacent nodes also belonging to the domain boundary. Finally, only the tangent[5] component of the velocity $\vec{v}$ obtained for these P-Nodes is used in the movement.

With the local velocity $\vec{v}$ obtained on the GB, it is possible to move the GB network using the procedure explained in section 3.2.5 over every node at the interface.

remeshing strategy

Till now, the steps explained in section 3.3.1 can be done iteratively to try to obtain the general behavior of the GBM, however, after a few iterations, the mesh could get stained as only the nodes of the interfaces are moving (the element flipping prevention will eventually discard all displacements). Hence there is a need for a general remeshing strategy that improves the mesh quality and prevents a mesh stagnation to happen. This remeshing strategy not only has to change the mesh structure of S-Nodes, but also L-Nodes and P-Nodes. For this, all the selective remeshing operators presented in section 3.2.4 will be used:

Node collapse and edge splitting: edge size control

If we consider for example a grain shrinking, interfaces (lines) given for that grain will decrease their length until they completely disappear, nodes (L-Nodes and P-Nodes) on the interface need to be gradually remeshed (collapsed) in order to allow the interface length diminution. Consider now the opposite case, where a grain is growing, some interfaces of that grain are going to increase their length and in order to maintain a certain accuracy on its approximation, it will be needed to insert some nodes in them. This operation can be made via the edge splitting remeshing operation performed on an edge of the line. Now, not only interfaces need to be remeshed, also volumes are shrinking and expanding, hence some remeshing operations are needed within the grains too. Node collapsing and edge splitting will be also available on these regions to control the size of the edges.

[5]In this context, the tangent component refers to the tangent of the domain limits and not to the tangent of the GB intersecting the domain boundaries.

A global node collapse strategy is performed over all nodes driven by a collapsing distance field δ_c which we have chosen to be dependent on the degree of the nodes involved on the collapsing. Note that when a node collapsing is performed on one node N_i, this node will query all its neighbors nodes N_j for their distance d_{ij} to N_i ($|\overline{N_iN_j}|$), if this distance $d_{ij} > \delta_c(N_i, N_j)$ and the collapsing is able to be done (see conditions in section 3.2.4) the operation is performed. In our model we have chosen to make some relations for the values of δ_c:

$$\left\{\begin{array}{rcl} \delta_c(N_i,N_j) = d_{min} & \forall & N_i \,/\, Type(N_i) = PNode \\ \delta_c(N_i,N_j) = d_{min} & \forall & (N_i,N_j) \,/\, Type(N_i) = LNode,\, Type(N_j) = SNode \\ \delta_c(N_i,N_j) = 3d_{min} & \forall & (N_i,N_j) \,/\, Type(N_i) = LNode,\, Type(N_j) = LNode \\ \delta_c(N_i,N_j) = 6d_{min} & \forall & (N_i,N_j) \,/\, Type(N_i) = SNode,\, Type(N_j) = SNode \end{array}\right\} \tag{3.7}$$

where d_{min} is a user defined parameter defining the global minimum edge size. The chosen relative values allow to maintain the minimum length of the edges depending on where it is located in the geometrical structure, i.e. the minimum allowed length of edges composed by only S-Nodes will be 2 times larger than for edges composed by only L-Nodes, enabling a denser discretization at the domain boundaries.

A global edge splitting strategy has also been implemented, driven by a splitting coefficient δ_s defined on the topological degree of the nodes N_i and N_j of the edge:

$$\left\{\begin{array}{rcl} \delta_s(N_i,N_j) = d_{max} & \forall & (N_i,N_j) \,/\, Type(N_i) = PNode,\, Type(N_j) = PNode \\ \delta_s(N_i,N_j) = d_{max} & \forall & (N_i,N_j) \,/\, Type(N_i) = PNode,\, Type(N_j) = LNode \\ \delta_s(N_i,N_j) = 3d_{max}/2 & \forall & (N_i,N_j) \,/\, Type(N_i) = PNode,\, Type(N_j) = SNode \\ \delta_s(N_i,N_j) = d_{max} & \forall & (N_i,N_j) \,/\, Type(N_i) = LNode,\, Type(N_j) = LNode \\ \delta_s(N_i,N_j) = 3d_{max}/2 & \forall & (N_i,N_j) \,/\, Type(N_i) = LNode,\, Type(N_j) = SNode \\ \delta_s(N_i,N_j) = 3d_{max} & \forall & (N_i,N_j) \,/\, Type(N_i) = SNode,\, Type(N_j) = SNode \end{array}\right\} \tag{3.8}$$

where d_{max} is also a user defined parameter, defining the maximum distance between nodes on the edges of the interfaces.

Finally, a global relation between d_{min} and d_{max} has been taken into account as $h_{trm} = d_{min} = d_{max}/6$ in order to pilot the mesh size at the interfaces with a single parameter h_{trm}, being the minimum distance between nodes at the interface in the tangent direction.

Vertex smoothing, vertex sliding and edge swapping: shape quality control

Shape quality is not very important for the approximation of interfaces, however, it is important for their movement as the flatter the elements of the interface in

their normal direction, the higher the risk of stagnation (see Fig. 3.16). Vertex smoothing, vertex sliding and edge swapping are performed to address this problem: a global vertex smoothing procedure is first performed on every S-Node of the mesh in order to homogenize the mesh triangulation, then, vertex sliding is performed on the L-Nodes of the mesh and finally, a global edge swapping operator is performed for all elements with a shape quality $Q_s < q_s$ where q_s is a user defined parameter, the operation is performed over the edges of the elements one at a time, checking if the quality Q_{mean} of the element patch of that edge (see figure 3.13) is improved, if it is, the operation is performed and the algorithm continues to the next edge.

The final remeshing algorithm performed in our TRM model can be summarized as:

Algorithm 3 TRM Remeshing algorithm [20]

for all Nodes : N_i **do**
  for all Neighbors of N_i : N_j **do**
    if $\delta_c(N_i, N_j) < |\overline{N_i N_j}|$ **then**
      selective node collapse : $N_j \rightarrow N_i$
for all S-Nodes : SN_i **do**
  selective vertex smoothing : SN_i
for all L-Nodes : LN_i **do**
  selective vertex gliding : LN_i
for all Edges : $\{N_i, N_j\}_{j>i}$ **do**
  if $\delta_s(N_i, N_j) > |\overline{N_i N_j}|$ **then**
    selective edge splitting : N_i, N_j
for all Elements with $Q_s < q_s$: E_i **do**
  for all Edges of E_i: $\{N_j, N_k\}_{k>j}$ **do**
    if quality $Q_{mean}(N_j, N_k)$ will improve by swapping **then**
      selective edge swapping : $\{N_j, N_k\}$

Grain Disappearance

Grain disappearance is a product of multiple topological changes on the polycrystal structure where complete interfaces from a grain successively collapse into multiple junctions, eventually reducing the volume of the grain to a single point. In the presented TRM model, this topological event is automatically handled by the node collapsing strategy of the remeshing algorithm, note that P-Nodes have a certain predominance over all other types of nodes when using the selective node collapsing algorithm of section 3.2.4, meaning that always when a phase disappears a point of its interface will remain at the moment of the event.

Interface (Line) creation

At the end of a grain disappearance event, it is highly possible that a multiple junction with more than 3 connected interfaces appears. This configuration is highly unstable in our physical context as a lower energy state can be obtained by creating new interfaces and redistributing the multiple junction in several triple junctions (multiple junction with 3 connected interfaces) [19]. In the case of anisotropic GB properties, the decomposition procedure is as follows: first, the decomposition procedure can not be performed arbitrarily, an order is imposed by the values of the surface energy of interfaces and their geometrical configuration. Consider the state presented in Fig. 3.17 (a) and (b) with a multiple junction of 4 phases (ϕ_1, ϕ_2, ϕ_3, ϕ_4) with anisotropic GB properties, here the local energy state depends on the energy values, the length and the different angles between interfaces (α_1, α_2, α_3, α_4). Here the structure decomposition can occur in 2 different ways: Figure 3.17 (c) and (d) show two examples of decomposition of the quadruple junction into 2 triple junctions with different energy states. It is clear that only one of these states (the one with the lowest energy state) can occur even if both reduce the local energy state of the quadruple junction shown in Fig. 3.17 (b). The determination of the right decomposition when using anisotropic properties is a complex procedure because, in addition to the multiple possibilities, the values of the surface energy of every interface depend on multiple local microstructural variables [143]. This will be discussed further in a future chapter focused on the simulation of microstructural evolutions with anisotropic properties using the TRM model.

When using isotropic GB conditions, the decomposition is only dependent on the angles between interfaces, as the lowest energy state can be obtained by the separation of the two interfaces presenting the lowest angle between them. An example of decomposition of a quadruple point is shown in Fig. 3.18, the decomposition is done by a partial reconstruction of the element patch surrounding the multiple point. Note that this remeshing operation is different from all operations presented in section 3.2.4. In fact, this operation is not accessible by a reconnection procedure of the *star-connection* algorithm of [184]. It consists on the *splitting* of a node of the mesh and a special mesh reconnection which does not erase the preexistent connections, note that all the edges from the previous mesh (Fig. 3.18(a)), still exist in the new mesh (Fig. 3.18(b)) plus three additional edges connected to the new node.

The new P-Node is located along the line determined by the angle $\alpha_4/2$ measured from one of the two separated interface to the inner side of ϕ_4 at a distance δ_p from the initial multiple point. The value of δ_p is also a user defined parameter usually taken as $\delta_p > \delta_c$ to prevent the two P-Nodes to be collapsed by the selective node collapsing strategy in the next increment, allowing them to separate over time, reducing the total boundary energy of the GB network.

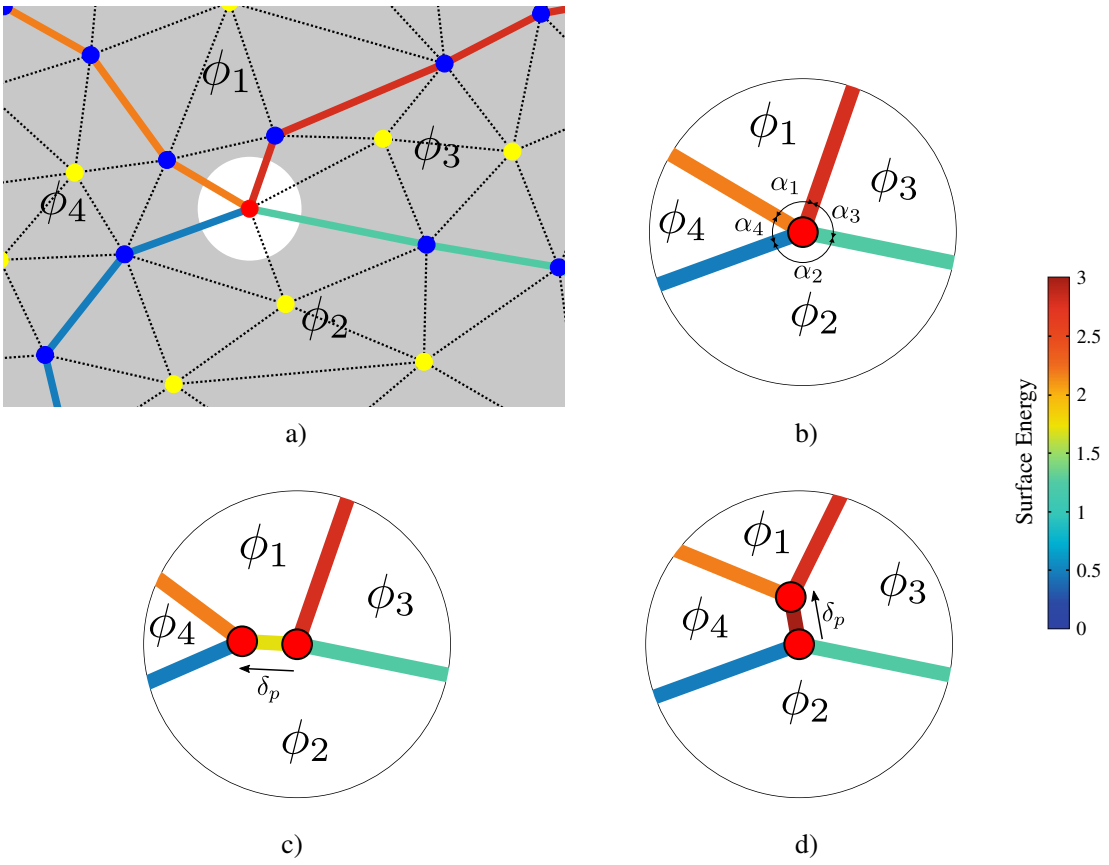


Figure 3.17: Example of a multiple junction decomposition with anisotropic energies, a) initial state. b) detailed view of the multiple junction. c) decomposition example, interfaces of ϕ_4 are separated from the initial multiple junction and a new interface is created (ϕ_1 - ϕ_2) d) decomposition example, interfaces of ϕ_1 are separated from the initial multiple junction and a new interface is created (ϕ_4 - ϕ_3)

3.3.2 The TRM algorithm for grain growth

The complete sequence for an increment of the TRM model is presented in Algorithm 4.

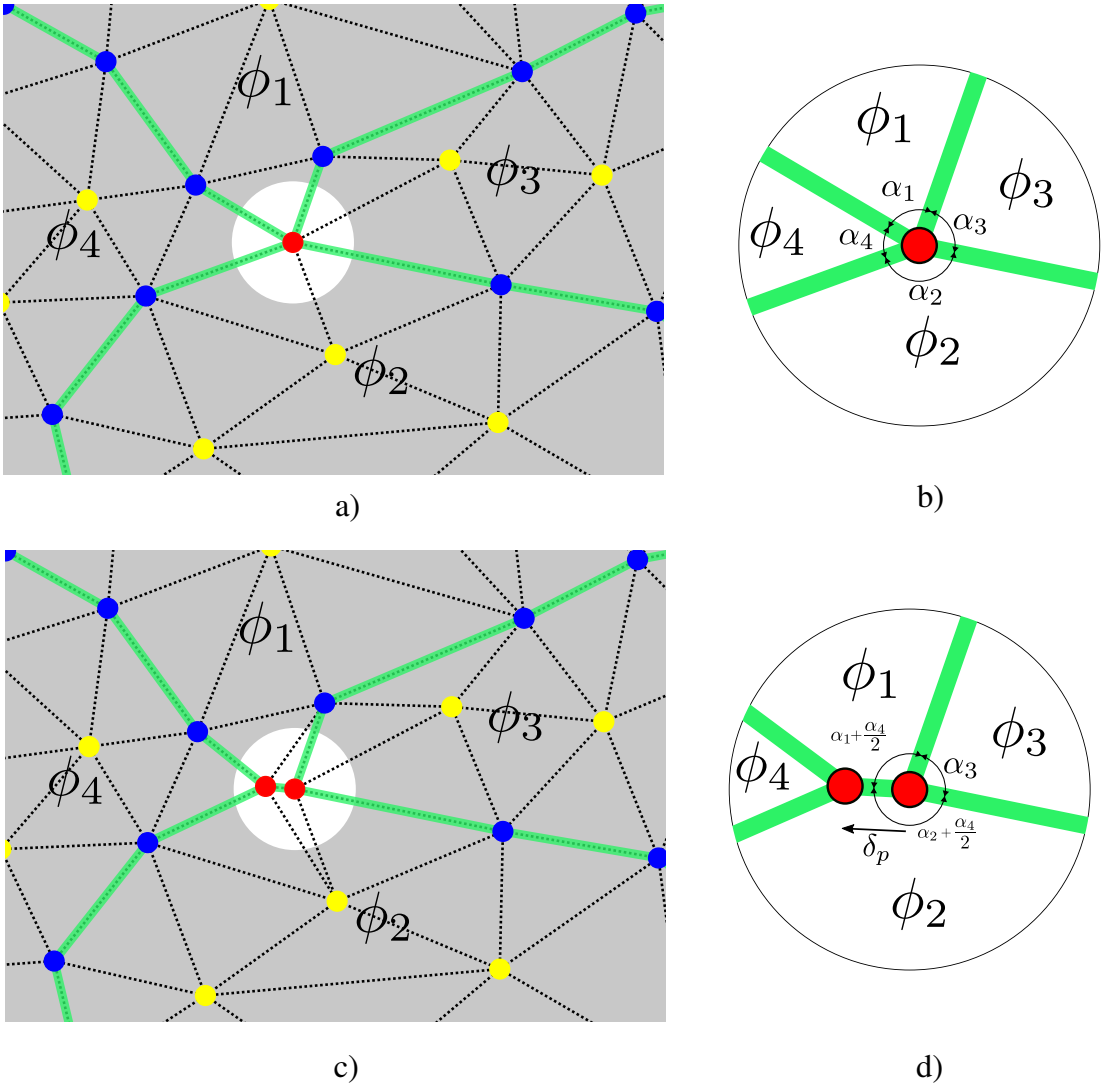


Figure 3.18: Example of a multiple junction decomposition with isotropic boundary energies, a) initial state. b) detailed view of the multiple junction. Here $\alpha_4 < \alpha_1 < \alpha_3 < \alpha_2$, the choice of the phase to detach from the multiple point is determined by the lowest angle. c) final state after the decomposition procedure, two new elements are created. d) detailed view of the decomposition, interfaces of ϕ_4 are separated from the initial multiple junction and a new interface is created (ϕ_1 - ϕ_2), the new point will be placed along the line determined by the angle $\alpha_4/2$ measured from one of the two detached interfaces to the inner side of ϕ_4 at a distance δ_p from the initial multiple point.

Algorithm 4 Isotropic Grain Growth TRM Algorithm

Perform **Remeshing Algorithm** (Algorithm 3)
for all Points: P_i **do**
    while Number of Connections > 3 **do**
        split multiple point P_i.
for all Lines : L_i **do**
    Compute the natural spline approximation of L_i.
for all L-Nodes : LN_i **do**
    Compute curvature and normal ($\kappa\vec{n}$) over LN_i.
for all P-Nodes : PN_i **do**
    Compute the product $\kappa\vec{n}$ over PN_i using model II of [14].
for all L-Nodes and P-Nodes : LPN_i **do**
    Compute velocity $\vec{v_i}$ of Node LPN_i
    Iterative movement with flipping check over LPN_i

3.4 Numerical results

A series of test cases have been realized to validate the TRM method using the same examples presented in chapter 2 for the benchmark of the LS method, results of that study will be used to compare the new TRM model to the LS approach. These benchmarks are focused mainly on the precision and the computational cost of the model, allowing to compare which approaches are more suitable for the simulation of GG under isotropic conditions.

3.4.1 Considered geometries

Correspondingly to chapter 2 the considered geometries will respond to different typical situations encountered during the GG mechanism, the first three cases being dimensionless and the last case representing a more realistic configuration. Firstly a circle shrinkage test will be used to test the accuracy and the stability of the TRM model. Next, a T-Junction configuration to evaluate the response of the model face to the evolution of an unstable triple junction. Then, a square shrinkage test to test the typical topological changes experienced during GG: grain disappearance and interface creation. Finally, one case using a 2D Laguerre-Voronoi tessellation composed of 10000 grains will be used to test the computational cost of the TRM model under typical conditions.

In chapter 2, three remeshing approaches in the context of the LS GG simulation have been tested: Static Mesh (SM), Isotropic Mesh Adaptation (IMA) and a New Fitting and Joining Algorithm (NFJA), these three approaches (SM, IMA, NFJA) will be compared to the TRM model knowing that the best approach in terms of stability, accuracy and computation time, was the one using a static mesh (see chapter 2 conclusions).

3.4.2 Circle shrinkage

Similarly to chapter 2, the circle shrinkage test will be used to observe the response of the model to the instantaneous local curvature, hence allowing to observe the stability of the interface subjected to a velocity directly proportional to the local geometric properties ($\vec{n}$ and κ).

The error to the analytical evolution of this test (see Eq. 2.14) will be computed in two ways, the first using a classical L2-Error computation over the surface of the *circular*[6] domain obtained by the TRM model, and the second using Eq. 2.18 Which express the relative error in terms of surface change per increment.

[6]In fact, the numeric *circular* domain can not define a perfect circle but the errors given by its real profile are ignored.

Note that the circle is obtained differently in the LS context (where it is defined by the positive interpolated domain of the LS field) than in the TRM model (where it is obtained by simply extracting the elements denoted by the Surface S_{circ} of the circle).

Correspondingly to chapter 2, we will compute a mean value for $\overline{\Delta s}_{\phi(h,\Delta t)}$ of $\Delta s_{\phi(h,t,\Delta t)}$ to obtain a mean value of the error $\overline{E}_{(h,\Delta t)}$ for each simulation having a different set of parameters of Δt and h. Moreover, dimensionless simulations will be considered and unitary values for the mobility M and the surface energy γ will be used. Here we will use the same initial state and dimensions used in the circle shrinkage test of chapter 2 (see Fig. 2.12).

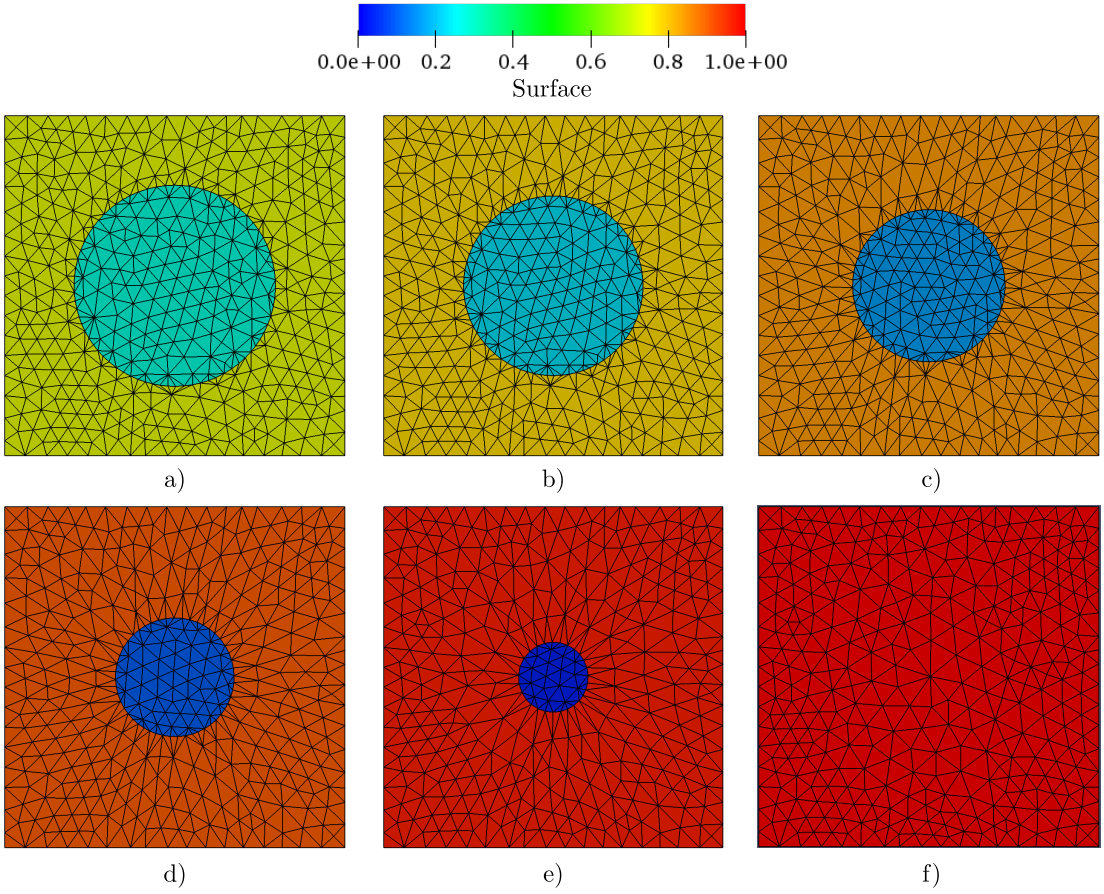


Figure 3.19: Example of the evolution of the circle shrinkage test using the TRM method and a parameter $h_{trm} = 0.006$, the mesh and the surface field per domain are displayed. a) Initial state, and b) to f) state at $t = 0.01$, $t = 0.02$, $t = 0.03$, $t = 0.04$ and $t = 0.05$.

Multiple runs with different mesh sizes h_{trm} and time steps dt were made for the TRM model. One example of the evolution of the mesh for the circle shrinkage test when using the TRM method is given in Fig. 3.19.

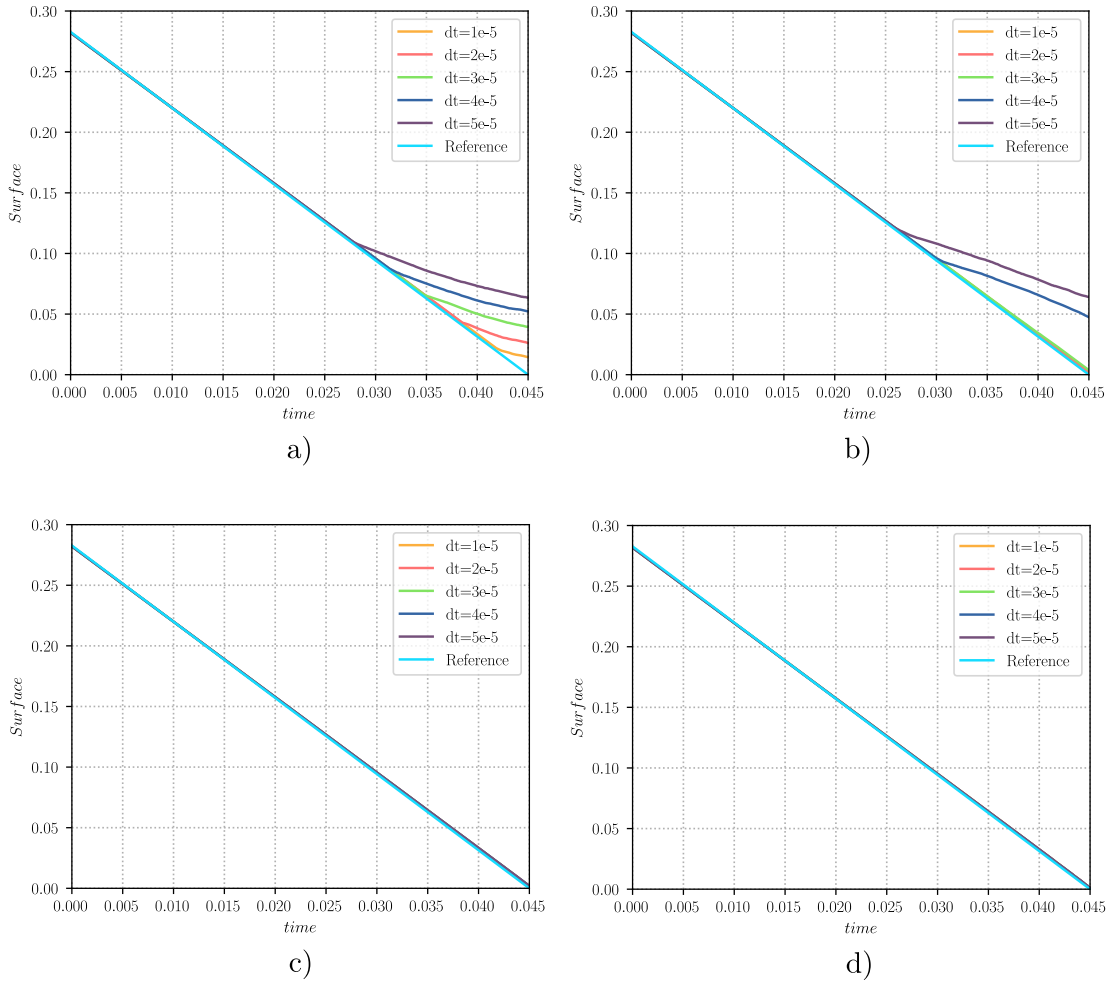


Figure 3.20: Evolution of the surface of the circle shrinkage test for the circular phase using the TRM method for different values of dt. The corresponding analytic evolution (Reference) is also shown. a) for a mesh size parameter $h_{trm} = 0.002$, b) $h_{trm} = 0.004$, c) $h_{trm} = 0.006$, d) $h_{trm} = 0.008$.

Figure 3.20 describes 4 subsets of evolutions of the circular phase, each subset for simulations with a different time step dt but maintaining a constant mesh size parameter h_{trm}. Figure 3.21 illustrates the corresponding L2-Error evolution for the 4 subsets of Figure 3.20. Figures 3.20(c-d) and 3.21 (c-d) exhibits the normal behavior of the model when subjected to normal conditions, the maximum error being not higher than 1%. Fig. 3.20(a) details the evolution of multiple simulations that eventually encountered instabilities in the velocity computation and produced a divergence from the analytical solution, Figure 3.20(b) shows the transition between the normal response (for time steps $dt = 1 \cdot 10^{-5}$ and $dt = 2 \cdot 10^{-5}$) and an unstable response (for time steps $dt = 3 \cdot 10^{-5}$, $dt = 4 \cdot 10^{-5}$ and $dt = 5 \cdot 10^{-5}$) of the TRM model. These instabilities manifest in the L2-Error plots (Fig. 3.21(a-b)) as sudden increases in the error value that go up to 16%

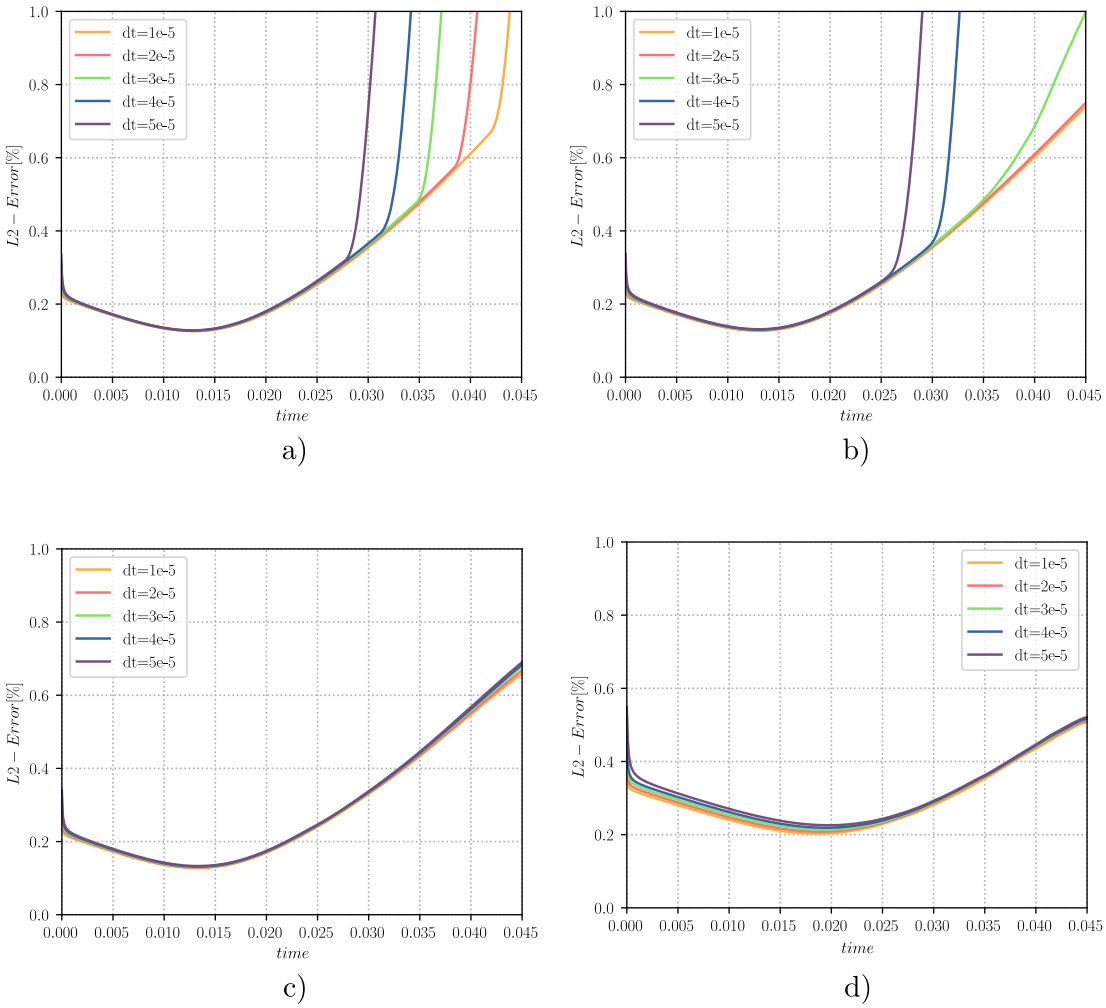


Figure 3.21: L2-Error computation for the evolution of the surface of the circle shrinkage test for the circular phase using the TRM method for different values of dt. a) for a mesh size parameter $h_{trm} = 0.002$, b) $h_{trm} = 0.004$, c) $h_{trm} = 0.006$, d) $h_{trm} = 0.008$.

in the worst case (not shown in the figures). Figure 3.22 illustrates the corresponding mean velocity $\vec{v}$ of the interface along with the analytic velocity shown as reference; stable responses show velocity values around the reference curve while unstable responses present an oscillatory global diminution on the velocity computation, this diminution is due to the fact that some of the nodes present a negative velocity value (where the direction of the velocity vector $\overrightarrow{v}$ points to the outer region of the circular phase). As mentioned in section 3.2.3 the solution over time for a velocity proportional to the curvature when using natural parametric splines as approximations could lead to instabilities when using high values of time step dt.

It is interesting to see that the obtained result for a simulation with a mesh

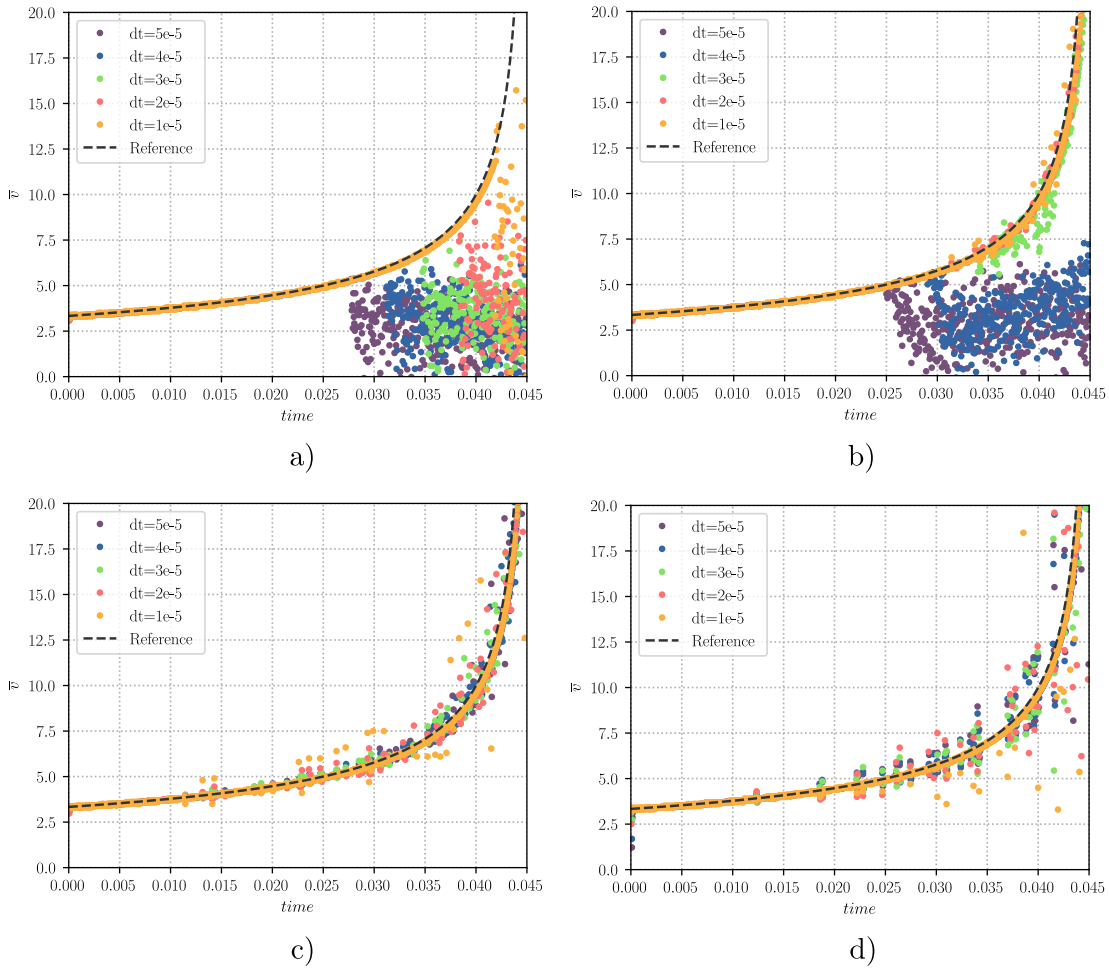


Figure 3.22: Mean velocity of the interface the circle shrinkage test using the TRM method for different values of dt. The corresponding analytic velocity (Reference) is also shown. a) for a mesh size parameter $h_{trm} = 0.002$, b) $h_{trm} = 0.004$, c) $h_{trm} = 0.006$, d) $h_{trm} = 0.008$.

size parameter $h_{trm} = 0.004$ and a time step $dt = 3 \cdot 10^{-5}$ lies exactly on the transition between the stable and the unstable region, Figures 3.20(b) and 3.21(b) show that with these values, the evolution of the surface is still very near the reference curve, obtaining an L2-Error of only 1% at the end of the simulation, while Fig. 3.22(b) shows that the evolution of the interface encountered a diminution on the computation of the mean velocity for a while (between $time = (0.032, 0.42)$) but it auto-stabilized after, hence presenting a "semi-stable" behavior.

Complementary simulations were made in order to identify the region of stability of the TRM model in function of the time step dt and the mesh size parameter h_{trm}. the results of this study are summarized in Fig. 3.23 and hereafter we will restraint the content of the present chapter to this stable region.

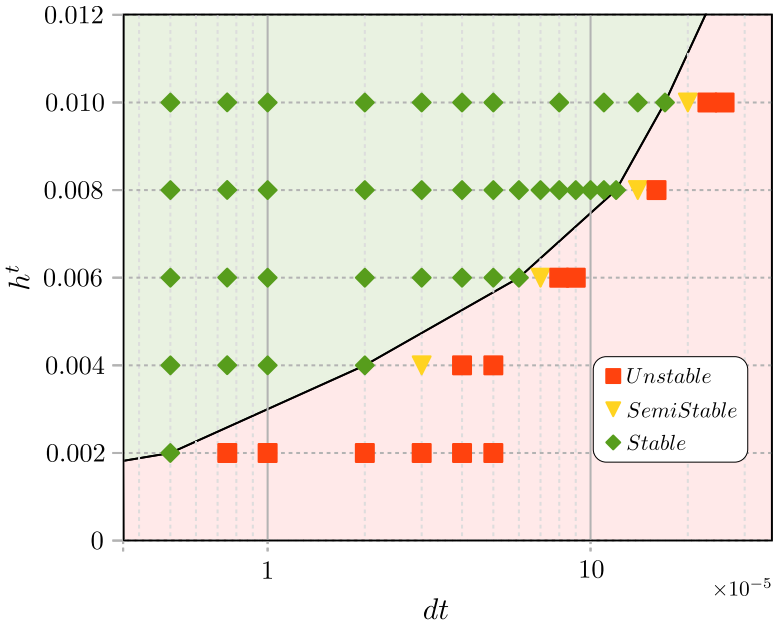


Figure 3.23: Stability of the model in function of the mesh size h_{trm} parameter and the time step dt.

Finally, Fig. 3.24 shows the different curves obtained when using Eq. 2.18 to compare the errors of the TRM method to the different approaches presented in chapter 2, the error of the TRM method for the stable responses was always $\xi_{TRM} < 2\%$ regardless of the mesh size and the time step used in the stable zone. These results are very promising as the accuracy of the TRM model in the stable region was the best.

3.4.3 T-Junction case

The T-Junction test case (see Fig. 2.20) will be used in order to observe the evolution of a triple point when using the TRM method and the so called **model II** described in [14] to compute the velocity of multiple points. Of course, results obtained here will be compared to the ones obtained in chapter 2 for the LS-FE model.

As stated in chapter 2, for the models based on the evolution of Level-Set fields on a Finite Element framework, convergence is obtained when the mesh size decreases, hence a simulation with 500000 elements (where convergence has been obtained, see Fig. 2.21) has been used as reference.

A test using the same parameters as in chapter 2 ($dt = 5 \cdot 10^{-5}$ and $h_{trm} = 0.006$) was made, the evolution of the mesh and surfaces for this test is described in Fig. 3.25. The triple point was found to be very near to the position of the reference case while the interfaces showed a similar curved shape (see Fig. 3.26) after

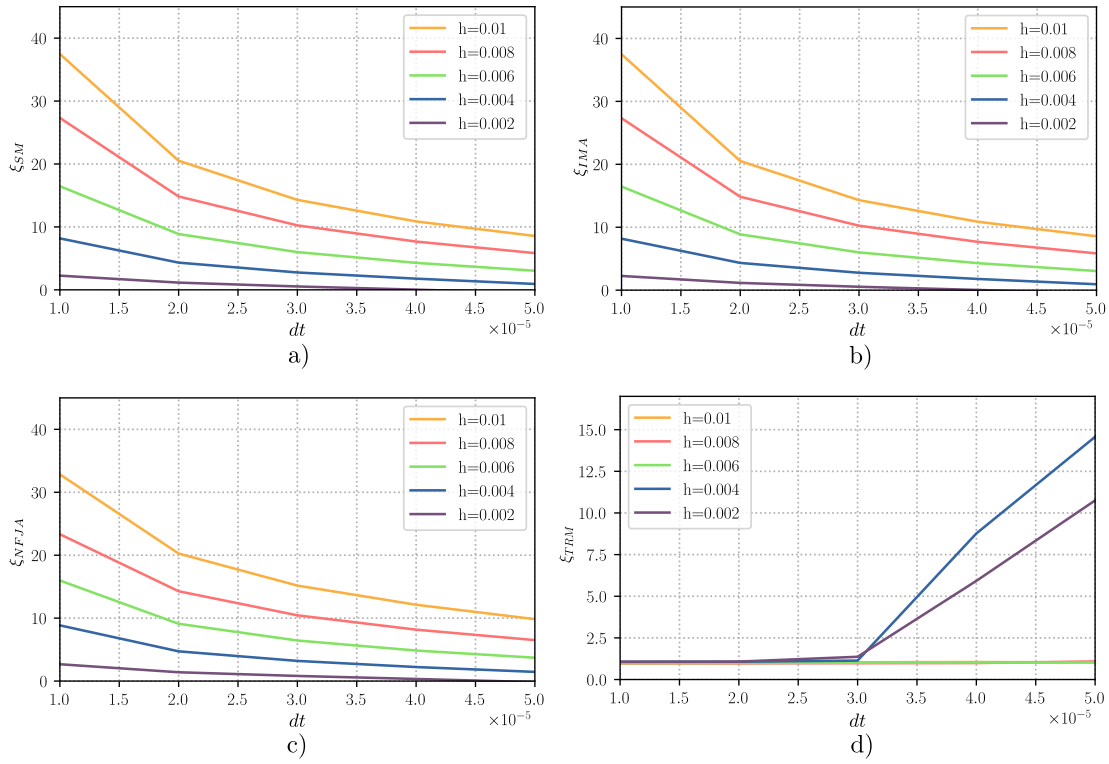


Figure 3.24: Comparisons of the error for different models for the circle shrinkage test using Eq. 2.17. a) Level-Set FE Static Mesh (SM) b) Level-Set FE Isotropic Mesh Adaptation (IMA) c) Level-Set FE New Fitting and Joining Algorithm (NFJA), d) TRM (TRM).

a time $t = 0.035$ where the quasi steady-state is already ensured. Moreover, the surface difference between the TRM model and the reference case shows an error of $\xi_{TRM} = 1.9\%$ while the other models using a LS approach obtained errors of $\xi_{IMA} = 10.1\%$ and $\xi_{NFJA} = 15.4\%$ .These results show the potential of the TRM method coupled with the explicit computation of the velocity at triple points of [14], as the results were almost the same as for the reference configuration.

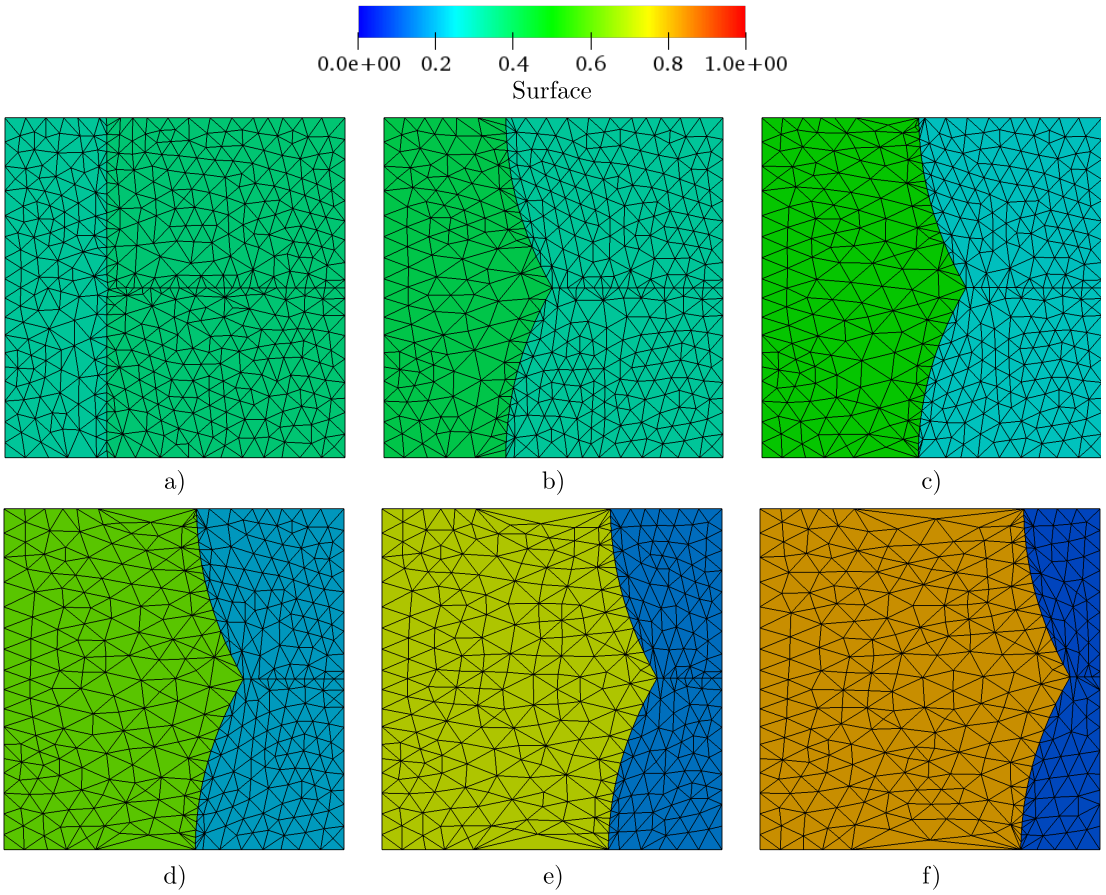


Figure 3.25: Example of the evolution of the T-junction test using the TRM method and a parameter of $h_{trm} = 0.006$, the mesh and the surface field per phase are displayed. a) Initial state, b) state at $t = 0.1$, c) state at $t = 0.2$ d) state at $t = 0.3$, e) state at $t = 0.4$, f) state at $t = 0.5$.

3.4.4 Square-Shrinkage case

As mentioned in chapter 2, the symmetry of the Squared-Shrinkage test makes it possible that the 4 triple points converge at the same place at the end of the shrinking, producing a meta-stable configuration: the quadruple point. Given this meta-stable state, the quadruple point should decompose into two triple

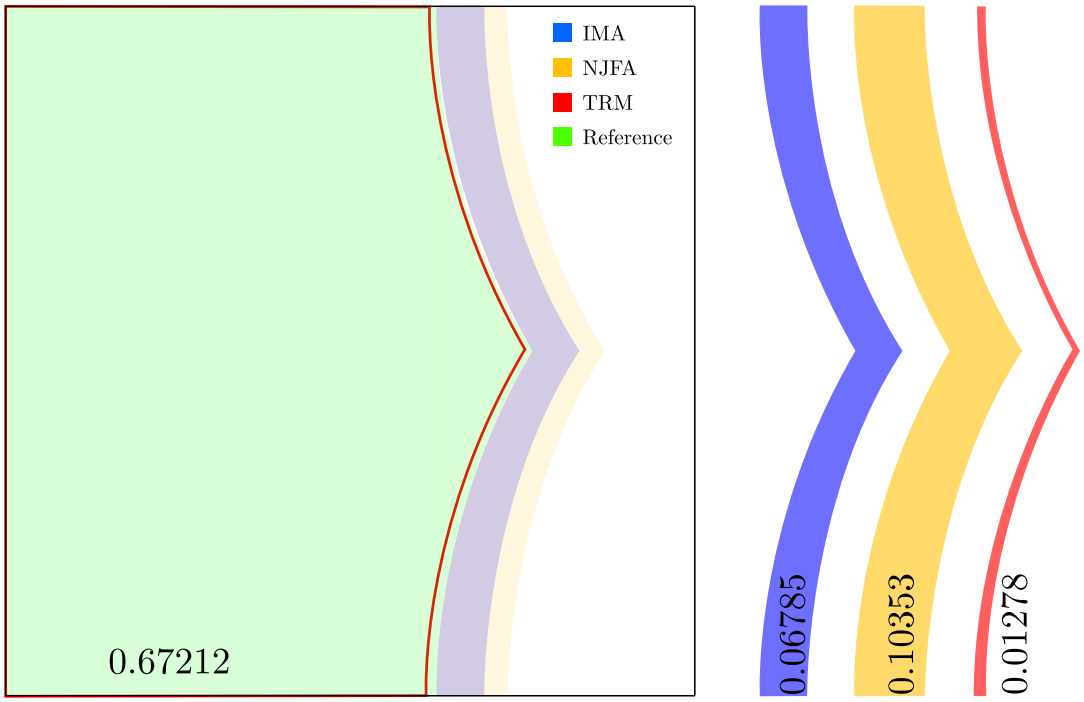


Figure 3.26: Difference between the growing phase of the T-junction test case for the reference model and the geometric difference of the same phase obtained with the other models (the IMA and NFJA) at time t= 0.35. Values for the area of each section are given.

points, reducing the total surface energy of the system (see Fig. 2.24).

As explained in section 3.3.1, the final decomposition state should be taken according to the configuration with the lowest total surface energy value, however, in the isotropic case, the choice for the decomposition is merely dependent on the angles between the interfaces at the multiple point. In the current case scenario where symmetry has been imposed at the initial state, these angles should be around $90° + \xi_{an}$ where ξ_{an} takes into account the numeric error (rounding errors, numeric precision or accuracy of the model). In the context of the TRM model, these small differences between the 4 angles are enough to decompose this meta-stable configuration into the expected two triple points configuration (either one of the two final configurations shown in Fig. 2.24 are valid).

One test for the square shrinkage case using the same parameters as in chapter 2 ($dt = 5 \cdot 10^{-5}$ and $h_{trm} = 0.006$) was made, the evolutions of the mesh and surfaces for this test are described in Fig. 3.27.

Figure 3.28 illustrates the comparison of the TRM model and the methods used in chapter 2 to the reference case (Using a LS-FE method with a static mesh composed of 2 millions elements) after t=0.05. The error in the area of the square

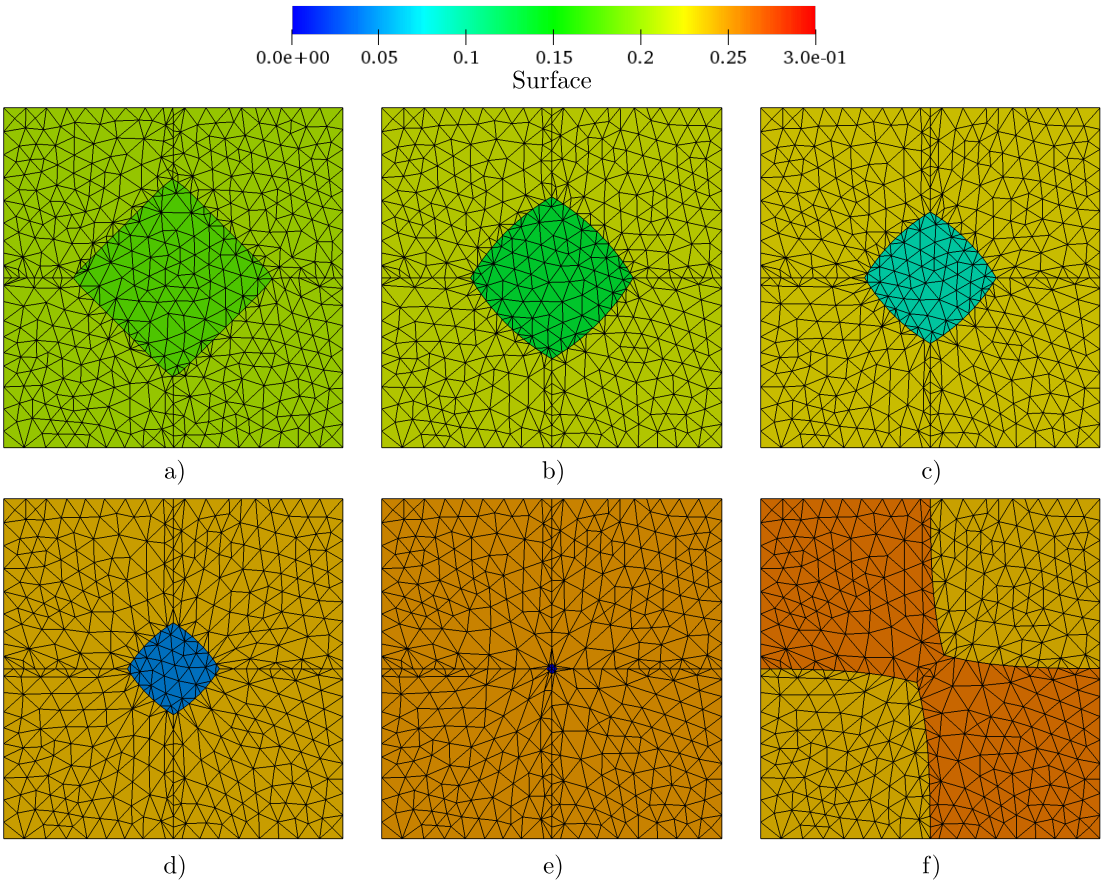


Figure 3.27: Example of the evolution of the square shrinkage test using the TRM method and a parameter of $h_{trm} = 0.006$, the mesh and the surface field per phase are displayed. a) Initial state, b) state at $t = 0.0225$, c) state at $t = 0.045$ d) state at $t = 0.0675$, e) state at $t = 0.09$, f) state at $t = 0.1125$.

shaped phase was $\xi_{TRM} = 6.6\%$ while for the other cases was $\xi_{NFJA} = 38.4\%$ and $\xi_{IMA} = 16.3\%$.

3.4.5 2D-10000 grains case

Finally, a massive multidomain test composed of 10000 initial grains is considered, corresponding to the same test (same initial topology and thermomechanical properties) of the test presented in section 2.5.5. Here, the TRM model will be compared to the Isotropic Mesh Adaptation (IMA) case and to the Static Mesh (SM) case using a mesh size of $h = 0.004\ mm$ at the interfaces, additionally, statistical comparisons with the response obtained by a FE-LS approach as presented in [5, 167, 7, 170] will be given, this approach uses a more classic method of mesh adaptation during calculations where the interfaces are captured with an anisotropic non-conform local refined mesh and it will be described as the Anisotropic Meshing Adaptation (AMA) case. Finally, the same reference case

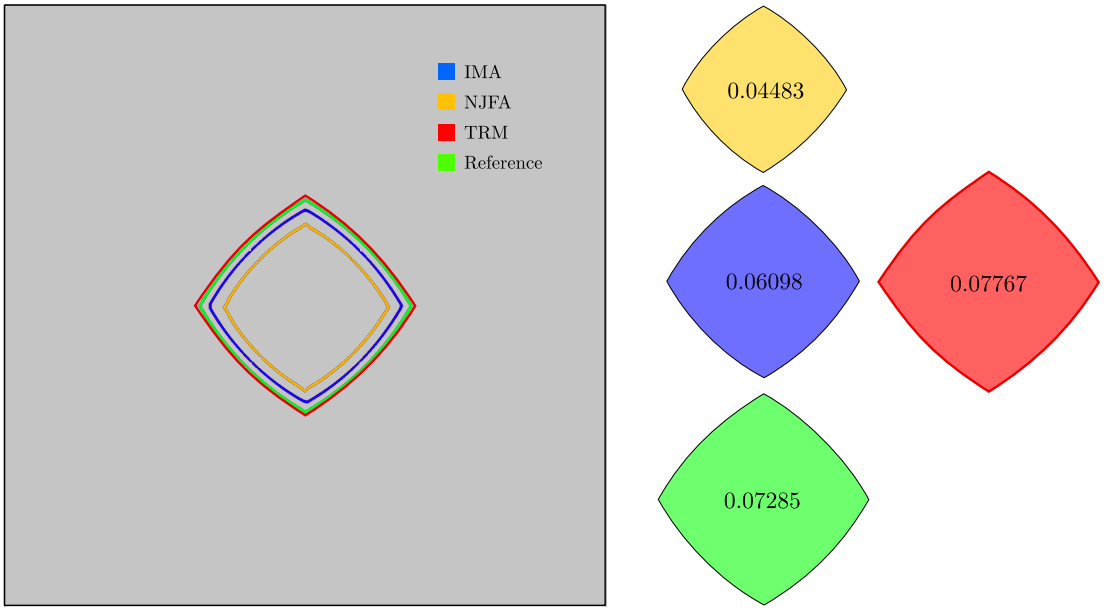


Figure 3.28: Square-Shrinkage. Comparison of the ϕ_1 phase at $t = 0.05$ (left side). Values for the area of each domain are given (right side).

of section 2.5.5 will be used in this context to compare the different models (a SM model using a mesh size of $h = 0.001\ mm$).

One test using the TRM method has been performed using a mesh size parameter $h_{trm} = 0.004\ mm$ and a constant time step $dt = 10\ s$. The evolution of the microstructure through time can be observed in Fig. 3.29, here a comparison with the SM model has been given. A detailed view of the same comparison can be found in Fig. 3.30 where a more specific comparison between grains can be made. The overall comparison shows that the grain growth phenomena occurs very similar in both cases, the morphology of the grains is very similar at the different moments during the simulation, while a very small difference in the grain sizes values (see the color distribution in Fig. 3.29(c-d)) can be observed.

A more quantitative comparison in terms of the evolution of the mean grain size and the grain size distributions can be observed in Figures 3.31 and 3.32 respectively. Here the comparisons with the AMA, the IMA and the reference cases are also displayed. These comparisons show that the speed of grain growth when using the TRM model is a little "slower" compared to the other three cases. However, its evolution is very near to the evolution of the SM case which has been the best case scenario in the comparisons made in chapter 2 to the reference case.

The CPU-time evolution over time for the different models can be found in 3.33, where the TRM model has been the one with the lowest computational cost, performing 14,8 times faster than the SM model and 156 times faster than the

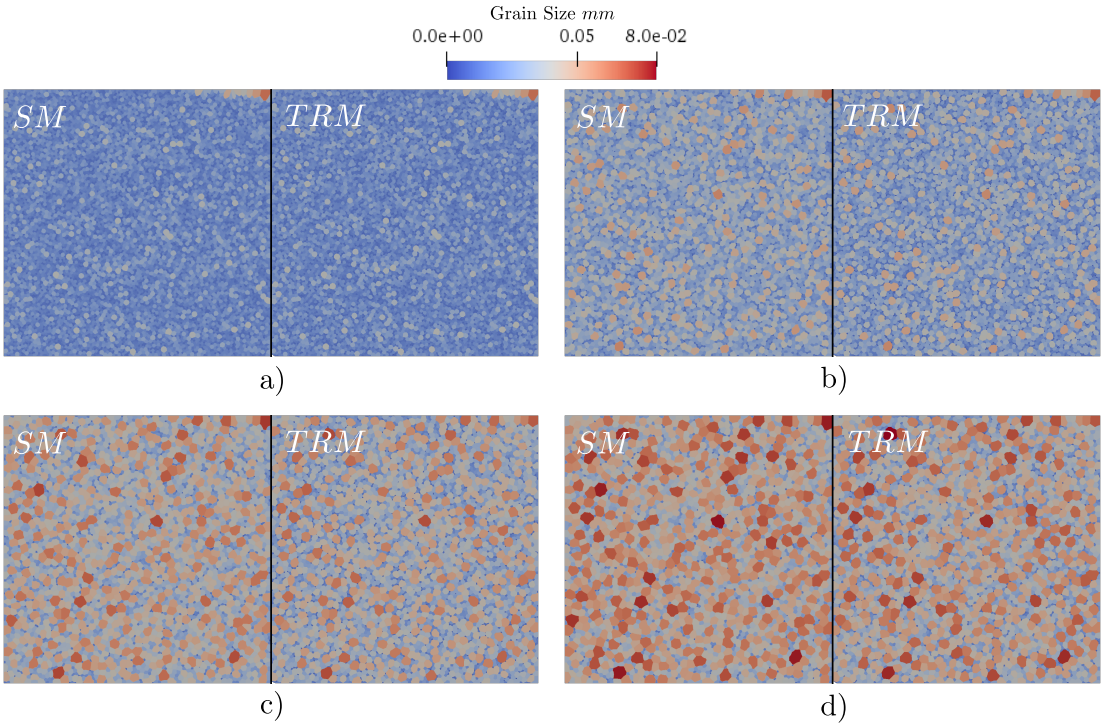


Figure 3.29: 2D 10000 grains case. Comparison of the evolution of the TRM model (right) and the SM model (left). a) Initial state b) state at $t = 840\ s$ c) state at $t = 1940\ s$ d) state at $t = 2780\ s$.

AMA case. These results are very promising as the ratio accuracy - CPU-time of the TRM model shows a significant improvement against all the other models.

Finally, the mobility term M can be fitted in order to minimize the difference between the curves of the grain size of the different models and the reference. A correction factor of -12.8% for the SM model and of +6.69% for the TRM model is necessary to minimize the error between both models and the reference response. Figures 3.34 and 3.35 shows the evolution of the grain size and the grain size distributions respectively for the SM and the TRM model after the adaptation of the mobility. These results illustrate a well known behavior of full field simulations of GG: the reduced mobility is classically impacted by the choice of the numerical method and is not only a universal physical parameter. Here the TRM model performed 13.63 times faster than the SM model to obtain a similar response.

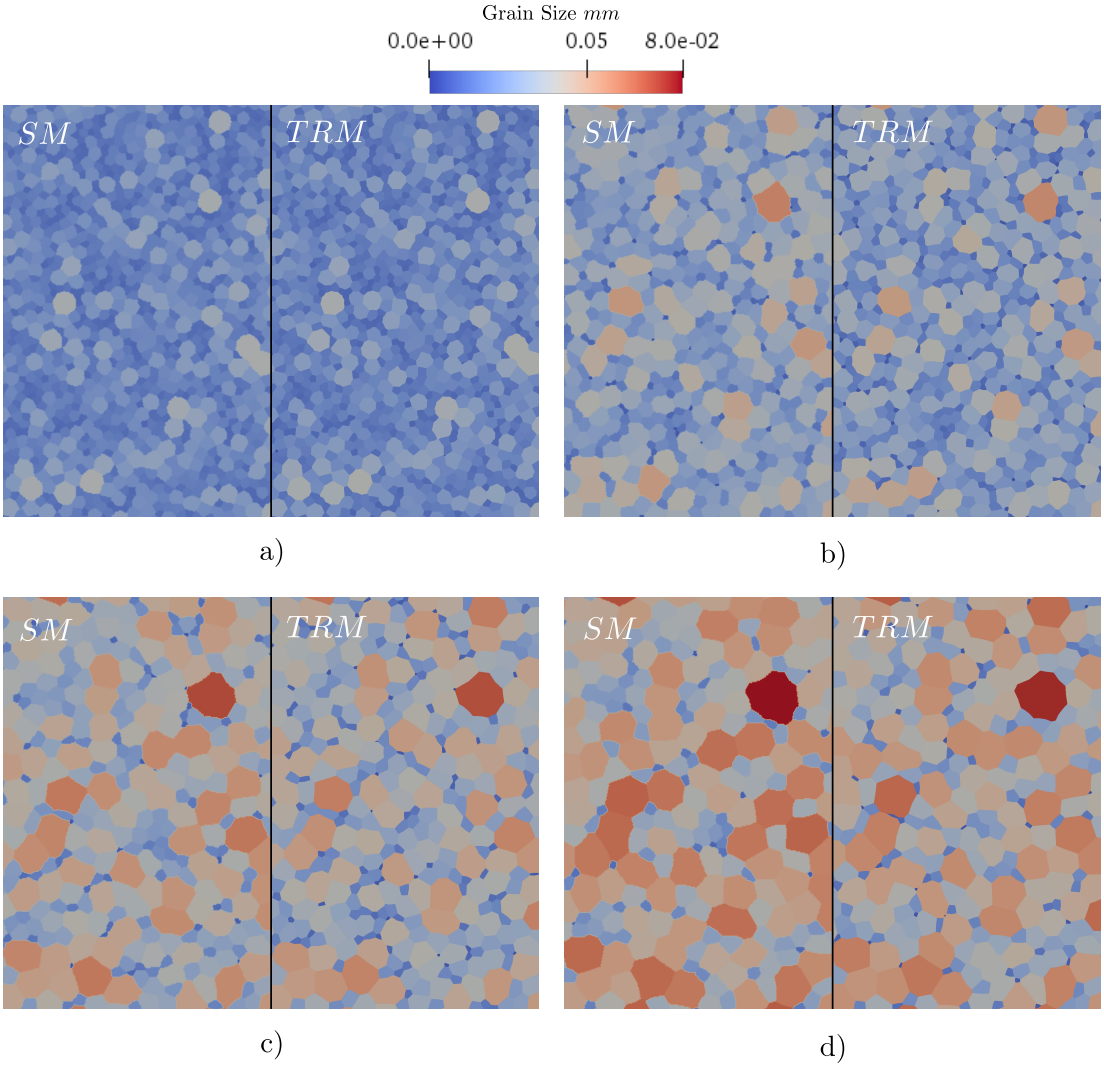


Figure 3.30: 2D 10000 grains case. detailed view of the evolution of the TRM model (right) and the SM model (left). a) Initial state b) state at $t = 840\ s$ c) state at $t = 1940\ s$ d) state at $t = 2780\ s$.

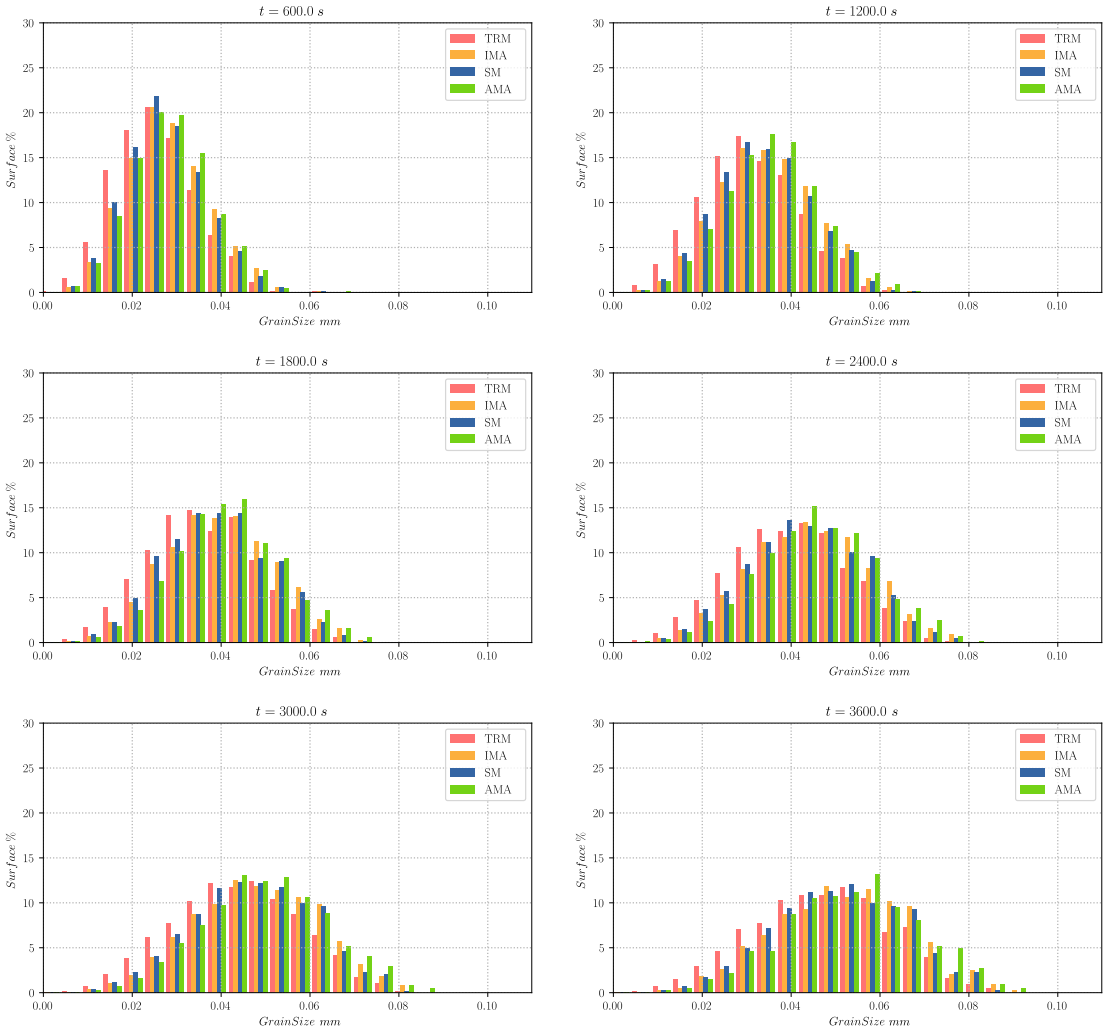


Figure 3.31: Evolution of the grain size distribution pondered by surface for the different models. The TRM model appears to be slower than the others.

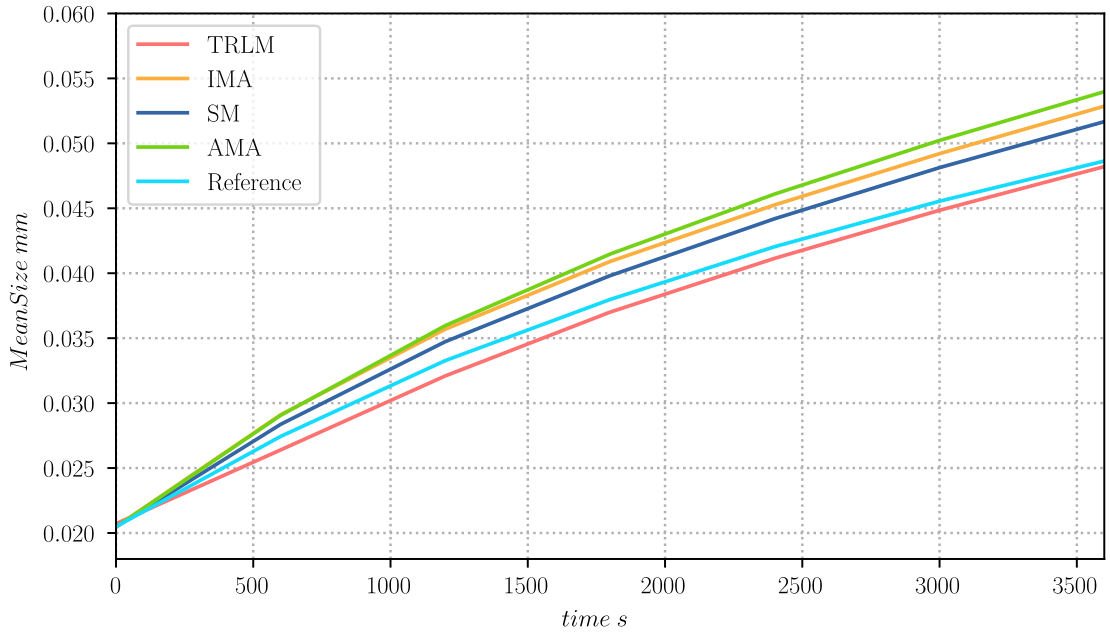


Figure 3.32: Grain size evolution for the 10000 grains test case. The TRM model appears to be slower than the others.

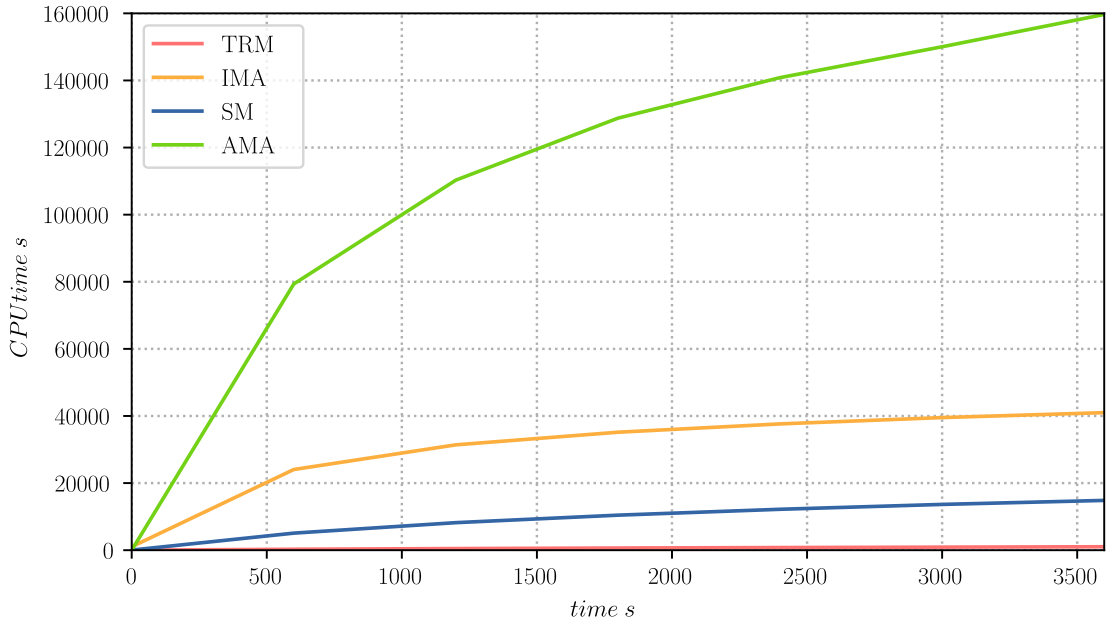


Figure 3.33: CPU-time comparison for the 10000 grains test case. the TRM model was 14,8 times faster that the SM model and 156 times faster than the AMA case.

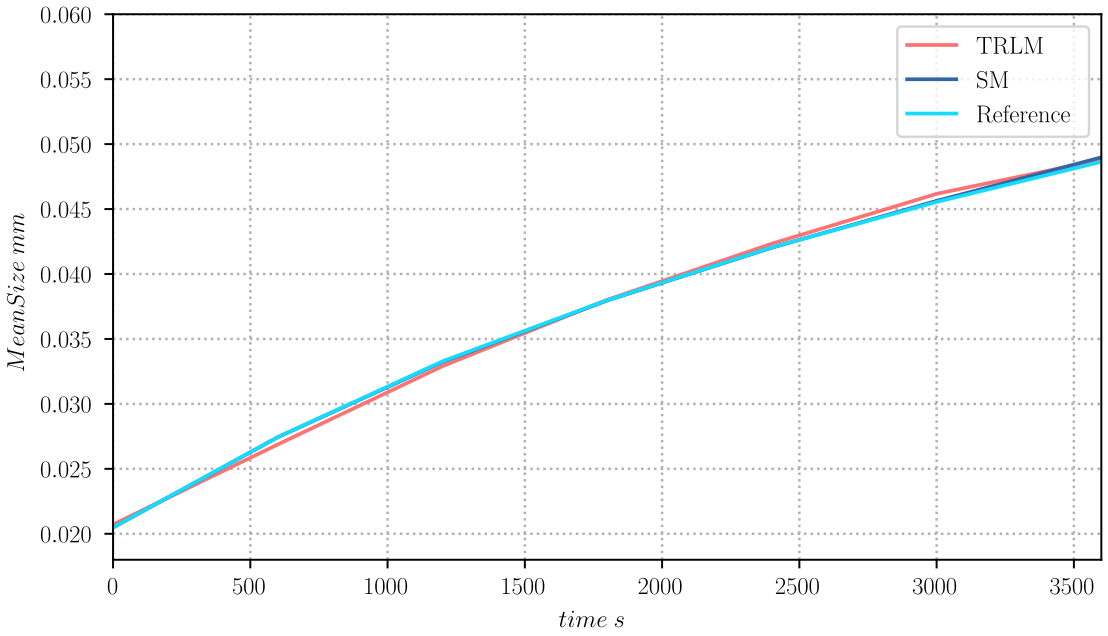


Figure 3.34: Grain size evolution for the 10000 grains test case after a correction on the Mobility M of -12.8% and +6.69% for the SM model and the TRM model respectively. All curves are superposed. This result illustrates a well known behavior of full field simulations of GG: the reduced mobility is classically impacted by the choice of the numerical method and is not only an universal physical parameter.

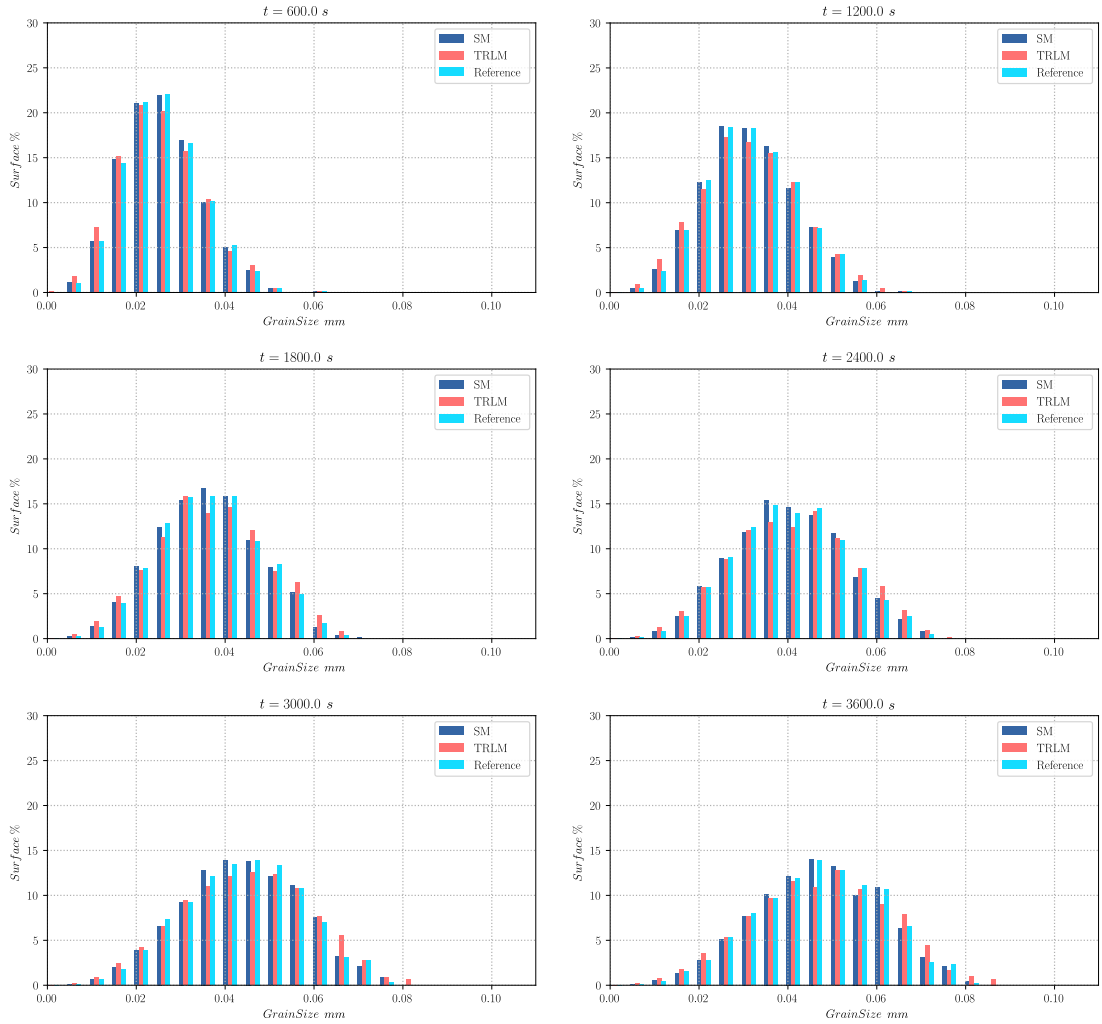


Figure 3.35: Evolution of the grain size distribution pondered by surface after a correction on the Mobility M of -12.8% and +6.69% for the SM model and the TRM model respectively.

3.5 Discussion and conclusion

The development of the TRM method has implemented multiple concepts: the initial LS-TRM preprocessor and geometrical reconstruction have granted a way to characterize microstructures defined using the LS approach to a body-fitted geometric-based data structure. This data structure was intentionally developed in order to approximate the interfaces with piece-wise third degree polynomials (natural splines) and to remesh using selective remeshing operators based on the local geometry present on the mesh. These developments coupled with an explicit Lagrangian movement of the interface have allowed to successfully simulate complex microstructural evolutions such as two dimensional grain growth. Multiple test cases were reproduced in order to compare the behavior of the model when subjected to different situations.

Firstly the sphere shrinkage test has aided to identify a stability range where the TRM model performs very accurately, obtaining errors not higher than 1% to the analytical solution. Then, the T-junction case and the square shrinkage case were used to evaluate the evolution of multiple junctions and the disappearance of a grain. Finally, a test using 10000 initial grains was performed and compared to other more classical methods to perform full field simulations such as grain growth.

The results are very promising as the accurateness of the model on the sphere shrinkage, T-junction and square shrinkage tests is very high, obtaining lower errors than any other model tested. On the final test, the evolution of the mean grain size and the grain size distributions showed that the TRM model is slightly "slower" than the other models; comparisons on the evolution of the grains showed that the morphology of the solution is very similar to the one obtained with the SM model (which had been the best case scenario in chapter 2).

A significant improvement in the computational cost of the full field modeling of grain growth was observed with the TRM model, being 14,8 times faster than the SM model and 156 times faster than the AMA case. Furthermore, Figures 3.31 and 3.32 illustrated the difference between the evolution of the grain size distribution and the mean grain size respectively for the different models, the TRM model obtained the nearest response to the reference followed by the SM case, with an L2-Error of 4.9% and 10.3% respectively on the prediction of the mean grain size and 10.2% and 19.8% L2-Error respectively on the prediction of the grain size distribution.

A final simulation was performed for the 10000 grains test case after adapting the values of the mobility M (which is classically always impacted by the chosen model) for the TRM and the SM models to obtain the same response as in the reference case for the evolution of the mean grain size. The value of the initial

mobility ($M = 8.27549 \cdot 10^{-07}\ mm^4/Js$) was corrected in -12.8% for the SM model ($M_{sm} = M * (1 - 0.128) = 7,21623 \cdot 10^{-07}\ mm^4/Js$) and of +6.69% for the TRM model ($M_{trm} = M * (1 + 0.0669) = 8,829120 \cdot 10^{-07}\ mm^4/Js$). Figures 3.35 and 3.34 illustrate the evolution of the mean grain size and the grain size distribution respectively after the correction, an L2-Error of 2.06% and 14.32% over the mean grain size and grain size distribution respectively was obtained for the TRM model and of 0.734% and 5.06% respectively for the SM model. Here the SM model performed better in terms of accuracy while the TRM model was 13.63 times faster in terms of CPU-time.

Even though the development of this model was only used to simulate 2D isotropic GG, the TRM approach can be applied to simulate mechanisms such as 2D ReX or 2D microstructural evolutions taking into account anisotropic grain boundary properties. Further efforts will be made in order integrate these mechanisms as well as the development of a parallel implementation of the TRM model, these developments will be the subject of study of the following chapters.

Résumé en Français du Chapitre 3

Ce chapitre introduit une nouvelle méthode pour la simulation des problèmes massivement multi-domaines avec applications aux évolutions de microstructures. Cette nouvelle méthode s'apparente en partie aux méthodes de type front-tracking comme les approches vertex mais introduit différentes innovations.

La nouvelle méthode introduite dans ce chapitre, ToRealMotion (TRM), utilise les notions de i. discrétisation explicite des interfaces (de type body-fitted) et ii. du mouvement Lagrangien des noeuds des interfaces pour simuler la migration des interfaces. Toutefois, la nouvelle méthode TRM maintien une discrétisation du volume des domaines à l'aide d'un maillage non-structuré où certaines des arêtes correspondent directement aux segments définissant les interfaces. Cette approche permet d'obtenir un meilleur contrôle sur les évènements topologiques se produisant pendant la simulation, étant donné que ces évènements peuvent être traduits par des opérations de remaillage. De la même façon, conserver un maillage de type non-structuré permet le calcul des problèmes aux Éléments Finis (EF) directement sur la discrétisation du modèle TRM (par exemple, calculs de plasticité cristalline).

La nouvelle méthode a été testée sur différentes configurations, identiques à celles étudiés sur le chapitre 2 pour la caractérisation du modèle Level-Set (LS) couplé à une résolution EF. Les résultats des cas académiques montrent que la méthode TRM possède une très grande précision, meilleure que celle proposé par la méthode LS-EF sur des configurations similaires (à pas de temps et taille de maille équivalente). De même, la méthode TRM a été testée sur une simulation de croissance de grains incluant 10000 grains initiaux, montrant un bon accord avec les résultats obtenus avec la méthode LS-EF, mais avec un temps de calcul largement minoré: 14,3 fois moins important que pour le meilleur cas de figure obtenu pour la méthode LS-EF.

Ainsi, la méthode TRM s'avère prometteuse et est développée plus avant dans les chapitres suivants.

Chapter 4

Parallelization of the TRM model

A new method for the simulation of evolving multi-domains problems has been introduced in chapter 3: the TRM model, for which the accuracy and performance of the sequential method have been proven very promising. In this chapter, further developments of the model will be presented. The main focus here is to develop a robust parallel implementation of the TRM model using a distributed-memory approach with the Message Passing Interface (MPI) library OpenMPI. The intention behind the development of the TRM model in a parallel context, is not only to accelerate the current CPU-times obtained by the sequential approach, but also to be able to simulate microstructural evolutions in the context of large RVE (10^5 to 10^7 grains). In this chapter, the new parallel implementation will be discussed and tested in the context of motion by curvature flow for polycrystals, i.e. by considering GG mechanism. Results of the performance of the model are given and comparisons with other approaches in the literature are discussed.

This chapter has been submitted for publication for publication in [21].

4.1 Introduction

As we have seen, the simulation of the dynamics of massive multidomain problems, have been addressed by several numerical approaches [52, 61, 77, 14, 9, 134, 137, 109, 110], especially in the context of modeling the microstructure of metallic materials. Usually, these approaches aim to enhance the accuracy and the performance of the developed framework by means of computational optimizations, especially when the model intends to simulate thousands of domains. The LS method coupled with a FFT framework (FFT-LS) [153, 150, 152] is an example of a highly efficient method in this context, accounting for a better computational performance than FE-LS approaches, and being able to simulate a very high number of grains ($\approx 6 \cdot 10^5$ initial grains where used in [152] in a 2D GG context). The FFT-LS approach is, however, limited by the use of regular grids, an aspect that could restrict its domain of use to a static context or for small deformations where remeshing is not necessary. Hence, this approach can not be used for the modeling of DRX that the FE-LS method has already proven to be able to simulate [7]. As an alternative to all aforementioned models (MC, CA, LS and MPF and Vertex methods), in chapter 3, inspired by the Lagrangian component of Vertex [94, 14, 15, 16, 17, 18] and Front-Tracking approaches [102, 9, 10, 11, 12, 13], we have proposed the TRM model. This model can approach the computational performance of FFT-LS simulations, while not presenting limits given by its numeric framework.

The initial TRM model algorithm proposed in chapter 3 described a series of selective remeshing operations strongly influenced by the works in [199, 200] and a data structure adapted specifically to multidomain problems. All remeshing operations were performed over a local patch of elements while the data structure was based on the local topology (of the simulated domain) that each node represents (multiple junction or *Point*, grain boundary or *Line*, grain bulk or *Surface*). Geometric properties of the interface were computed with the help of piece-wise polynomials (Natural Splines) and the movement of interfaces was based on a Lagrangian model. Moreover, some topological changes on the structure of the multidomain problem were addressed by means of local remeshing operations, where the Node-Collapse algorithm was used to treat the disappearance of domains while the Point-Splitting algorithm allowed the creation of new interfaces.

The performance of the TRM model was tested and compared against a classical front capturing LS-FE framework [5, 167, 7] in an isotropic GG context. Multiple test cases were performed concluding in an improvement of the accuracy and performance when using the TRM model. A 14 times reduction in the computational cost was observed as compared to the best-case scenario of the LS-FE framework.

Even though the results obtained in chapter 3 are very good, the current

state of the TRM model lacks a very important component in order to make a more accurate comparison in terms of computational performance: a parallel implementation. Two categories of parallel frameworks can be designed to use all the capabilities offered by modern computational units. These two categories are differentiated by the management of the active memory of the running processes: *shared memory*, where all processors share and interact with the same location in memory, and *distributed memory*, where each process has its own independent memory location. The choice of whether to use one or the other relies strongly on the hardware architecture on which the model is intended to run. Normally an application with a shared memory framework can not be used over a supercomputer cluster with thousands of cores, as the memory in these systems is not connected to a single board but it is distributed between several independent CPUs connected to the same network. In other words, shared memory can only be used in single machines while distributed memory is intended to be used both within single machines and over a network of interconnected CPUs. Each processor in a distributed memory system needs a way to communicate information to other processors. This communication can be established by the use of standard protocols such as MapReduce [202] or the Message Passing Interface (MPI) [203]. These methods of sending data between processors cause some overhead on the global multi-processor application, hence obtaining a lower performance than when using a shared memory approach. On the other hand, shared memory protocols can also add some overhead when implementing an environment safe of race conditions: when two or more processors try to write to the same memory location at the same time. Here, a distributed memory approach using the MPI protocol in order to address the parallel implementation of the TRM model is proposed to enable a broader range of hardware compatibility.

Very few publications exist in the literature regarding the parallel implementation of Front-Tracking models, examples of these works can be found in [204, 205] in the context of two-phase flows, and in [206] in a more general context of multiphase flows, however, tested for only just a few domains (3 domains). Similarly, examples of vertex models using a parallel scheme can be found in [207, 208] using a GPU-based parallel approach (hence using a *shared memory* approach) in the context of molecular-dynamics simulations. These examples, although very impressive, are considerably different from the TRM approach, as an important strength of the proposed methodology is to deal with unstructured finite element meshes, allowing (i) the discretization of domain boundaries by more than a vertex-vertex connection [207, 208], (ii) the direct evolution/migration of the nodes of the mesh as a mean of boundary kinetics, without the use of two discretization approaches [204, 205] and (iii) large deformation modeling (thus adapted to the context of recrystallization modeling in hot metal forming) which is generally not accessible to regular grid approaches (MC, CA, and FFT models).

In this chapter, the parallel implementation of the TRLM model will be pre-

sented and tested in multiple hardware settings. The special algorithms to address the parallel framework will be explained. Moreover, an additional tool is needed when performing parallel computations over a distributed memory approach, in order to solve mesh-based problems: the initial partitioning of the numerical domain and the redistribution (when necessary) of charges (repartitioning) through the evolution of the simulation. Here we opted to use the open-source library Metis [209] to obtain the initial partitioning while the redistribution algorithm has been developed and will be also presented in this paper.

Performance and speed-up of the model will be given and compared to other highly efficient parallel methods in the literature in the context of GG using a FFT-LS approach [153, 152].

4.2 The TRM model: sequential approach in a GG context

In chapter 3, a numerical method for the TRM model was presented, this numerical method is built on a data structure defining the current state of the multidomain framework, defining geometrical entities such as *points*, *lines* and *surfaces*. Each *point* is composed of a P-Node (defining a node of the mesh with a topological degree equal to 0) and a set of connections to other *points* and *lines*. Each *line* is defined by an ordered set of L-Nodes (nodes with a topological degree equal to 1), an initial *point* and a final *point*. Finally, *surfaces* are defined by a set of S-Nodes (nodes with a topology degree equal to 2), a set of elements and a set of delimiting *lines* and *points*.

Once the data structure was defined, a preprocessor of the TRM model was introduced. This preprocessor makes an interface between the LS domain definition and the TRM data structure based on the works presented in [19, 161], where an initially implicit mesh (with an immersed LS data set as in [146, 6]) is transformed into a body-fitted mesh via a *joining and fitting algorithm*, where all limits of domains are explicitly defined by some of the nodes of the mesh (see Fig. 3.3 for an illustration). Subsequently, four algorithms for the reconstruction of the domains for the TRM model were presented: Nodal geometric tagging, Point Reconstruction, Line Reconstruction, and Surface Reconstruction. These algorithms completely define the data structure of the TRM model of a LS data set immersed into a body-fitted mesh. It is important to emphasize that from this point forward, the LS data set is no longer of use for our model, and that all geometric properties of the interfaces and domains are computed using purely geometric approximations built upon the data structure of the TRM model. Natural parametric splines [196] are used to approximate the domain interfaces (lines) with third degree piece-wise polynomials, local geometric properties of the interfaces such as the curvature κ and the normal $\vec{n}$, are deducted from these approximations.

Once a velocity has been computed on the mesh, depending on the physical model being simulated, each node N_i of the mesh is moved to a new position $\vec{r}_i$ in a Lagrangian way using Eq. 3.6. Mesh conformity (in a FE sense) is ensured by a locally-iteratively movement-halving algorithm, reducing at each iteration the movement by half when an invalid mesh configuration is encountered, i.e. an element flipping (see Fig. 3.16).

TRM model also involves a particular remeshing procedure. The TRM data structure needs to be maintained at all times during remeshing to ensure all geometric computations over the defined geometric entities. When a remeshing procedure is performed, the mesh evolves and the sets defining each geometric entity have to adapt. Therefore, the remeshing procedure must take into account

the local data structure of the nodes and elements involved in each remeshing operation. The remeshing strategy of the TRM model uses the separate definition of local *selective*[1] remeshing operators: selective vertex smoothing, selective node collapsing, selective edge splitting, selective edge swapping and selective vertex gliding, see chapter 3 for a complete definition of each operator. As a general rule, the remeshing procedure is performed to increase the general quality Q of the mesh (or a patch of elements of the mesh). Here, the mesh quality Q is as a factor of the shape and the size of the elements using the same approach as in [184]. However, the selective remeshing procedure is not only driven by the local mesh quality Q, but also by the local topological degree of the nodes involved in the operation. In chapter 3 a global remeshing procedure was introduced, driven by two nodal fields δ_c and δ_s corresponding to the collapsing and splitting fields, and a minimum quality shape coefficient q_s. The complete remeshing procedure is summarized in Algorithm 3.

Finally, the TRM model was applied to the modeling of isotropic GG (see section 3.3) where special attention was given to all possible topological changes occurring on the grain boundary network during this phenomenon.

[1]The word selective denotes a variation of the original remeshing operations when performed over the data structure of the TRM model, as each remeshing operation is performed differently over nodes with different topology (P-Node, L-Node and S-Node)

4.3 Parallel strategy for the TRM model

The TRM model introduced in chapter 3 lacks an important component to be a competitive tool to the more classical methods presented in the literature for the modeling of microstructural evolutions, this component is the ability to perform large computations thanks to a parallel implementation.

As mentioned in the introduction, in order to address the parallel implementation of the TRM model a *distributed memory* approach is proposed, using the standard communication protocol MPI [203] to communicate information between processes.

Regardless of the choice of the memory management of the parallel framework (*shared memory* or *distributed memory* system), two additional tools are needed when performing parallel computations to solve mesh-based problems: the initial partitioning of the numerical domain and the redistribution of charges (repartitioning) through the evolution of the simulation. These two tools are essential to ensure an equilibrium of the memory and the charge that each processor has to handle. In a *distributed memory* system, the whole numerical domain can be divided to give to each processor one part of that domain to handle at the beginning of the simulation. Multiple tools already exist in order to partition a mesh made of simplices, normally these tools can work with topologies much more complex than an Eulerian mesh, as they are engineered to treat graphs[1]. Here we opted to use the free library Metis [209] to obtain the initial partitioning.

Once the partition of the domain is performed and a strategy for the re-equilibrium of charges is developed, one problem arises: making changes on the mesh at the boundaries of each partition is very difficult. Every partition involved in the remeshing of the local patch of elements needs to perform exactly the same operations but without all the information required (because a part of the local patch of elements is not present in its memory). This has been addressed by using the properties of the repartitioning process as explained further.

In this section the parallel implementation of the TRM model will be explained, all situations described before will be addressed below.

4.3.1 Initial partitioning

Two ways of partitioning an Eulerian mesh are classically used: the first is to partition the graph of the mesh (each node is a vertex in the graph and each edge of the elements is an edge on the graph) to obtain multiple subsets of nodes. Each partition will receive a subset where their nodes are not present in any other

[1] mathematical structures used to model pairwise relations (edges) between objects (vertices, nodes or points)

subset. The second is to compute the partition over the dual graph of the mesh (or dual graph of a planar graph[2]) of the initial mesh. In the dual graph, each vertex represents an element of the initial planar graph and each edge represents an edge of the initial planar graph that is shared by two elements. A planar graph may have multiple dual graphs, depending on the embedding of the planar graph in the plane (at this stage it is no longer a planar graph but a plane graph). However, once this embedding is defined, the dual graph of the plane graph is completely defined. Coherent Eulerian meshes are planar graphs: in a 2D context, the computation of the dual mesh uses each 2-simplex (elements) as a vertex and each 1-simplex (edges) as an edge, while in a 3D context each 3-simplex (elements) is treated as a vertex and each 2-simplex (facets) as an edge. Partitioning the dual graph of the mesh produces subsets of elements of the initial mesh, each partition will receive a subset where their elements are not present in any other subset. Figure 4.1 shows examples of the dual graph of a mesh in 2D and 3D.

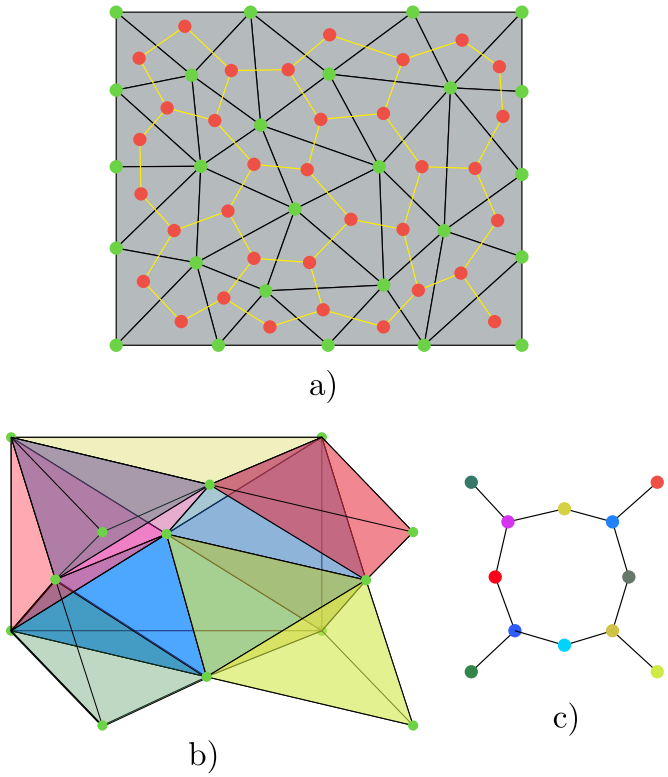


Figure 4.1: Example of dual graph in 2D and 3D. a) mesh and dual graph of the mesh, green points are nodes from the mesh and red points are nodes from the dual graph of the mesh, b) mesh in 3D c) dual graph of the mesh in b).

Metis can make partitions using either the graph of the mesh or the dual graph of the mesh. In our context, we have chosen to use the dual mesh. When a mesh

[2]a graph that can be projected into a plane and that its edges intersect only at their nodes

is directly partitioned, some of the edges of the mesh crossed by the boundaries of partitions must be managed. While it is still possible to use this configuration in a parallel context by repeating the elements that contain at least one edge crossed by the partitions as in [189, 210], in our context where a certain geometrical data structure must be respected, using such a way of partitioning is more complex. By partitioning the dual mesh, even though each part receives elements not belonging to any other part, some nodes can be shared by several parts at the same time, and from this point forward these are defined as Shared-Nodes. Figure 4.2 shows examples of partitions made using the graph of the mesh and the dual graph of the mesh in two parts and three parts.

Once the mesh is partitioned, each processor receives a part of the elements that are more or less equally distributed among all the other processors. However, this is true only during the preprocessor step of the simulation. Once the TRM model starts remeshing and changing the position of nodes, the charge of each processor evolves (see section 4.3.3 for this aspect).

4.3.2 Numbering geometric entities and regularization

In the present parallel context, some geometric entities may appear in more than one partition at the same time. This creates issues as the algorithms to reconstruct the geometric entities had been developed in a sequential context (see section 3.2.2). By performing the sequential algorithms for the reconstruction of entities, each partition will consider that the geometric entities stop at the boundaries between partitions. Problems as the one presented in Fig. 4.3 could arise: in this configuration with two surfaces and a line, the expected identification of each geometrical entity should be the one illustrated in Fig. 4.3.a, instead, as each partition does not contain all the information, the identification of entities would be the one illustrated in Fig. 4.3.c where multiple geometric entities of the same type (Point, Line or Surface) can have the same identity (i.e. $Surf_1$ from $Part_1$ and $Surf_1$ from $Part_2$) but do not correspond to the same entity, or inversely, entities having different identities within the same partition but being the same entity in the whole domain (e.g. $Line_1$ and $Line_1$ from $Part_2$ or $Surf_1$ and $Surf_3$ from $Part_2$).

We have solved these situations in a two-step process: first a non-repeating numbering system in parallel and then a regularization of the identity of entities crossed by the partitions.

The non-repeating numbering system can be implemented very easily: Each geometric type (Point, Line or Surface) will be numbered according to the partition where it is present, starting with the number of the local partition (i.e. if numbering in $Part_3$ the first Line will be numbered as $Line_3$ the first point as

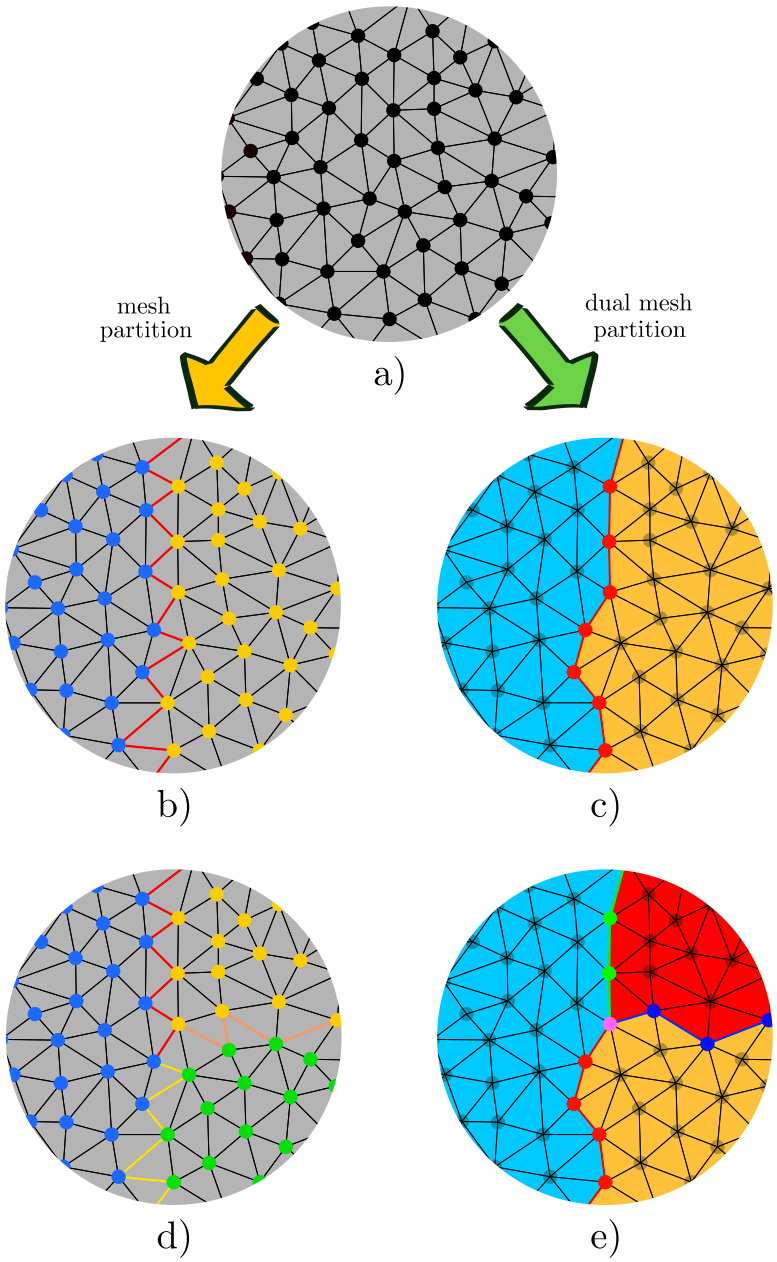


Figure 4.2: Example of partitioning using the mesh and the dual mesh. a) mesh to partition, b) and d) partitioning using the mesh in two parts and three parts respectively, each part is represented by a color on the nodes, each part receives nodes not belonging to any other partition, some vertex (in color) are crossed by the partitioning. c) and e) partitioning using the dual mesh in two parts and three parts respectively, each part is represented by a color on the elements, each part receives elements not belonging to any other partition, some nodes (colored nodes) are in several partitions at the same time, these are Shared-Nodes.

$Point_3$ and the first surface as $Surf_3$) subsequently the numbering increases by

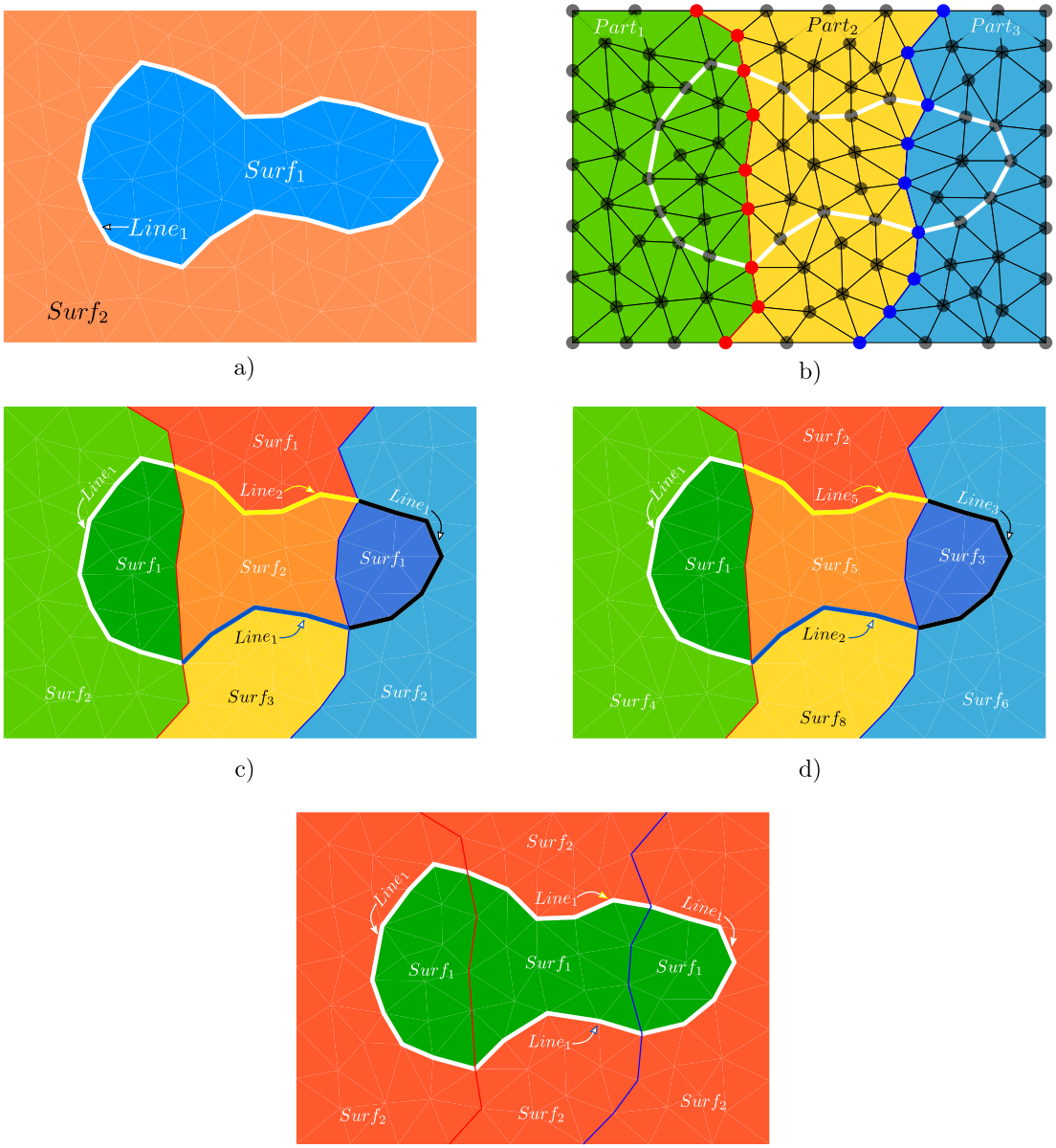


Figure 4.3: Example of the initial numbering of geometric entities of a domain crossed by the boundaries partitions. a) geometric entities to number and also sequential numbering b) the same domain partitioned in three parts, each color represents a partition, c) configuration of the geometric entities on each part of the domain, the numbering is done by default d) configuration of the geometric entities on each part of the domain, the numbering for each geometric type (Point, Line or Surface) begins with the number of the local partition and increases by the number of total partitions, e) configuration of the geometrical entities after renumbering.

the number of total partitions (i.e. if numbering, in $Part_3$ and the total number of partitions is 4, the second Line will be numbered as $Line_7$). This numbering system is illustrated in Fig. 4.3.d. This solves the problem that multiple entities can be identified as the same in different partitions but not correspond to the same (see Fig. 4.3.c).

The regularization helps to identify the entities that have been crossed by the boundaries of the partitions. This procedure is implemented via a local identification of the entities coupled to the shared-nodes, a communication process of that information, a remote identification of all entities and a re-numbering of the entities that need it. Algorithm 5 contextualizes this procedure with the help of a recursive function presented in Algorithm 6. Note that the lines 4: and 10: of Algorithm 5 send information to other partitions. These operations are realized in the actual implementation using some predefined functions of MPI (MPI_Alltoallv and MPI_AllGatherv) performed over all processors. The purpose of the call of Algorithm 6 in line 14: is to fill the set *SameEntity*. This set gathers list groups the identities given to a single geometric entity over all partitions, this list is computed with the same information in all processors hence it gives the same answer in all of them. In the configuration given in Fig. 4.3.d, when Algorithm 5 is performed over the Surfaces (the geometric entities defining Surfaces in the TRM model), *SameEntity* would store two lists one for each iteration of the while loop of line 11: in the first iteration the list would be composed of three pairs: $\{\{Surf_1, Part_1\}, \{Surf_5, Part_2\}, \{Surf_3, Part_3\}\}$ and the second iteration of four pairs: $\{\{Surf_4, Part_1\}, \{Surf_8, Part_2\}, \{Surf_2, Part_2\}, \{Surf_6, Part_3\}\}$. On each iteration, the entity will be named on all partitions with the lowest identity found for it, hence in our example, $Surf_5$ and $Surf_3$ will be named $Surf_1$ in their respective partitions and $Surf_4$, $Surf_8$, and $Surf_6$ will be named $Surf_2$ in their respective partitions. This finally solves the inconsistencies found by the initial numbering of entities (see Fig. 4.3.e).

At this stage one final situation that needs solving appears: if there are non *locally connected*[3] entities in the same partition with the same identity (e.g. $Surf_2$ in $Part_2$ and $Line_1$ in $Part_2$ of Fig. 4.3.e) these entities have to be gathered in one single entity. This procedure is easily made for surfaces as it is only necessary to exchange the nodes and elements from one Surface to the other and to remove the one that remains empty. On the other hand, in order to regroup Lines, a piece of additional information is needed as one Line can now be made of disconnected segments and this was not taken into account in the algorithms developed in chapter 3. This has been solved by adding a piece of additional information to L-Nodes: the *previous* and the *next* nodes on its Line. If there is no previous node or next node at that position in the Line, the value attributed to it is $Null$. i.e. if in a Line L_1 the L-Node b is the first Node, the L-Node c is the second node and

[3]Geometric entities that are connected on partitions different from where part of them are stored

the Line has an initial Point with a P-Node a, the previous and next nodes of b are a and c respectively and the previous node of c is b. Note that node a does not have information about its previous and next node as it is a P-Node and not a L-Node. This implementation solves the fact that one line can be segmented in multiple parts without the need for further changes to the already existent code.

Algorithm 5 Identity Regularization Algorithm performed on $Part_i$

for all Shared-Nodes : N_i **do**
  for all SharedRanks of N_i : $Part_j$ **do**
    $I_L \leftarrow$ local identity of the coupled entity of N_i
    send to $Part_j$: Pair(N_i, I_L)[4]
for all Parts: $Part_j \neq Part_i$ **do**
  for all Received Pairs from $Part_j$: $Pair_k$ **do**
    $N_i \leftarrow$ the Node $First(Pair_k)$
    $I_L \leftarrow$ local identity of the coupled entity of N_i
    $I_R \leftarrow$ Second($Pair_k$)
    send to all Parts: Triplet(I_L, I_R, $Part_j$) [5]
while Triplets to Treat **do**
  $TT \leftarrow$ Take First Non-Treated Triplet
  Create empty list of Pairs: $SameEntity$ ▷ [Identity, Partition]
  Call **RecursiveTripletTreatment**(TT, $SameEntity$)
  $I_{Lowest} \leftarrow$ the lowest value of the first item in all pairs of $SameEntity$
  for all Pairs in $SameEntity$: $Pair_k$ **do**
    if $Second(Pair_k) == Part_i$ **then**
      $I_{Old} \leftarrow First(Pair_k)$
      change identity of the entity with number I_{Old} to I_{Lowest}

Algorithm 6 Recursive function for the identification of the same entity given a triplet Tx and a list of pairs to fill P_L

function RecursiveTripletTreatment(Tx, P_L)
  $I_L \leftarrow First(Tx)$ (local identity)
  $I_R \leftarrow Second(Tx)$ (remote identity)
  $Part_r \leftarrow Third(Tx)$ (remote Partition)
  for all Received Triplets from Part $RPart$: $Triplet_i$ **do**
    if $Triplet_i$ is still not treated and $First(Triplet_i) == I_R$ **then**
      Set $Triplet_i$ as treated
      Add to P_L : Pair($Second(Triplet_i)$, $Third(Triplet_i)$)
      Call **RecursiveTripletTreatment**($Triplet_i$, P_L)

4.3.3 Re-Equilibrium of charges, repartitionning: Mesh Scattering

Repartitioning in a distributed memory framework is much more complex than making an initial partitioning as all the information is scattered through all processes. Even though there exist tools that solve these kinds of problems (e.g. an extension of Metis called ParMetis [209]) we have opted to develop our own repartitionner. This is mainly because in the present context we have to make sure that the data structure of the different geometric entities stays consistent during the repartitioning process. Of course, our objective is not to develop a partitioner having all the capabilities of the well-established libraries of the domain, but to develop a simple and robust way of exchanging information between process having parts of a scattered unstructured mesh on its memory.

Dynamic Ranking System.

The repartitioning algorithm developed for the TRM model will use a ranking system to determine the direction of the information flow, each process having its own unique rank[6]. This rank can be determined for example by using the number of elements, the number of nodes, the number of edges, or the total surface of the domain stored on each process, and in the case were two processes have the same value, a random attribution is held. Here, we will use the number of elements to obtain the rank R_i of process i. Once each process has determined the number of elements present on its memory, this information is sent to all other processors via MPI, each processor makes the comparison procedure and stores a list $RankingOrder$ of the ranks of each processor, the lower the number of elements on a partition, the higher is its rank. This list is computed equally on all processors. Moreover, if each part i needs the rank of part j, this information is obtained via $RankingOrder[j]$.

Shared-Nodes.

As defined before, Share-Nodes are the nodes belonging to multiple processes at the same time, these nodes reside at the boundaries between partitions. Additionally to the Dynamic Ranking System, some other information will be required for the repartitioning operation, this information denotes for all Shared-Nodes, to which partitions they are shared. by considering for example Fig. 4.2.c, here the red nodes are shared by the cyan and by the yellow partitions, the red nodes from the cyan partition must know that they are shared by the yellow partition and vice versa. Some nodes can be shared by more than two partitions as in Fig. 4.2.e. In this particular case, three partitions share the magenta node. when this node is processed the blue part must know that it is shared by the yellow part and the red part. This information can be obtained very easily: each part

[6]in our scope, these ranks refer to different ranks than those attributed to each process by MPI, at the beginning of the parallel environment.

communicates via MPI to all processes which nodes are at its boundary, then for all received nodes, if the node exists in the current part, it means that it is shared with the sender part. Every Shared-Node then stores a list *SharedRanks* with the parts containing it.

Unidirectional Element Sending.

As explained in section 4.3.1, our parallel scheme maintains sets of non-repeated elements scattered in all processes, this means also that the repartitioning scheme must exchange elements between partitions maintaining also this property. If a partition sends one element, this element must have only one destination to maintain coherence in the proposed algorithm. This goal is achieved using the ranking system and the *SharedRanks* list of the Shares-Nodes in Algorithm 7.

Algorithm 7 Unidirectional element selection in Part $Part_i$

1: Store a list of list: SharedNodesPerPart
2: **for all** Shared-Nodes : N_i **do**
3: **for all** SharedRanks of N_i : $Part_j$ **do** add item N_i to SharedNodesPerPart[$Part_j$]
4: Store a list of list: ElementsToSendToPart
5: **for all** Parts: $Part_j$ **do**
6: **if** RankingOrder[$Part_i$]<RankingOrder[$Part_j$] **then**
7: **for all** SharedNodesPerPart[$Part_j$] : N_j **do** add $Elements(N_j)$[7] to ElementsToSendToPart[$Part_j$]
8: take out repeated elements in ElementsToSendToPart[$Part_j$]
9: **for all** Parts: $Part_j$ **do**
10: **for all** ElementsToSendToPart[$Part_j$] : E_j **do**
11: **for all** Nodes(E_j)[8] : N_k **do**
12: **for all** SharedRanks of N_k : $Part_k$ **do**
13: **if** RankingOrder[$Part_j$]<RankingOrder[$Part_k$] **then**
14: Erase E_j from ElementsToSendToPart[$Part_j$]

In the first part of this Algorithm 7, two lists are created, these lists contain the Share-Nodes and the Elements to send to each partition. Note that each processor performs this algorithm, hence, each processor has a list of elements to send to all the other processors (even if that list is empty). A first selection is made via the inequality of line 6: of Algorithm 7, where the list of elements to export only accepts the elements that have at least one node shared with a partition with a superior rank (a partition with a lower number of elements). In the last section of the algorithm, a final selection is made for the elements that are in the list to be sent to multiple processors (i.e. for elements with nodes shared by more than two processors at the same time). The final destination selection for

these contentious elements is achieved by choosing the partition with the highest rank (lines 10: to 15: of Algorithm 7), erasing the element in question from all the other lists. Once this algorithm is executed in all partitions, a *scattering* process begins, sending all elements to their new partitions along with some associated information: Node positions, Node fields, element fields and some data necessary to rebuild the structure of the geometric entities involved in the scattering (see section 4.3.4).

Figure 4.4 illustrates one example of the behavior of the unidirectional element selection algorithm for three partitions. The rank of each partition is computed according to the number of elements in decreasing order. In Fig. 4.4.b are shown the elements to be sent to $Part_2$. Here, Elements 3, 4 and 5 appear only to be sent to $Part_2$, and not to $Part_1$. This filter is applied at the initial part of the algorithm as the inequality RankingOrder[$Part_3$]<RankingOrder[$Part_1$] is not evaluated to true. Elements 1 and 2 appear initially on the list to be sent to $Part_2$ and $Part_3$ but are filtered in the last part of the algorithm, as $Part_2$ is of higher rank. In Fig. 4.4.c the selected elements to be sent to $Part_3$ are displayed. Note that some of the elements of $Part_3$ are going to be sent to $Part_2$ hence they appear in a different color. Also, the intersection of elements from $Part_1$ to be sent to $Part_2$ and $Part_3$ is empty, hence no errors will be made in the scattering process. Figure 4.5 illustrates the initial and the final configuration after the scattering. The boundaries of all parts are displaced by the scattering including the shared node between the three parts (Node a) after the scattering, Node b is shared by the three parts while Node a is part of the bulk nodes of $Part_2$.

4.3.4 Geometry reconstruction

In the parallel re-equilibrium of charges explained in section 4.3.3, a particular problem appears. When exchanging elements and nodes between partitions, a reconstruction of the geometrical entities that involve those elements and nodes have to be considered. Having the data structure presented in section 3.2.1, it is clear that these reconstructions must be addressed depending on the type of geometry in question. While the case of Surfaces is trivial (they move along with elements), the consideration of the geometric entities attached to L-Nodes and P-Nodes is more complex.

When reconstructing the entities coupled to the new L-Nodes and P-Nodes (Lines and Points) more information is needed from the sending part: new P-Nodes on the receiving partition need to create new Points, hence, information regarding the connections of that point to other points and lines is needed (see the data structure of points of section 3.2.1). In some cases, this information needs to be collected from multiple partitions at the same time as illustrated in Fig. 4.6. Here the scattering procedure is being performed near a Point attached to the

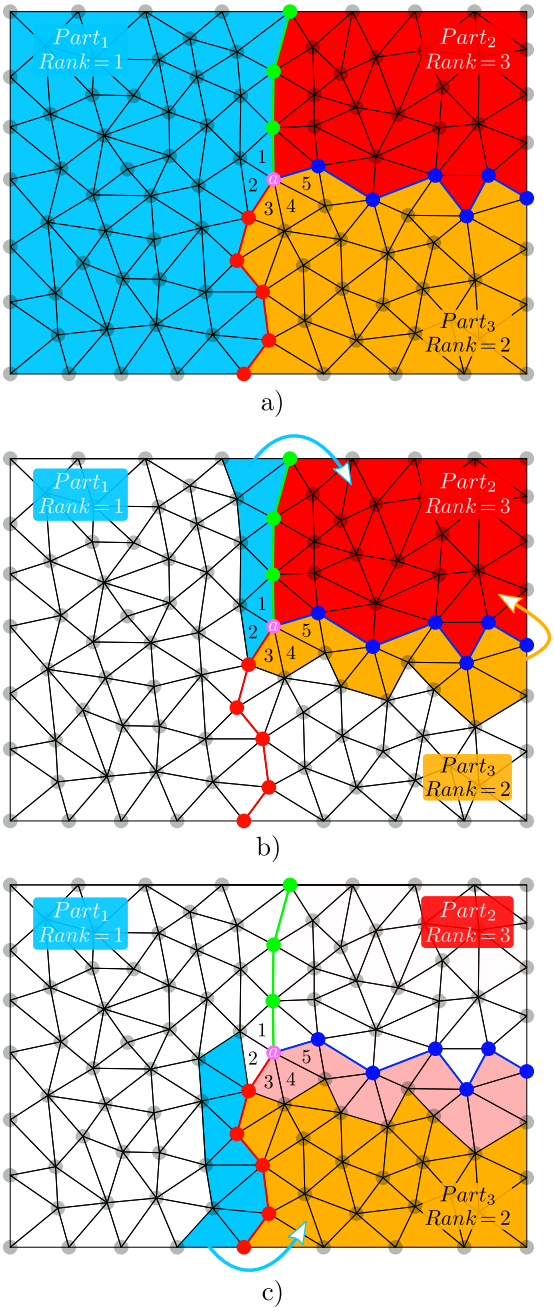


Figure 4.4: Example of the behaviour of the Unidirectional selection element algorithm. a) initial state with three parts, the name and the rank of each partition is displayed. b) Selected elements to be sent to $Part_2$, elements 3, 4 and 5 appear only to be sent to $Part_2$, and not to $Part_1$. Elements 1 and 2 appear initially on the list to be sent to $Part_2$ and $Part_3$ but they are filtered in the last part of the algorithm, as the higher rank of the nodes of these elements belongs to $Part_2$. c) selected elements to be sent to $Part_3$, the intersection of elements from $Part_1$ to be sent to $Part_2$ and $Part_3$ is empty.

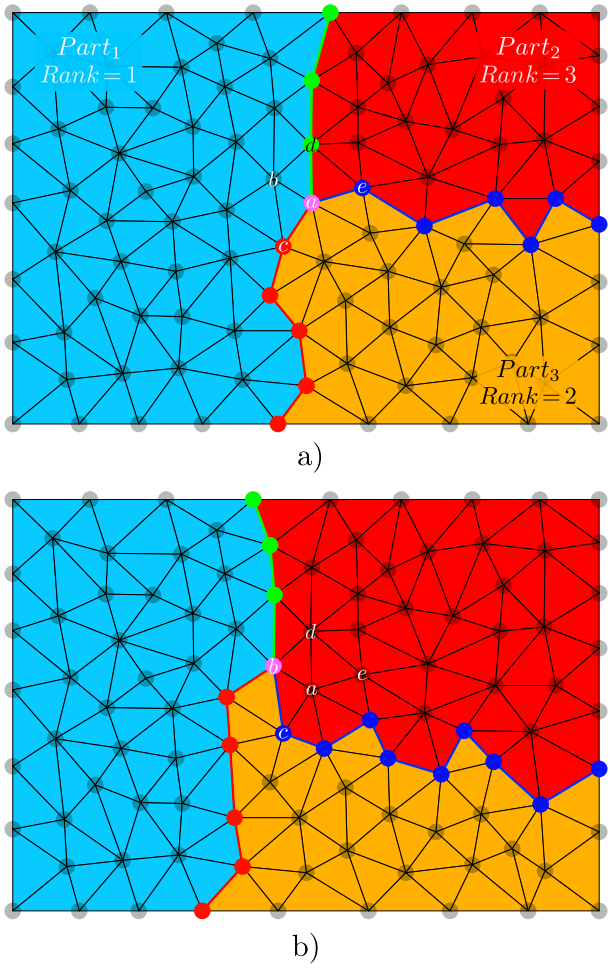


Figure 4.5: Example of the new configuration of each part after the scattering of Fig. 4.4 to their respective new partitions, the boundaries of all parts are displaced by the scattering including the shared node between the three parts a) initial configuration, node a is shared by the three parts, b) configuration after scattering, now node b is shared by the three parts while node a is part of the bulk nodes of part 2.

P-Node a, in the initial configuration (Fig. 4.6.a). This Point is in the memory of $Part_1$ and $Part_3$ (hence P-Node a is a Shared-Node). The connections of L-Nodes a and b are only known by $Part_1$ and $Part_3$ while the connection to P-Node c is known by both partitions, note that $Part_2$ does not have any of the previous information regarding the connections of P-Node a. After the scattering, elements 1 and 2 were sent to $Part_2$ by $Part_1$ and $Part_3$ respectively, thus both partitions need to send the adjacent information to create all the nodes unknown by $Part_2$. $Part_1$ sends the node a and b and $Part_3$ sends the node a and d, both partitions sent information regarding the node a because they have no way to know that the other partition has already sent it. This is not necessarily a costly step because the information regarding the connections of the Point of P-Node a

can be attached to the communication. Indeed, information regarding all connections of the the P-Nodes being sent to other partitions is also broadcasted. This means that $Part_1$ sends the P-Node a along with the identities c and b and $Part_3$ sends also the P-Node a along with the identities c and d corresponding to the connections known by $Part_1$ and $Part_3$ respectively. $Part_2$ receives two times the P-Node a, however it is only created once. On the other hand, the information received about its connections is used: $Part_2$ searches on its memory for the existence of the nodes b, c and d received in the communication to be connected to the Point of P-Node a and if they exist, the connection is made. Note that the P-Node c can not be found by $Part_2$ hence that connection can not be created.

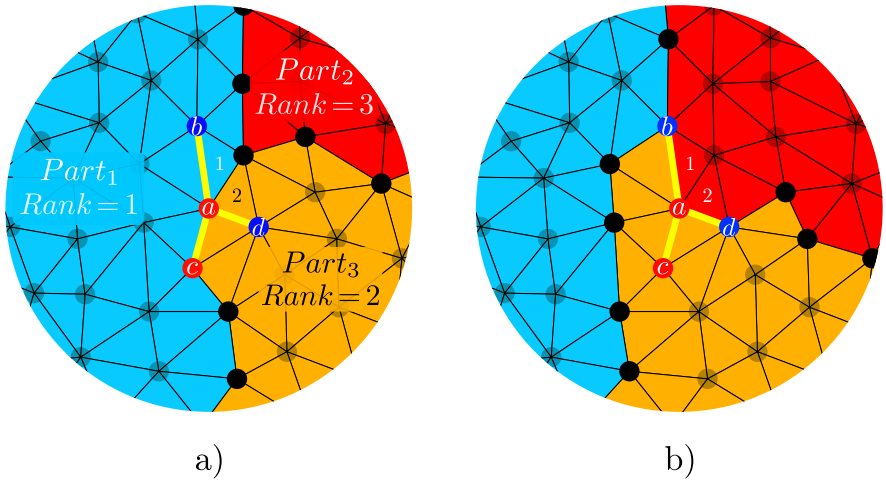


Figure 4.6: Example of the scattering near a P-Node a the corresponding connections of its coupled Point are displayed in yellow to L-Nodes b and d and to P-Node c. a) Initial configuration, connections to the Point of P-Node a are stored in $Part_1$ and $Part_3$ and b) Configuration after the scattering, connections to the Point of P-Node a are distributed within all parts

Line reconstructions are performed similarly as for Points. The data structure of Lines is special as it is composed of an ordered sequence of L-Nodes and optional initial and final Points as stated in section 3.2.1. Furthermore, in section 4.3.2 of this chapter an additional property was included for L-Nodes: information regarding the previous and next nodes within the same Line was added. This additional property helps to reconstruct the Lines involved in the scattering by sending the identity of the previous and next nodes along with the L-Node, additionally to the identity of the Line where it must be added. The position of the L-Node within that Line is then obtained by its relative position to these next and previous nodes if they exist in the memory of the partition (similarly to the reconstruction of Points). If one of those nodes do not exist, it means that the Line is segmented at that L-Node. Figure 4.7 shows the initial and final states of a scattering between two partitions where a Line has been crossed by the boundary of two partitions $Part_1$ and $Part_2$. In the scattering, two L-Nodes

are sent by $Part_1$ to $Part_2$. Node a carries the identity of its previous (L-Node g) and next (L-Node b) nodes. Similarly, node d carries the identity of its previous (L-Node e) and its next (L-Node h) nodes. This information helps to connect the new nodes in $Part_2$ to the Line in their respective positions and to fill the information about the previous node of L-Node b and the next node of L-Node e (and vice versa). Note that even though the information about the previous node of L-Node a was sent, this information is not used as the node g does not appear in the memory of $Part_2$ (see Fig. 4.7.d). Consequently, the information of the previous node of L-Node a is set as $Null$.

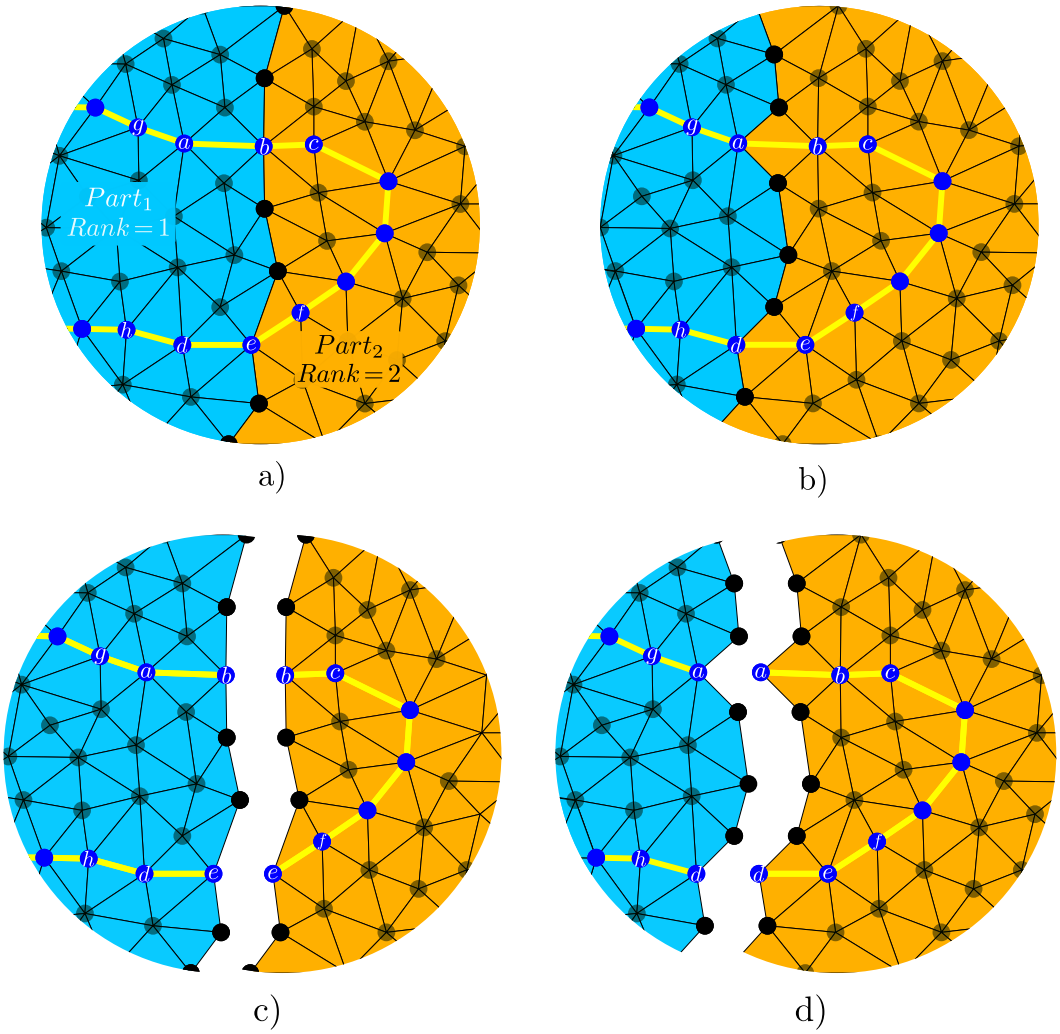


Figure 4.7: Example of the scattering with a Line (yellow) crossed by the boundary of two partitions $Part_1$ and $Part_2$, the assembled and separated views are displayed, the assembled view shows the domain being simulated while the separated view show the actual memory of each partition. a) and c) Initial configuration, assembled and separated views respectively, b) and d) configuration after scattering, assembled and separated views respectively.

4.3.5 Blocking remeshing at partition boundaries

Making changes to the mesh at the boundaries of each partition is very difficult because part of the local patch of elements is not present in its memory. Every partition involved in the remeshing of the local patch of elements needs to perform the same operation but without all the information required. We have solved this in a simple manner by blocking all operations involving a change (change of connectivity or position) on the edges belonging to the boundaries between partitions. For example, in Fig. 4.5.a, Node Collapse is blocked between nodes a and c, Edge Splitting and Edge Swapping is also blocked for the edge $\overline{ac}$ delimited by the same nodes. However, note that this is not the case for nodes d and e, while Node Collapse is still blocked between those nodes, Edge Swapping and Edge Splitting is allowed for edge $\overline{de}$, as this would not change the configuration of the edges of the boundaries between partitions. Other operations are also blocked as the Vertex Smoothing and the Node Gliding operations which are not allowed for any of the nodes at the boundary between partitions.

The repartitioning approach was also developed as a solution for the complexity of the remeshing process in parallel. Indeed, the repartitioning process presented in section 4.3.3 exchanges a complete layer of elements between the sending partition and the receiving partition, hence completely changing the boundaries every time it is performed. Figure 4.5.b illustrates how all the blocked operations mentioned above between edges $\overline{ac}$ and $\overline{de}$ are unblocked since these edges no longer belong to the boundary. A new remeshing process can be executed for all elements and nodes involving the previously blocked (now unblocked) edges to complete the remeshing process of the whole domain. The solution for the remeshing process is not the same as if it would be performed in a sequential framework as the different remeshing operations are not performed in the same order. However, the influence of this artifact is expected to be minimal. This will be proven and discussed further in sections 4.4 and 4.5.

4.3.6 Other parallel treatments

At this stage, two particular problems need to be solved. These particular situations correspond more directly to the physical model. However, they may appear for the majority of boundary migration problems modeled with the TRM model. The first is the computation of properties at Shared-Nodes and the second is the Lagrangian movement of these nodes. Within the parallel framework used for the TRM model, Shared-Nodes do not have all the necessary information of their surroundings in the total domain, hence the treatments mentioned above can not be computed in the default sequential way as in chapter 3.

Computation of properties at Shared-Nodes.

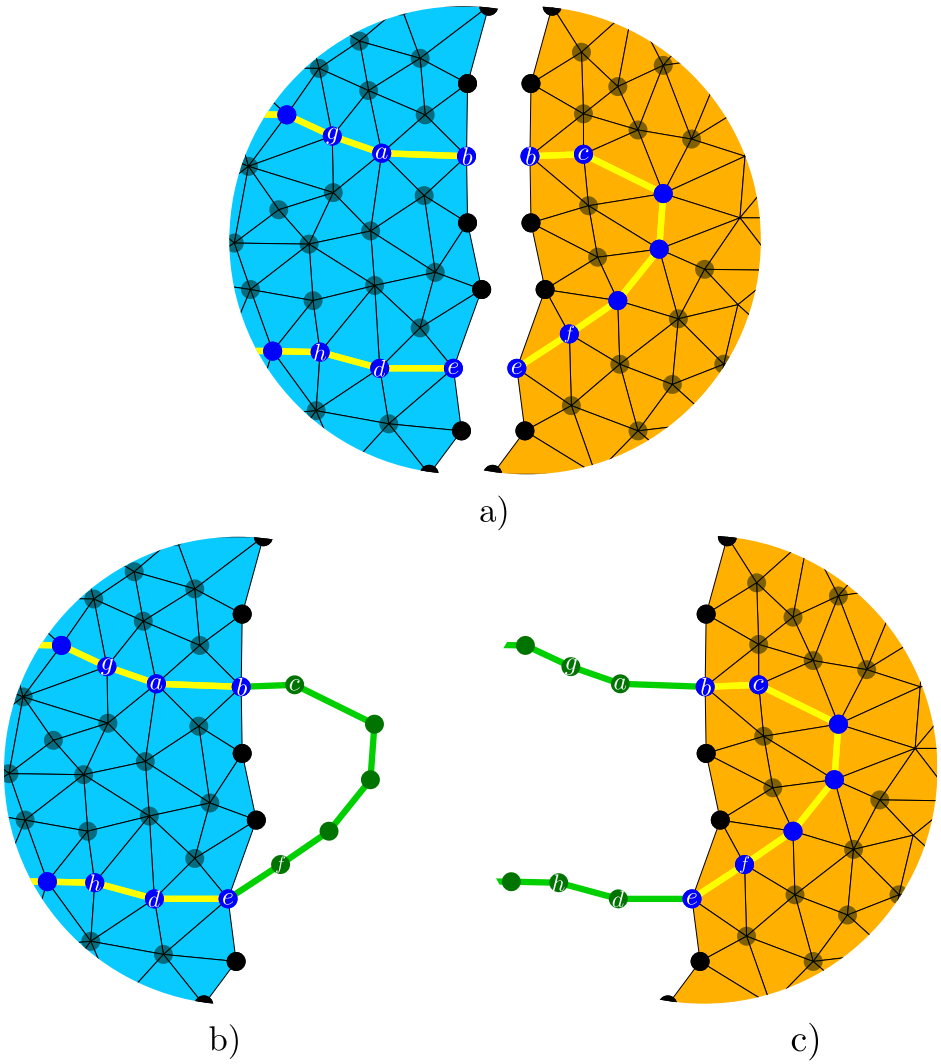


Figure 4.8: Example of Line patch completion with temporary nodes (nodes in green), a) configuration for two partitions, b) $Part_1$ after adding the needed temporary nodes c) $Part_2$ after adding the needed temporary nodes.

For GG mechanism, the necessary geometric properties to be computed are the mean curvature κ and the normal $\vec{n}$ of the interfaces. As mentioned in section 4.2, these properties are computed for L-Nodes using piecewise polynomials of third order (Natural Parametric Splines [196]). Hence the information needed is the position of the nodes contiguous to the Shared-Nodes of each partition. This information is transmitted by the neighboring partitions and stored in *temporary* nodes. Figure 4.8 shows this situation, here the blue nodes correspond to L-Nodes belonging to the partition while the green nodes correspond to the temporary nodes obtained. Geometrical properties can now be computed for the L-Nodes b and e on each partition, obtaining the same result in both of them.

These temporary nodes only exist during the computation of the geometrical properties. A similar operation is done for P-Nodes however only one contiguous temporary node is necessary for the computation of model II of [14] (used for the computation of the points's velocity in our model).

Lagrangian movement in Parallel.

The Lagrangian movement presents a very similar problem as the one exposed in the previous section. In section 3.2.5, it is shown that the Lagrangian movement checks for invalid configurations every time one nodes moves (see Fig. 3.16). In our parallel context, this can not be checked for all the elements when a Shared Node is moved as some of these elements belong to other partitions and do not exist on the memory of the local partition. The solution to this has been implemented as follows: considering that all Shared-Nodes computed the same velocity (using the method of section 4.3.6), each partition makes their respective flipping checking for the elements known to them. If flipping is encountered, this is communicated to all partitions where the Node is shared and the movement is stored for retrying with half the displacement in a further iteration, thus obtaining the same response as it would be in a sequential implementation.

4.3.7 The TRM algorithm for grain growth in parallel

The final sequence for a time step of the TRM model in the context on GG is summarized in Algorithm 8.

Algorithm 8 Grain Growth TRM Algorithm in parallel

1: **Remeshing Algorithm over non-blocked entities** (Algorithm 3 with the constraints explained in sec 4.3.5)
2: **Recompute Ranking System** (section 4.3.3)
3: **Mesh Scattering** (section 4.3.3)
4: **Reconstruct Geometries** (section 4.3.4)
5: **Remeshing Algorithm over non-blocked entities**
6: **Complete Lines and Point Connections** (temporary nodes) (section 4.3.6)
7: **for all** Points: P_i **do**
8: **while** Number of Connections > 3 **do**
9: split multiple point P_i.
10: **for all** Lines : L_i **do**
11: Compute the natural spline approximation of L_i.
12: **for all** L-Nodes : LN_i **do**
13: Compute curvature and normal ($\kappa\vec{n}$) over LN_i.
14: **for all** P-Nodes : PN_i **do**
15: Compute the product $\kappa\vec{n}$ over PN_i using model II of [14].
16: **Delete temporary nodes**
17: **for all** L-Nodes and P-Nodes : LPN_i **do**
18: Compute velocity $\vec{v_i}$ of Node LPN_i
19: **Iterative movement with flipping check in parallel** (section 4.3.6)

4.4 Numerical results

In chapter 3, three test cases corresponding to the Circle-Shrinkage, T-Junction, and Square-Shrinkage tests, were performed in order to validate the accuracy of the TRM model. A final test was performed to validate the model in the context of a more realistic environment, where a 2D 10000 grain test was performed to compare the results to a classical LS-FE approach. Here we will perform 2 sets of 2D simulations over a large number of grains, using a strong and weak scaling benchmark: the first will consider a constant domain size performed with N_p number of processors, this simulation will be denoted to be as "Variable charge" (strong scaling) as the mean charge per processor will evolve inversely proportional to the number of processors. Then, a set of simulations at "Constant Charge" (weak scaling) will be performed for a simulated domain increasing its size proportionally to the number of processors N_p considered. For example, the simulation performed over 2 processors will consider a total domain surface twice as large as the one performed over 1 processor.

All simulations will be performed on a cluster facility composed of *nodes*[1] Bullx R424 equipped with a chip-set Intel Xeon E5-2680v4 with 28 processors at 2.4 GHz. The nodes are connected between them using an Infiniband FDR at a speed of 56 Gb/s. Hereafter we will use the syntax N_nxN_p to describe the configuration used in our computations, where the term N_n describes the number of CPUs used and N_p the number of processors used per CPU, hence, for example for a simulation using 56 processors distributed on 2 CPUs the syntax 2x28 will be used.

4.4.1 Variable Processor Charge (strong scaling benchmark)

Here two similar tests to the one presented in chapter 3 with 10000 initial grains will be performed. The size of the domain will be extended to fit first 50000 and then 560000 grains, maintaining the same initial grain size distribution, the same thermal treatment (1 hour at 1050 $°C$) and identical physical properties: the generation of the initial tessellation will be performed with a Laguerre-Voronoi cell generation procedure [55, 56, 57] over a squared domain and the values for M and γ are chosen as representative of a 304L stainless steel at 1050 $°C$ (with $M = M_0 * e^{-Q/RT}$ where M_0 is a constant $M_0 = 1.56 \cdot 10^{11}\ mm^4/Js$, Q is the thermal activation energy $Q = 2.8 \cdot 10^5\ J/mol$, R is the ideal gas constant, T is the absolute temperature $T = 1323\ K$ and $\gamma = 6 \cdot 10^{-7}\ J/mm^2$) [6, 7]. Moreover, the initial grain size distribution is imposed as a Log-Normal distribution curve with a median value of 0.017 mm and a standard deviation of 0.006 mm.

[1] Here the meaning of the word *node* represents a physical CPU connected in the cluster network.

Additionally, maximal and minimal limits for the size of grains introduced are defined as 0.04 mm and 0.011 mm respectively. With this distribution, one can expect to obtain approximately 1000 grains for every 1 mm^2 of surface. As such, the simulation with 50000 grains fits in a domain with 50 mm^2 and the one with 560000 grains in a domain of 560 mm^2 of surface. After the generation of the Laguerre-Voronoi tessellation, the real number of grains was 52983 and 544913 grains for the domains of 50 and 560 mm^2 respectively.

2D grain growth 50000 Initial grains.

Multiple simulations for the case with 50 mm^2 of surface were performed on 1 (sequential), 2, 4, 10, 20, 28, 56 (2x28) and 84 (3x28) processors, the mesh size parameter and the time step will be held constant and equal to to $h_{trm} = 0.004$ mm and $dt = 10$ s respectively.

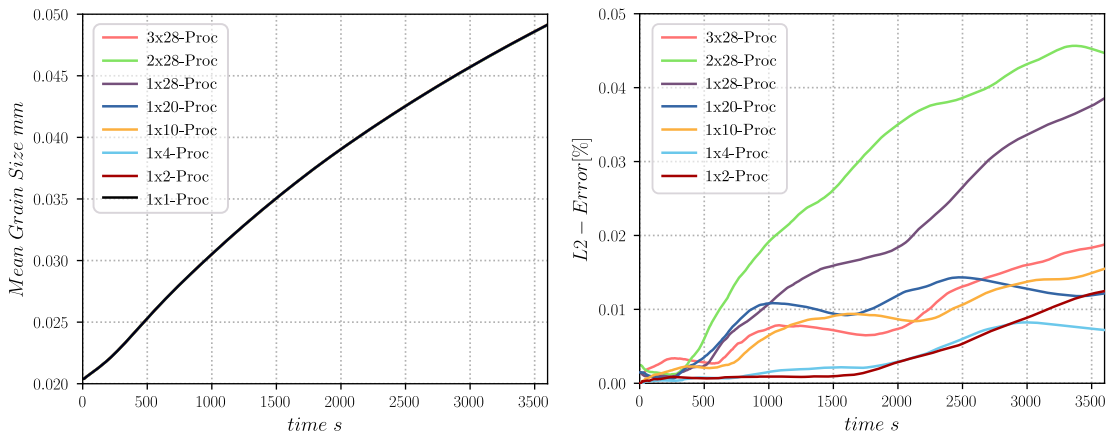


Figure 4.9: Results of the test with a surface of 50 mm^2 performed in 1, 2, 3 and 4, 10, 20, 28, 56 (2x28), and 84 (2x28) processors. Here the mesh size parameter is the same for all runs and equivalent to $h_{trm} = 0.004$ mm and the time step is $dt = 10$ s. Left: Mean grain size evolution, right: L-2 Error of the evolution of the Mean Size with the test performed in sequential (1 processor) as a reference.

Figure 4.9 gives the results for the evolution of mean grain size pondered by surface and the L2-Error (measured with respect to the sequential case) for the test with 50 mm^2 of surface. Here, all parallel tests behave almost exactly like the sequential one (1 processor). The L2-Error is found to be lower than 0.05 % concerning the test performed in sequential. Similarly, Fig. 4.10 shows the evolution of the mean grain size distributions performed in 1, 28, 56 and 84 processors. Once again the results are almost identical for all tests. This is illustrated in Fig. 4.11 which shows that the error over the grain size distributions in surface is in all cases lower than 1.0 %. Furthermore, Fig. 4.12 shows the CPU-time needed for all the configurations to accomplish the test. Here, the parallel tests

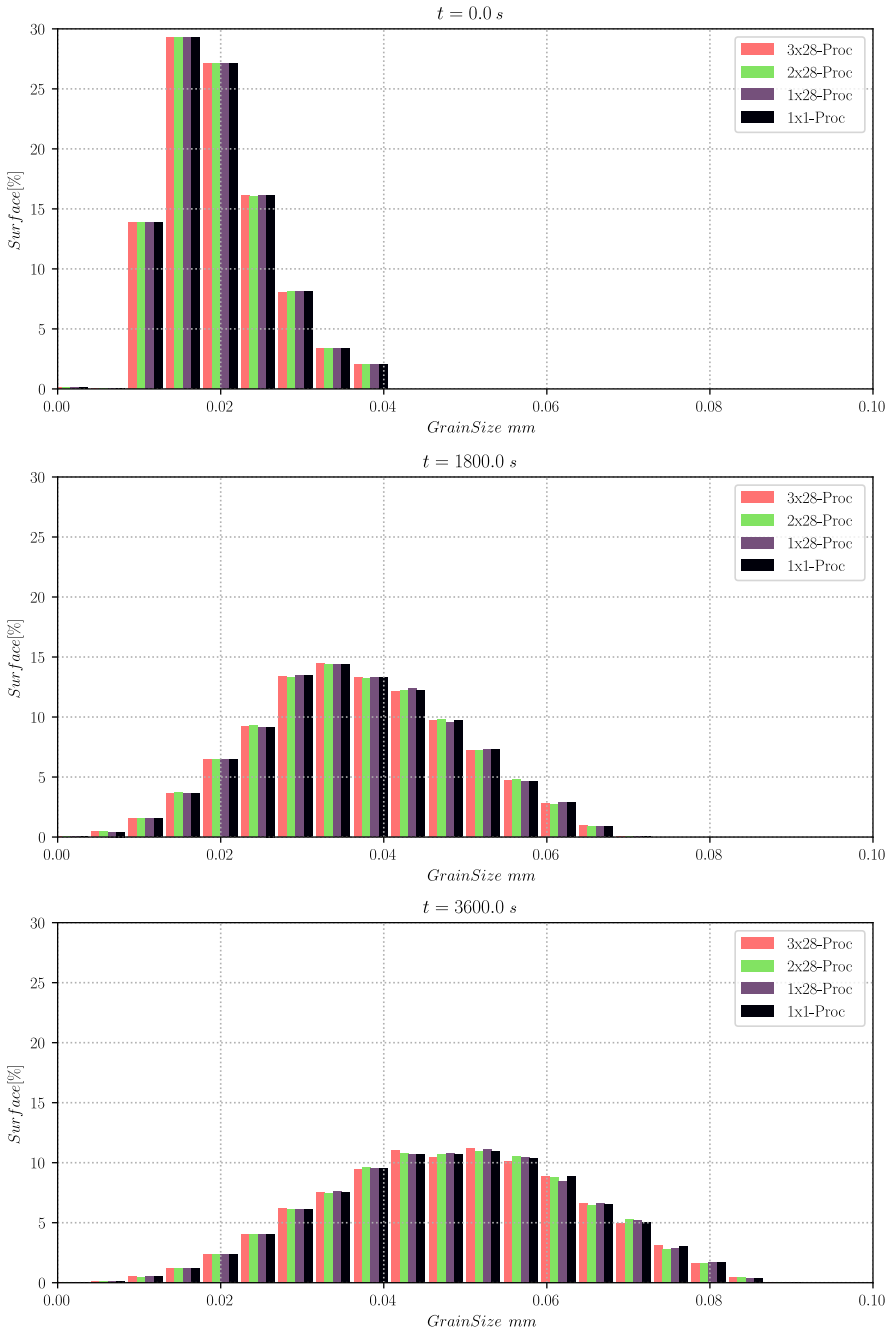


Figure 4.10: Results of grain size distribution for the test with a surface of 50 mm^2 performed in 1, 28, 56 (2x28) and 84 (3x28) processors. Initial state (top), distributions after 1800 s (center), distributions after 3600 s (bottom).

obtained a speed-up[2], being the test with 56 processors the fastest one contrary to the expected behavior, where the fastest test should be the one performed

[2]The speed up in the *Variable Charge* test case (strong scaling benchmark) is obtained by dividing the CPU-time of the sequential case by the CPU-time of the parallel case in question.

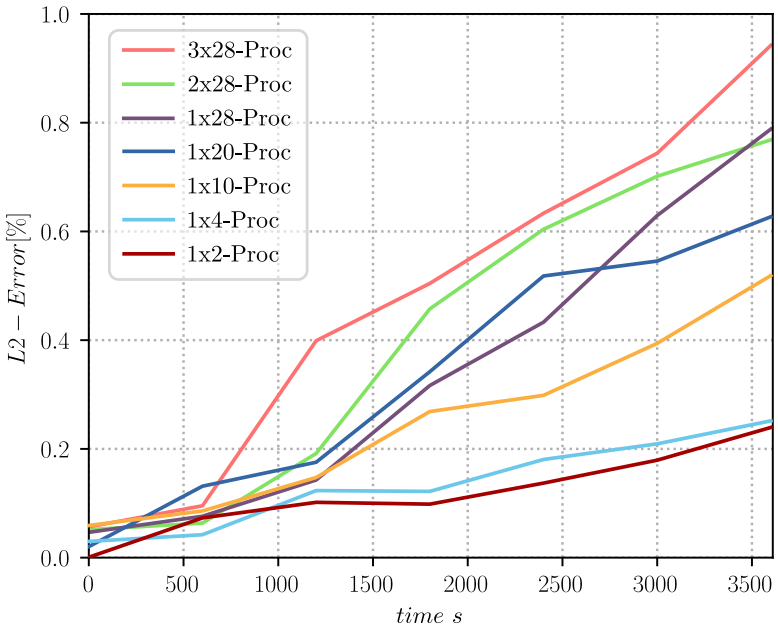


Figure 4.11: L2-Error over the grain size distribution for the test with a surface of 50 mm^2 performed in 2, 4, 10, 20, 28, 56 (2x28) and 84 (3x28) processors compared to the simulation performed in 1 processor.

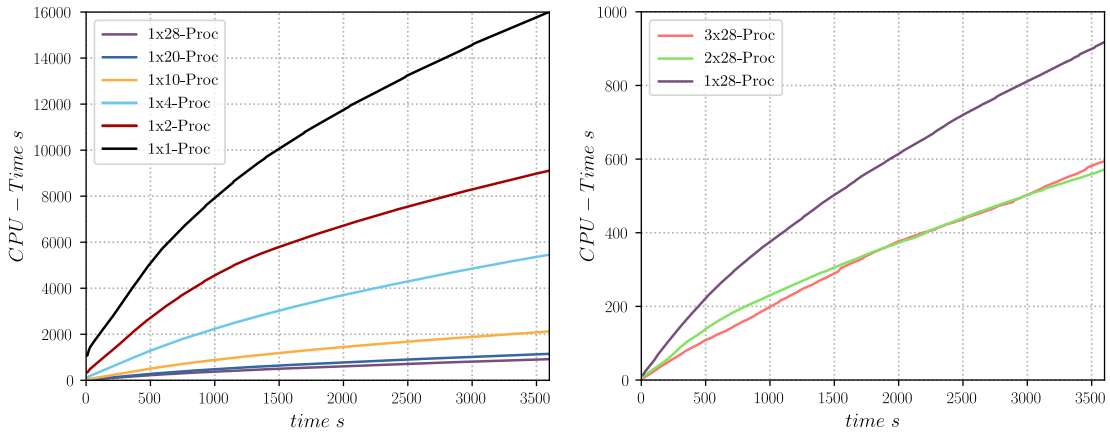


Figure 4.12: CPU-times for the test with a surface of 50 mm^2 performed in 2, 4, 10, 20 and 28 processors (left) and 28, 56 (2x28) and 84 (3x28) processors (right).

with 84 processors. In fact, the simulation performed over 84 processors may be over-partitioned, meaning that the number of partitions is too high compared to the number of elements in the simulation.

Figure 4.13 illustrates 2 examples of the evolution of elements on each partition for 2 cases: 2 processors and 4 processors. Note that in this case (the parallel case, contrary to the sequential one), the response is not a curve but a range of data in the Y axis, as the amount of Elements on each partition is changing ev-

ery time the mesh scattering is performed. Note that this range is higher when the number of processors increases, as the boundaries between partitions present more changes, however for this two examples this range is well contained around 1/2 and 1/4 of the total number of elements, for the simulation performed with 2 and 4 processors respectively. The re-equilibrium of charges here is optimal. Figure 4.14 gives the mean number of elements and its range for the simulations performed in 10, 20, 28, 56 and 84 processors, here two plots are given: for the simulations performed in one CPU (one node) and in multiple CPUs (multiple nodes).

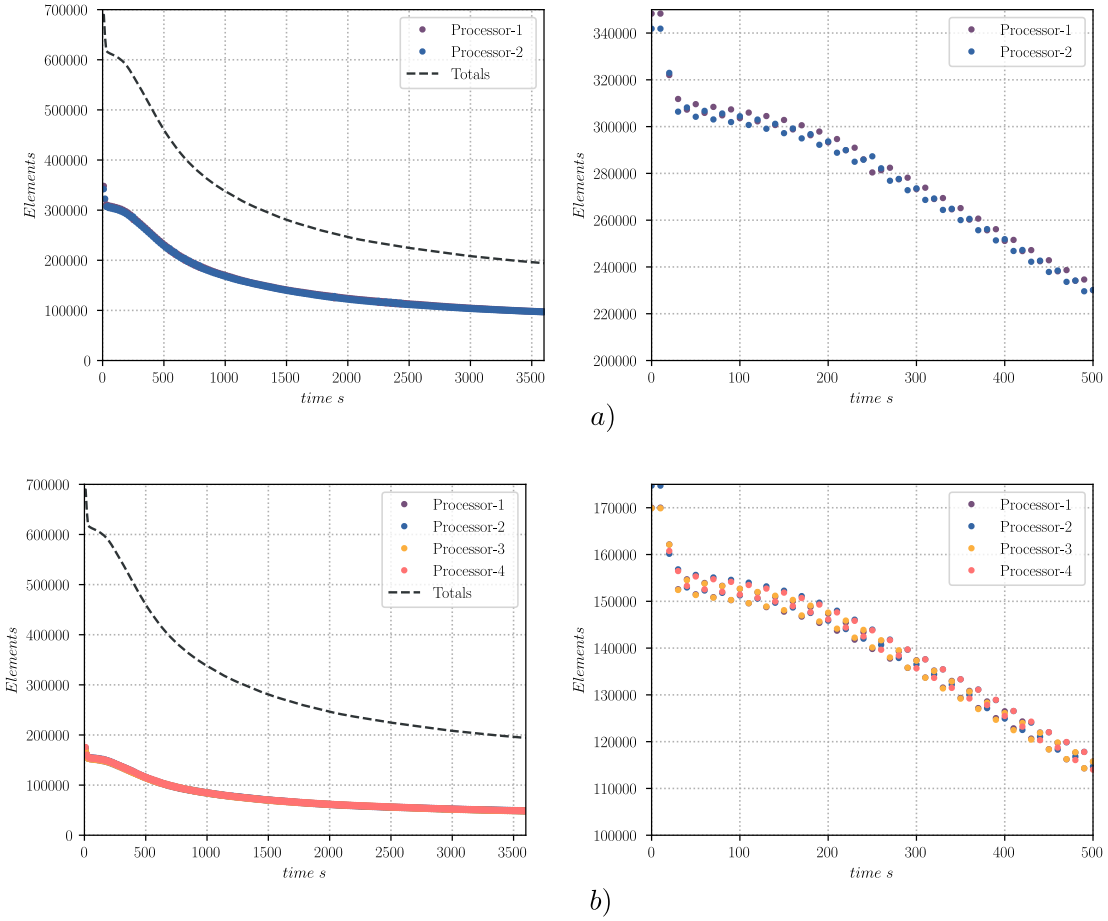


Figure 4.13: Number of elements per processor for the for the test with a surface of 50 mm^2 performed in 2 and 4 processors compared to the total number. The evolution of the number of elements (left) and a zoom at the begining of the simulation is shown (right). a) 2 Processors, b) 4 processors.

One interesting index to follow is the value obtained by dividing the range of elements over its mean value, hereafter called the index $EROM$. In parallel computations, this index has been plotted in Fig. 4.15 and shows how many of the elements may be scattered relative to the mean number of elements on each

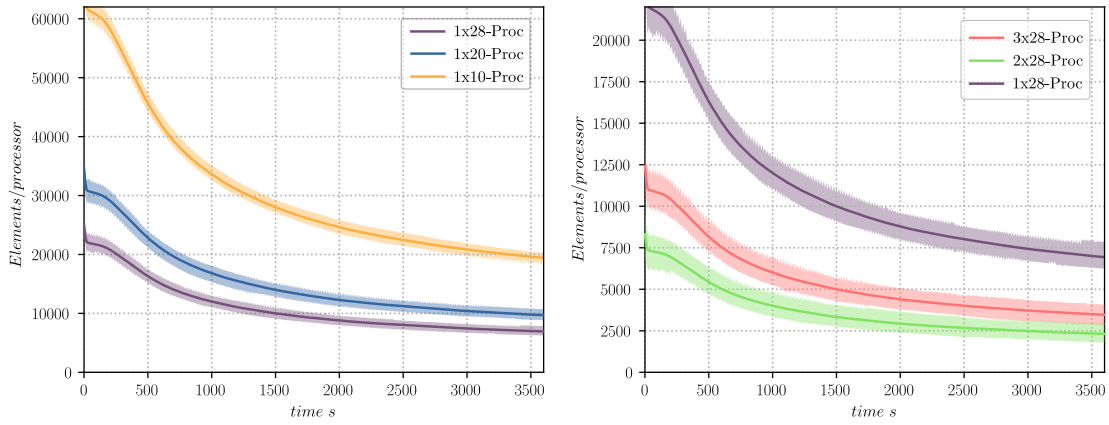


Figure 4.14: Mean number of elements per processor for the test with a surface of 50 mm^2 performed in 10, 20, 28, 56 (2x28) and 84 (3x28) processors. The evolution of the number of elements for the simulations contained in one node (left) and in multiple nodes (right) is shown. The range for the number of elements of all processors in the same simulation is shown in the same color with an alpha component.

partition. The index $EROM$ reflects the maximum limit percent of elements being transported from one partition to another by the Mesh Scattering procedure presented in section 4.3.3, hence one can use it to study the global efficiency of the numerical procedure. Here we define the efficiency of one increment i for a simulation with N_p number of processors as follows:

$$Efficiency = \frac{t_1^i}{t_{N_p}^i \cdot N_p} \tag{4.1}$$

where the term $t_{N_p}^i$ determines the CPU-time needed to perform increment i in a simulation with N_p number of processors. This equation computes the number of resources needed for an increment of a parallel simulation compared to a sequential one. Figure 4.16 plots the Efficiency of the simulation for the case with 50000 grains against the mean number of elements (left) and against our $EROM$ index (right). Of course the lower the efficiency per increment for a given simulation, the lower we expect to be the speed-up of the parallel simulations. The mean efficiency of the 84 processors test (approximately 0.29) is much lower than the mean efficiency of the 56 processors test (approximately 0.48), thus a reduction of the efficiency of 65% for an increase in the number of processors of 50% (from 56 to 84 processors). These results suggest that the simulation performed with 84 processors is indeed over partitioned and that one should aim to obtain a mean number of elements not lower than 10000 or an $EROM$ index not higher than 30% (The real impact of the $EROM$ index will be studied in the case at a constant charge). Note, however, that the relative overhead obtained by the high number of partitions is not higher than the reduction in CPU-time obtained, as the speed-up is still greater than 1 even for the simulation with 84

processors.

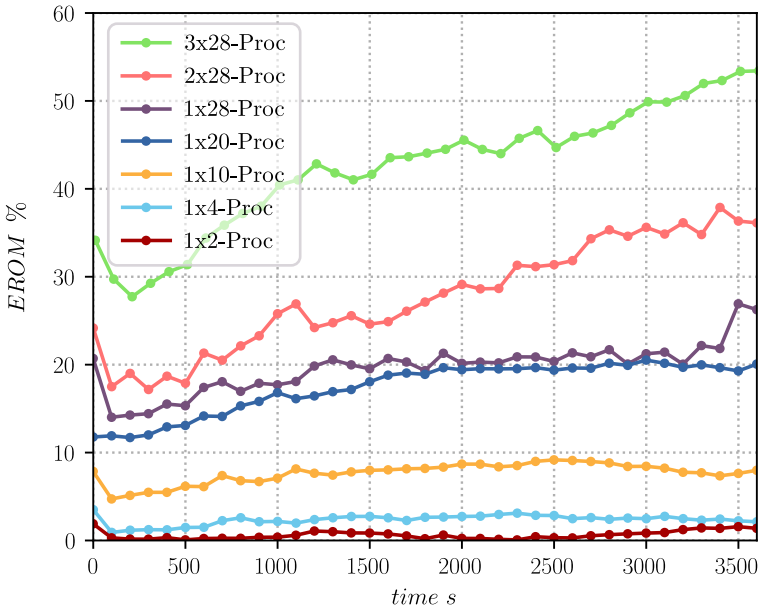


Figure 4.15: Range of the number of elements over the Mean number of elements per processor (index *EROM*) for the test with a surface of 50 mm^2 performed in 10, 20, 28, 56 (2x28) and 84 (3x28) processors. The percent value increases with the number of processors plotted every 10 increments.

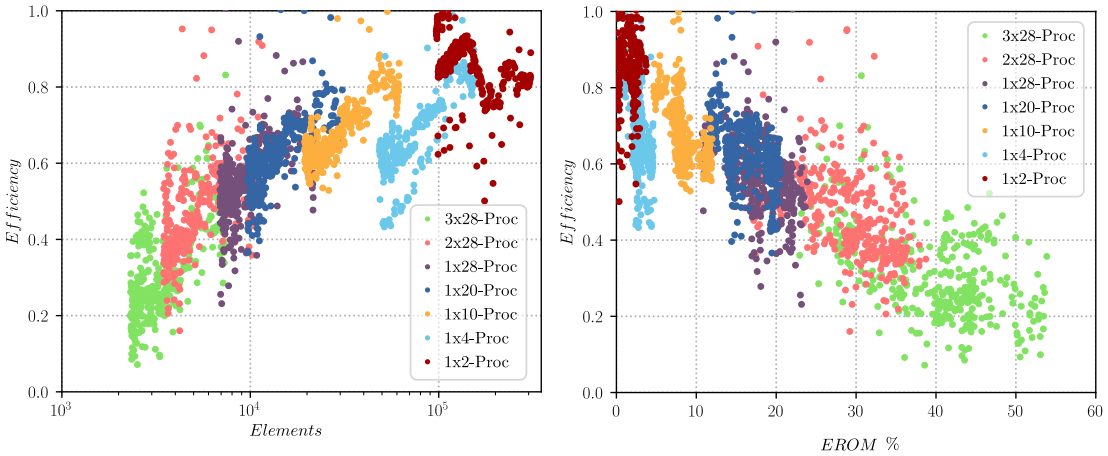


Figure 4.16: Efficiency of the simulation for the case with a surface of 50 mm^2 against the mean number of elements (left) and against the *EROM* index (right).

Finally, Fig. 4.21 plots the speed-up of the parallel implementation of the TRM model against the number of processors, here the reference for the optimal speed-up is also shown. Of course, the optimal speed-up can not be obtained

in our case as the operations produced by the communication create additional overhead.

2D grain growth 560000 Initial grains.

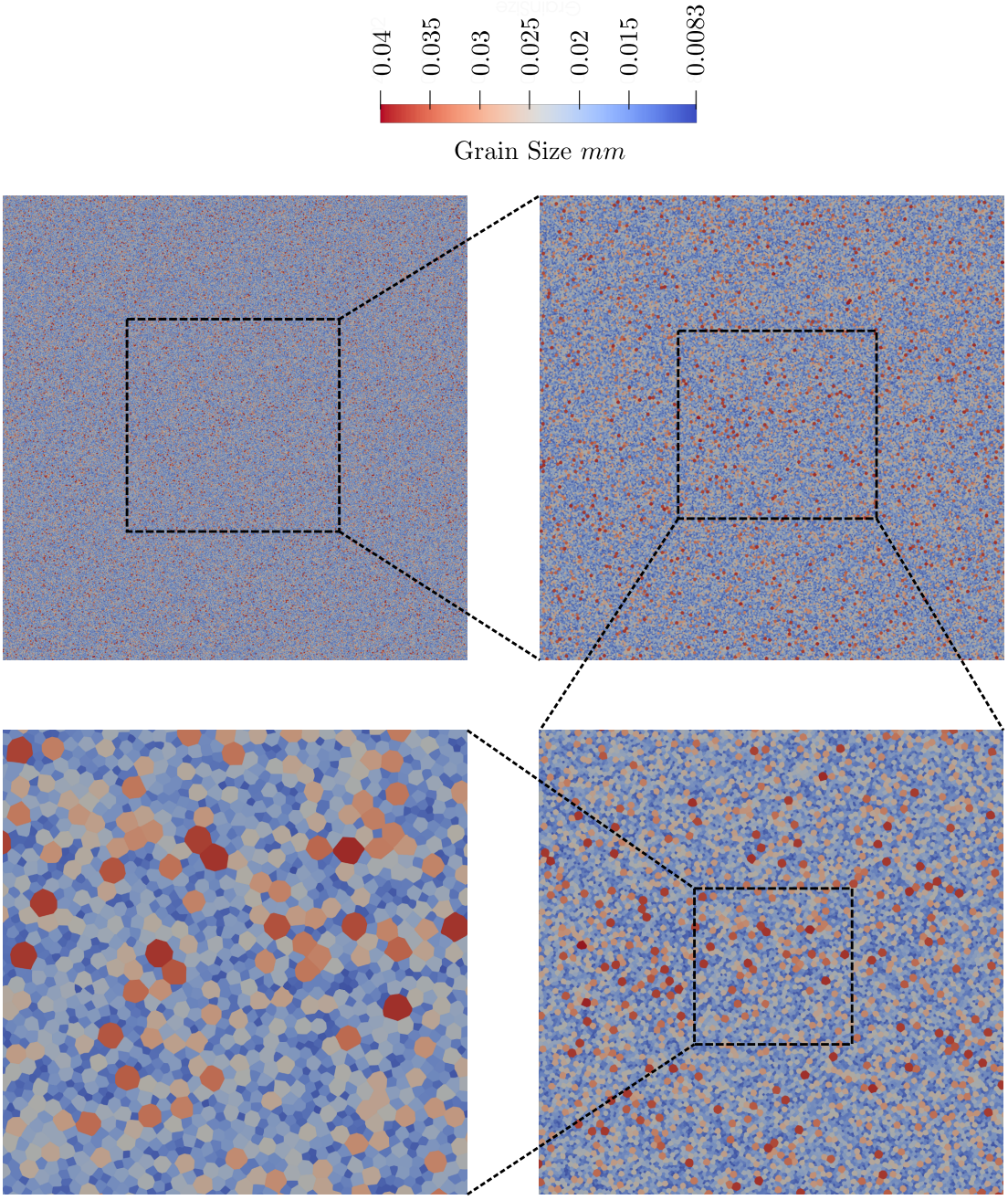


Figure 4.17: Initial state of the case with a surface of 560 mm^2, 544913 grains are shown in the bigger image. Subsequent zoom views are given.

Here a similar GG test will be performed but the surface of the domain will

be increased to $560mm^2$. The test will be performed on 1, 14, 28, 56 (2x28), 84 (3x28), 112 (4x28) and 140 (5x28) processors. Figure 4.17 illustrates the initial microstructure for this case. To the knowledge of the author, this is the maximum amount of grains that have been simulated using 2D unstructured FE meshes, and the second largest simulation in the context of GG, the first being the one presented in [152] for a microstructure with 671000 initial grains, but in a FFT context (thus using regular grids).

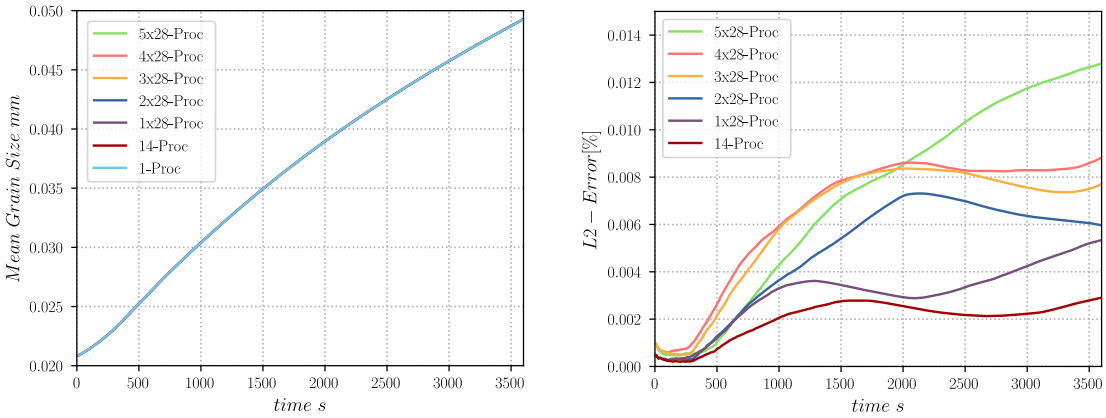


Figure 4.18: Results of the case with a surface of 560 mm^2 performed in 1, 14, 28, 56 (2x28), 84 (3x28), 112 (4x28) and 140 (5x28) processors. Here the mesh size parameter is the same for all runs and equivalent to $h_{trm} = 0.004\ mm$ and the time step is $dt = 10\ s$. left: Mean grain size evolution, right: L-2 Error of the evolution of the Mean Size with the test performed in sequential (1 processor) as a reference.

Figure 4.18 gives the results for the evolution of mean grain size and its L2-Error for the test with 560 mm^2 of surface. Similarly, Fig. 4.19 plots the L2-Error over the grain size distribution of the present test. The range of both L2-Errors has been reduced by at least 1/5 compared to the tests with 50 mm^2 of surface. At this level, we consider that the influence of the parallel TRM model over the precision of the simulation is quasi-non-existent, while for the test with 50 mm^2 of surface this influence is very low.

By increasing the simulated domain size, the index $EROM$ should be reduced as the mean number of elements per processor is increased. Figure 4.20 shows the evolution of the index $EROM$ for the present test, here the evolution of this index has been reduced by more than half (considering the simulation with 3x28 in both the 50000 grains and the 560000 grains case) which maintains the efficiency of a time step over 0.4 accordingly to Fig. 4.16.

Figure 4.21 illustrates the evolution of the speed-up of both test cases. This result shows that the speed-up is relatively the same up to 56 (2x28) processors

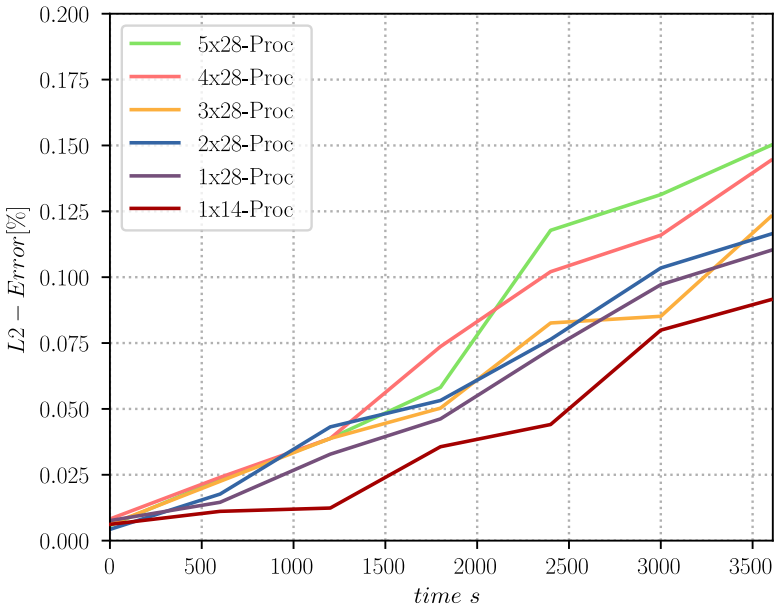


Figure 4.19: L2-Error over the grain size distribution for the case with a surface of 560 mm^2 performed in 14, 28, 56 (2x28), 84 (3x28), 112 (4x28) and 140 (5x28) processors compared to the simulation performed in 1 processor.

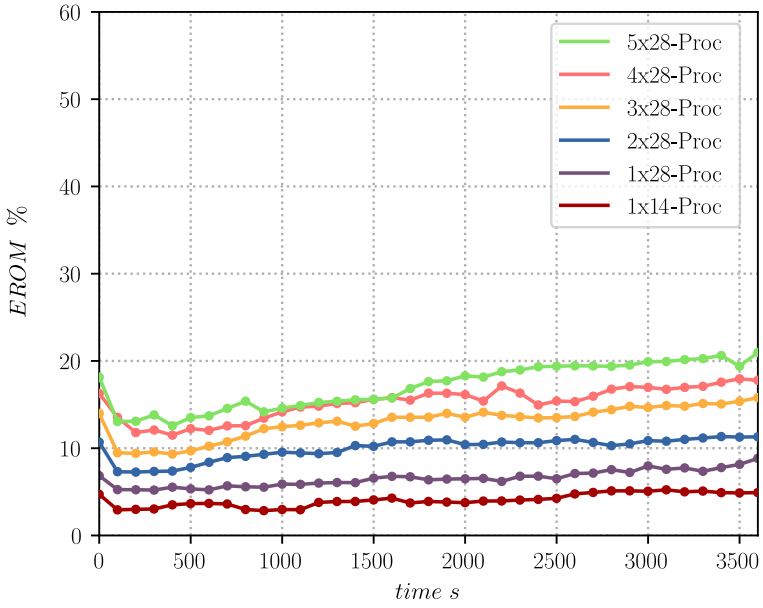


Figure 4.20: Range of the number of elements over the Mean number of elements per processor (index $EROM$) for the test with 560 mm^2 of surface performed in 14, 28, 56 (2x28), 84 (3x28), 112 (4x28) and 140 (5x28) processors. The percent value increases with the number of processors plotted every 10 increments.

but it diverges for a higher number of processors in favor of the simulation with 560 mm^2 of surface. The maximum speed-up obtained was for the case with a

surface of 560 mm^2 performed over 140 (5x28) processors for which a speed-up of 48.5 was obtained, this simulation took 38 minutes and 45 seconds while the one performed in 1 processor took 31 hours and 20 minutes. The total number of grains at the end of this simulation was of 117157 grains, hence 21.5 % of the total initial number of grains while for the simulation with 50 mm^2 of surface was of 10399, hence 19.6 % of the total initial number of grains.

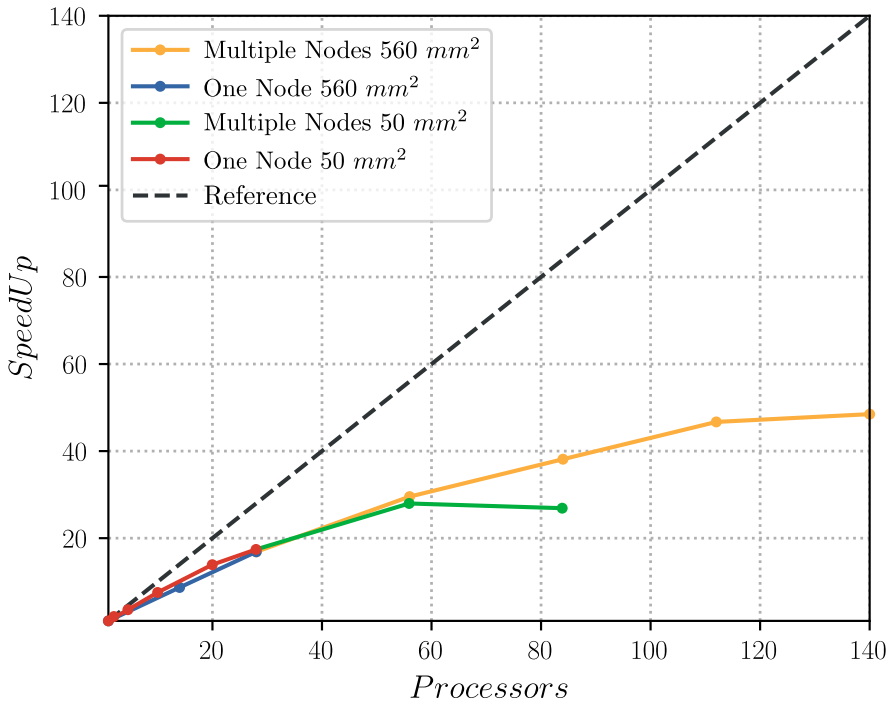


Figure 4.21: Performance of the TRM model in parallel for a variable processor charge: the domain surface is maintained constant while the number of processors increases, two test were performed, one with a surface of 50 mm^2 and the second with a surface of 560 mm^2. The data is compared to the optimal speed up (Reference).

4.4.2 Constant Processor Charge (weak scaling benchmark)

When performing a benchmark on a strong scaling context, the amount of memory that a processor has to maintain decreases when the number of processors increases. Of course in the same context, when performing a sequential simulation all the memory has to be maintained by only one processor, making it longer to read or to add pieces of information to the data set of the running process. This artifact makes it, that the optimization made by the parallel implementation in a strong scaling context, be both, for memory management (to access and to write in a given memory location) and in number of operations (such as remeshing or moving nodes) to perform by a processor. On the other

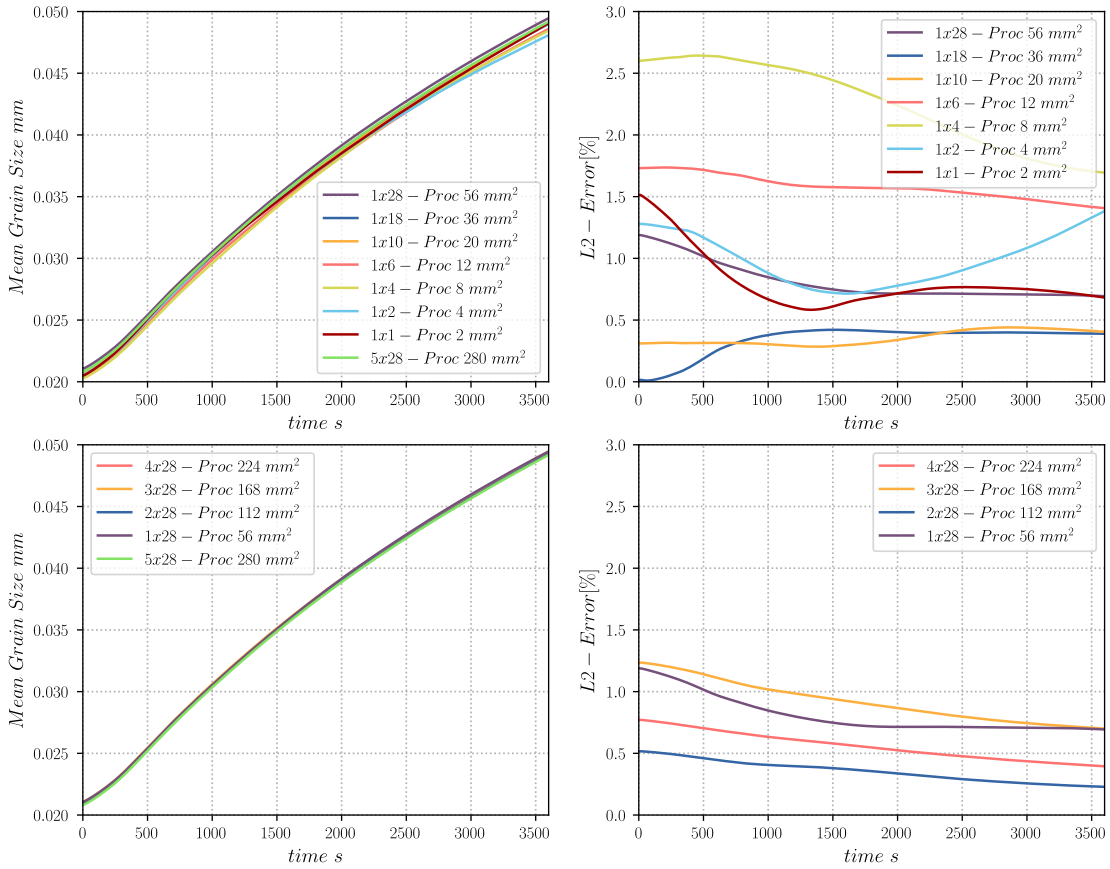


Figure 4.22: Mean size results of the case with a DSPP of 2 mm^2 performed in 1, 2, 4, 6, 10, 18, 28, 56 (2x28), 84 (3x28), 112 (4x28) and 140 (5x28) processors. The plot has been divided in two plots for a clearer visibility: top: simulations performed over one Node (1, 2, 4, 6, 10, 18, 28), bottom: simulations performed over multiple Nodes (28, 56 (2x28), 84 (3x28), 112 (4x28), 140 (5x28)). Each plot is given its respective L2-Error to the response of the largest simulation in this context (140 (5x28) processors).

hand, a benchmark at constant processor charge aims to measure the speed-up[3] generated by a parallel implementation only on the number of operations to be performed inside a processor, as the memory to be maintained by every processor is constant when the number of processors increases. While it is not possible to maintain an equally distributed and constant charge on all processors here, it is possible to approach the concept by increasing the size of the simulated domain proportionally to the number of processors used in a parallel simulation with the TRM model. Two sets of simulations will be performed: the first with a domain surface per processor (DSPP) of 2 mm^2 (approximately 2000 grains per

[3]contrary to the speed up in the *Variable Charge* test case (strong scaling benchmark), the speed up for the *Constant Charge* test case is obtained by multiplying the CPU-time of the sequential case with the number of processors of the parallel case, divided by the CPU-time of the parallel case.

processor) and the second with a DSPP of 4 mm^2 (approximately 4000 grains per processor). Moreover, even though the same grain size distribution is used in the generation of the Laguerre-Voronoi tessellation, it is not possible to obtain a perfectly equal statistical distribution for different sizes of domains. As such, at the beginning of the simulations, small variations on the mean grain size or the grain size distributions may appear.

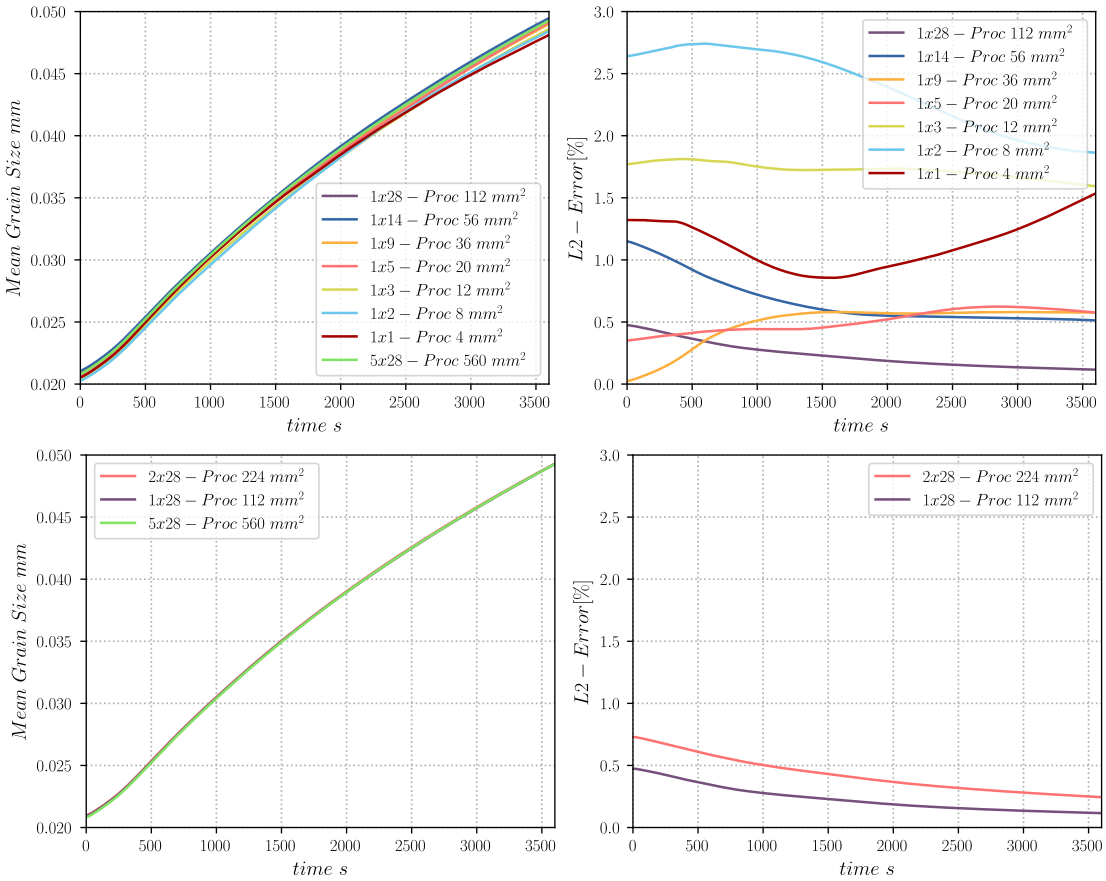


Figure 4.23: Mean size results of the case with a DSPP of 4 mm^2 performed in 1, 2, 3, 5, 9, 14, 28, 56 (2x28) and 140 (5x28) processors. The plot has been divided in two plots for a clearer visibility: top: simulations performed over one Node (1, 2, 4, 6, 10, 18, 28), bottom: simulations performed over multiple Nodes (28, 56 (2x28), 140 (5x28)). Each plot is given its respective L2-Error to the response of the largest simulation in this context (140 (5x28) processors).

Figure 4.22 and 4.23 give the results for the evolution of the mean size of the cases with a DSPP of 2 mm^2 and 4 mm^2 respectively. Furthermore, L2-Error plots are also given corresponding to the difference of the simulations performed in parallel to the reference cases (here chosen as the largest simulation of each context which was performed in 140 (5x28) processors: 280 and 560 mm^2 respectively for the cases with a DSPP of 2 mm^2 and 4 mm^2). In both cases, the

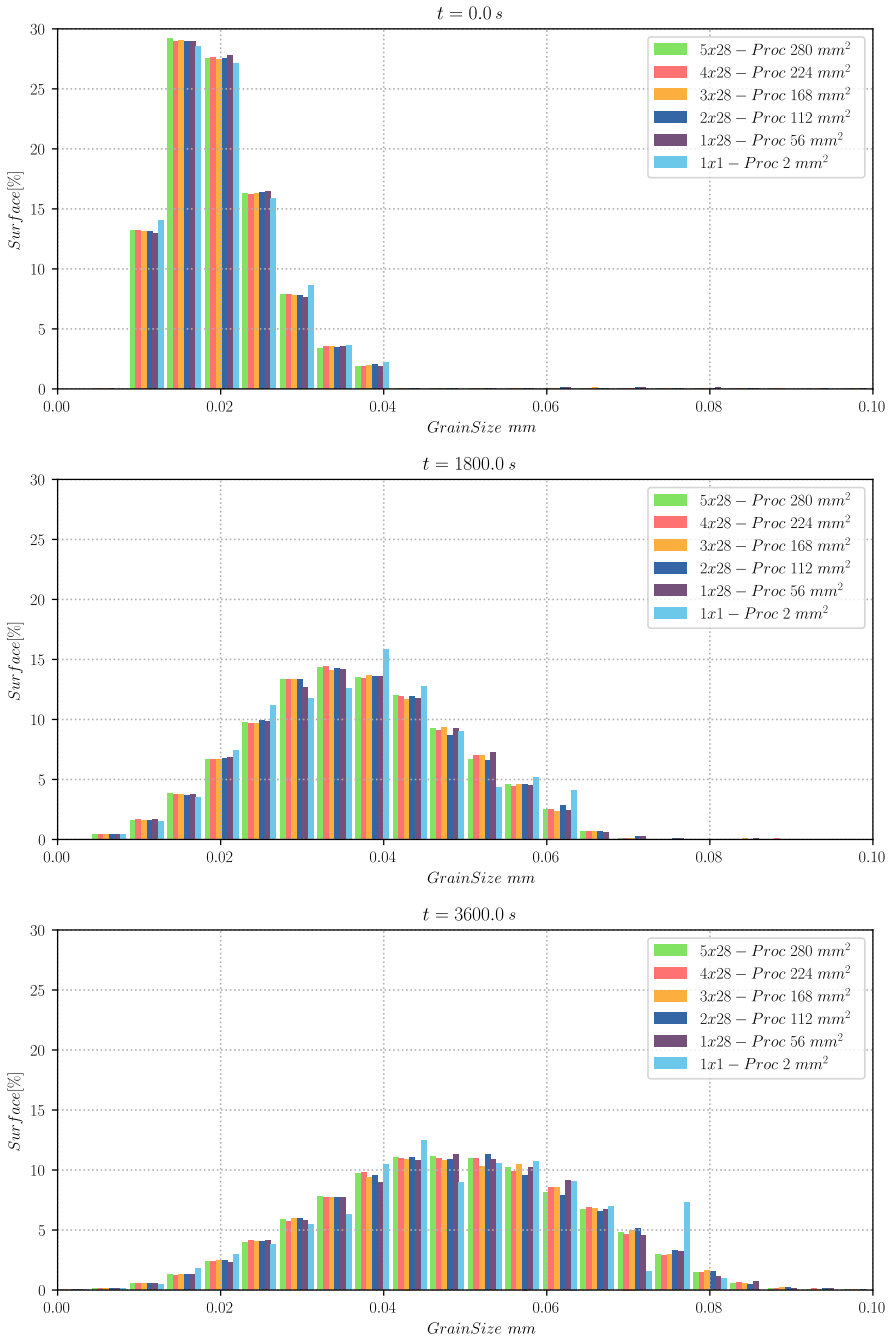


Figure 4.24: Results of grain size distribution for the test with a DSPP of 2 mm^2 performed in 1, 28, 56 (2x28), 84 (3x28), 112 (4x28) and 140 (5x28) processors. Initial state (top), distributions after 1800 s (center), distributions after 3600 s (bottom).

curves appear to be very similar to their references with an error inferior to 3%. Note that the L2-Error have a tendency to decrease when the simulation domain increases (as expected), and that the error after a total surface of 56 mm^2 (ap-

proximately 56000 initial grains) is inferior to 1%.

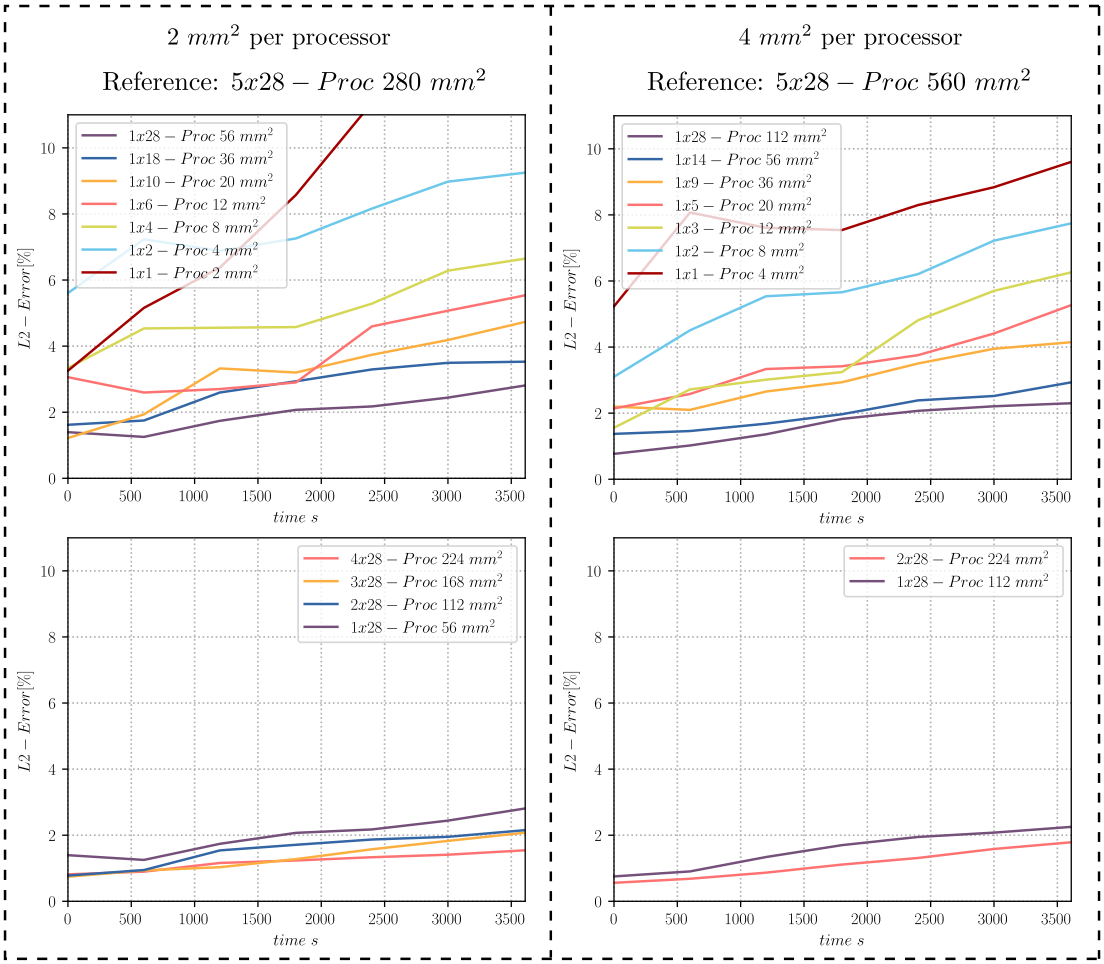


Figure 4.25: L2-Error of the grain size distribution for the test with a DSPP of 2 mm^2 (left) and a DSPP of 4 mm^2 (right) compared to of the largest simulation of each context (140 (5x28) processors) (280 and 560 mm^2 respectively). Each plot have been divided in two for a clearer visibility: top: simulations performed over one Node, bottom: simulations performed over multiple Nodes.

Figure 4.24 illustrates the evolution of the mean size distribution in surface for one set of simulations with a DSPP of 2 mm^2, at the beginning, after 1800 s and after 3600 s of simulated time, here only the simulations performed in 1, 28, 56, 84, 112, and 140 processors are plotted for a clearer visualization. Figure 4.25 gives the evolution of the L2-Error over the grain size distributions for both sets of simulations with a DSPP of 2 mm^2 and of 4 mm^2 and for all the simulations performed. Similarly to the evolution of the mean grain size, the L2-Error is obtained to be lower than 3% after a domain size of 56 mm^2, suggesting that our microstructure can be statistically well represented by a simulation with over 56000 initial grains. On the contrary, simulations performed with a domain of

20 mm^2 and below, show an error at the end of the simulation of more than 5% which suggest that they contain too few grains (at the end of the simulation) or that statistical results may be highly influenced by the boundary conditions, simulations with lower than 20000 grains should be avoided in our GG context.

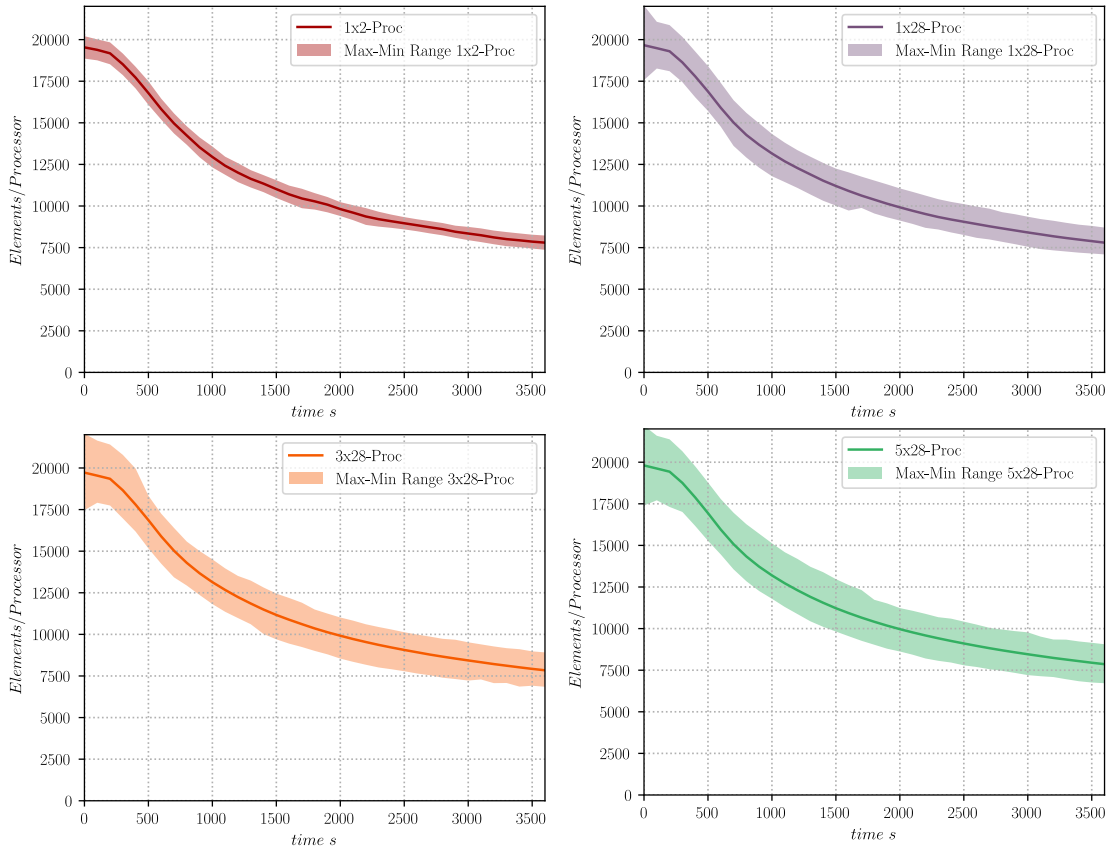


Figure 4.26: Mean number of elements per processor for the test with a DSPP of 2 mm^2 performed in 2, 28, 84 (3x28) and 140 (5x28) processors. The range for the number of elements of all processors in the same simulation is shown in the same color with an alpha component.

Figure 4.26 illustrates the evolution of the number of elements for the simulation with a DSPP of 2 mm^2, the mean number of elements is very well maintained for all processors although their range in the Y axis increases for the simulations with a high number of processors, similarly to the results obtained for the test with a variable processors charge. The evolution of the $EROM$ index is presented in Fig. 4.26 for both sets of simulations, where the sets using a DSPP of 4 mm^2 have a lower overall value, this suggests that the efficiency of these simulations should be higher than the efficiency of those with a DSPP of 2 mm^2.

Figures 4.28 and 4.29 show the CPU-time and the speed-up of both sets of simulations. Note than in Fig. 4.28, the scales have been adapted so the reference

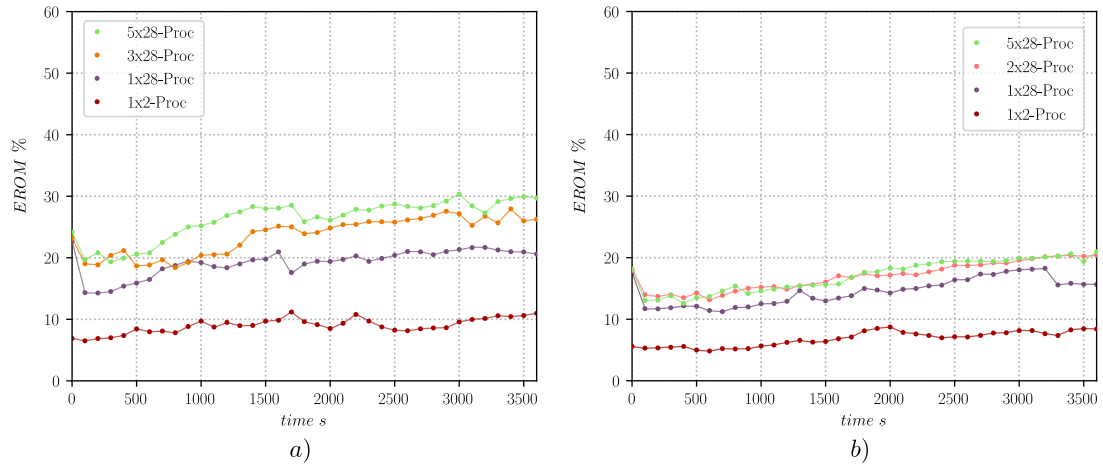


Figure 4.27: Range of the number of elements over the Mean number of elements per processor (index *EROM*) for the test with a DSPP of: a) 2 mm^2 and b) 4 mm^2 per processor.

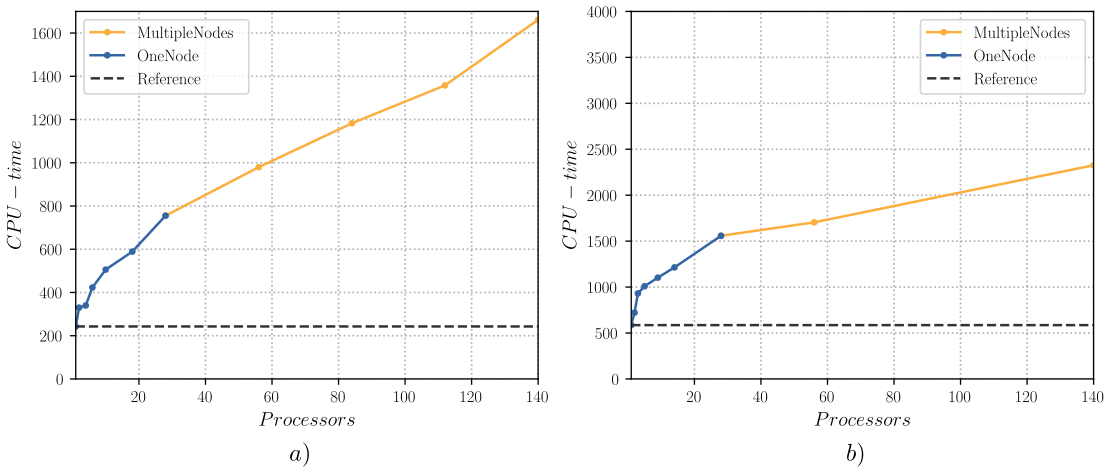


Figure 4.28: CPU-times at the end of the simulation for the test with a DSPP of: a) 2 mm^2 and b) 4 mm^2 per processor.

curve (the value of the CPU-time for the sequential simulation) be at the same height for both the simulations with a DSPP of 2 mm^2 (Fig. 4.28.a) and of 4 mm^2 (Fig. 4.28.b), hence allowing a better comparison of the relative CPU-time needed for all simulations when compared to their respective reference. Clearly, the simulations with a DSPP of 4 mm^2 need a relatively lower CPU-time to achieve the end of the simulations. This can also be seen in Fig. 4.29: for the simulations run in one node (one single CPU with a maximum 28 processors) the speed-up is very similar, however above 28 processors (for multiple nodes) the speed-up favors the simulations with a higher DSPP.

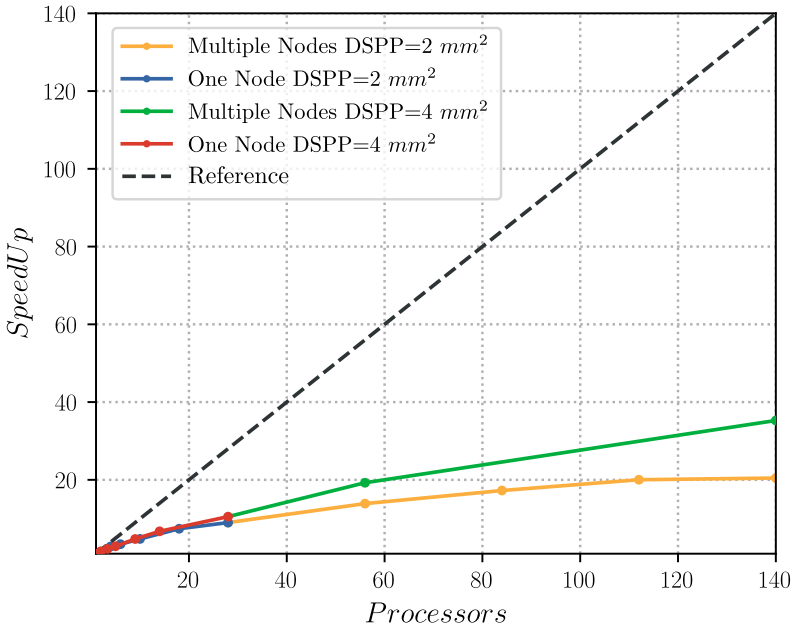


Figure 4.29: Performance of the TRM model in parallel for a constant processor charge: the domain surface increases proportionally to the number of processors, two test were performed, one with a DSPP of 2 mm^2 (approximately 2000 grains per processor) and the second with a DSPP of 4 mm^2 (approximately 4000 grains per processor). The data is compared to the optimal speed up (Reference).

4.5 Discussion and conclusion

The parallel implementation of the TRM method employs several sub-algorithms, each one of them addressing a different problem: the partitioning of the domain, using the dual graph of the initial mesh with the open-source library Metis, and the re-numbering scheme developed to maintain the coherence between partitions generated by the initial partitioning. Similarly, the *mesh scattering* algorithm allows a re-equilibration of the charges between partitions. The remeshing is performed in a 2-step remeshing sequence by blocking the boundaries between partitions (which are dynamic, following the mesh scattering algorithm) and the Lagrangian movement has been adapted to a parallel context. Predictions were validated in parallel compared to sequential simulations.

Two sets of test cases were studied. The first characterized the performance of the TRM model at "variable processor charge" (strong scaling benchmark). The accuracy of our model when performed in sequential and parallel was studied, obtaining the same response with negligible errors in every test. The speed-up for these simulations was shown to be dependent on the mean number of elements and on the "Element Range over Mean" index $EROM$. It was observed that for simulations performed with a high number of processors the speed-up may not be satisfactory if the number of elements is too low or if the $EROM$ index is too high. For a simulation with 600000 initial elements (52983 initial grains) performed over 84 processors, the speed up was lower than for the same simulation performed over 56 processors. These results suggest that this simulation performed with 84 processors is indeed *too* partitioned, and that one should aim to obtain a mean number of elements not lower than 10000 or a $EROM$ index higher than 30%. Then a test with 544913 initial grains was performed obtaining good results in terms of speed-up for all simulations. The maximum speed-up obtained was for the case with a 544913 initial grains, performed over 140 (5x28) processors for which a speed-up of 48.5 was obtained, this simulation took 38 minutes and 45 seconds to perform 360 increments while the one performed in 1 processor took 31 hours and 20 minutes. The total number of grains at the end of this simulation was of 117157 grains, hence 21.5 % of the total initial number of grains while for the simulation with 52983 initial grains was of 10399, hence 19.6 % of the total initial number of grains.

Tests at "constant processor charge" (weak scaling benchmark) were performed with a simulated domain increasing its size proportional to the number of processors considered. Here a wide range of simulations were performed divided into two sets: the first with a domain surface per processor (DSPP) of 2 mm^2 (approximately 2000 grains per processor) and the second with a DSPP of 4 mm^2 (approximately 4000 grains per processor). The results of these simulations showed that the speed-up obtained for the simulations with a DSPP of 4 mm^2 was higher than for the simulations with a DSPP of 2 mm^2 as a relatively higher

EROM index was found for the latter.

Another observation of the tests at "constant processor charge" was that the L2-Error over the grain size distributions is obtained to be lower than 3% after a domain size of 56 mm^2, suggesting that our microstructure can be statistically well represented by a simulation with over 56000 initial grains. On the contrary, simulations performed with a domain of 20 mm^2 and below, show an error at the end of the simulation of more than 5% which suggest that they contain too few grains (at the end of the simulation, 3954 grains for the case of with 20000 initial grains) or that statistical results may be highly influenced by the boundary conditions (in our case, orthogonality of grain boundaries with the domain limits). Thus, it seems that simulations with lower than 4000 grains at any time during the simulation should be avoided when studying statistical values of a predicted (simulated) microstructure in our GG context. This also enforces the argument behind the studies aiming to increase the performance of massive multidomain simulations, as the main obstacle to increasing the number of domains in such simulations is given by their high computational cost.

In the context of massive multidomain simulations, to the knowledge of the authors, only two methodologies in the literature have attempted to perform simulations with hundreds of thousands of domains: the one in [153] with a maximum number of 100000 grains and the one in [150, 152] with 671000. Results in [153] showed a better speed-up of their parallel implementation than the one presented in this chapter for our TRM model. Concerning the CPU-time, the parallel strategy presented in [153] was able to perform 20 increments of a simulation with 100000 initial grains in 32 s when performed in 128 processors. In [152] no data concerning the speed-up was provided, however, it was mentioned that the simulation with 671000 initial grains was performed until less than 4000 grains remained, with a total CPU-time between 9 and 12 days, when running on 18 Intel Nahalem processors. Moreover, the main originality of the proposed model here, is, for the first time, to exhibit very efficient simulations in the context of unstructured finite element meshes (regular grids in FFT context are considered in [153, 152]). Indeed, such a strategy will enable to consider large deformation of the calculation domain, paving the way to more complex mechanisms such as dynamic recrystallization. Our meshing/remeshing strategy, conserving a description of the bulk of the grains, will make it possible to investigate mechanisms involved in the grain boundary network, but also, in the grains substructures.

The implementation of this parallel scheme corresponds to the first perspective fulfilled for the general TRM approach. Other perspectives concern: the development of a DRX and PDRX TRM model and the possibility to perform simulation taking into account heterogeneous or anisotropic grain boundary properties, these questions will be discussed in the following chapters.

Résumé en Français du Chapitre 4

Ce chapitre a été dédié à l'introduction de nouveaux développements pour la méthode TRM introduite au chapitre 3, avec comme objectif la parallélisation de ses algorithmes. En effet, même si la méthode TRM a obtenu de meilleur temps de calcul que la méthode LS-EF, il est nécessaire qu'elle soit capable d'effectuer des simulations sur des stations de calcul à grand nombre de processeurs (clusters), afin d'être compétitive comparativement aux autres méthodes détaillées au chapitre 1 (Monte-Carlo, Cellular-Automata, LS, Multi Phase-Field), incorporant naturellement cette possibilité.

Les nouveaux développements pour la méthode TRM incluent plusieurs sous-algorithmes, chacun dédié à résoudre un problème spécifique. Ainsi la liste des nouveaux développements est comme suit: i. le développement d'une interface avec les outils de partitionnement des graphes de la librairie open-source Metis pour l'obtention d'un premier partitionnement du domaine; ii. un algorithme d'identification des entités divisés par les bords des partitions; iii. un algorithme de rééquilibrage et repartitionnement des charges; iv. un nouveau protocole de remaillage en parallèle et v. l'adaptation du modèle de mouvement Lagrangien des noeuds.

La nouvelle méthode parallélisée a été mise à l'épreuve sur des simulations de croissance de grains comportant un nombre élevé de grains initiaux. Deux types de benchmark ont été testés, le premier ayant une "charge variable" par processeur (CVPP), où un domaine de taille fixe est simulé sur un nombre variable de processeurs, et le deuxième avec une "charge constante" par processeur (CCPP), où la taille de domaine est choisie comme directement proportionnelle au nombre de processeurs utilisés. Le nombre maximal de grains initiaux testé sur les travaux de ce chapitre a été de l'ordre de 5.5×10^5 (conservant environ 20% des grains à la fin de la simulation), avec un nombre maximal de processeurs utilisées de 140. La simulation avec un nombre maximal de grains, calculé sur le nombre maximal de processeurs (5.5×10^5 grains sur 140 processeurs), prend environ 39 minutes de temps de calcul. Le temps de calcul est divisé par 48.5 par rapport à la même simulation réalisée en séquentiel. Or, les résultats à CCPP montrent que le modèle améliore sa performance en parallèle lorsque le nombre de grains initiaux par partition (GIPP) augmente. Ainsi, les simulations avec 4000 GIPP sont plus performantes que celles avec 2000 GIPP.

Finalement, des observations sur les courbes d'évolution de tests à CCPP ont permis de conclure que les variations sur les résultats statistiques des modèles à croissance de grains à champ complet (Full-Field) liés à la taille du Volume Élémentaire Représentative (VER) sont réduits à moins de 3% pour les simulations comportant plus de 50000 grains initiaux (ou conservant au moins 10000 grains pendant la durée de la simulation).

La méthode TRM parallélisée est donc capable de modéliser la croissance de grains en champ complet à des échelles uniquement atteignables à ce jour par des méthodes utilisant une discrétisation du domaine basée sur des grilles régulières et une résolution de type FFT (transformé de Fourier). Ceci confirme la pertinence de la nouvelle méthode et justifie son développement. Sur les chapitres suivants, la méthode TRM sera étendue pour la simulation d'autres mécanismes propres aux évolutions microstructurales: recristallisation dynamique et post-dynamique et la prise en compte d'anisotropies sur la définition des propriétés des joints de grains.

Chapter 5

Modeling of dynamic and post-dynamic recrystallization with the TRM model

A new method for the simulation of evolving multi-domains problems has been introduced in chapter 3 and further developed in parallel in chapter 4. However, in chapters 2 and 3, the new TRM model has only been applied in the context of isotropic GG with no consideration for the effects of the SE due to dislocations. In this chapter, further developments and studies of the TRM model will be presented, mainly on the development of a model taking into account GBM by SE. Further developments for the nucleation of new grains will be presented, allowing to model DRX and PDRX phenomena. Here, the results for multiple test cases will be given in order to validate the accuracy of the model taking into account GG and SE. Finally, the computational performance will be evaluated for the DRX and PDRX mechanisms and compared to a classical FE framework using a LS formulation.

The contents of this chapter have been submitted for publication in [22].

5.1 Introduction

The modeling, at the mesoscopic scale, of GG and ReX in polycrystalline materials during thermal and mechanical treatments has been the focus of numerous studies in the last decades. Indeed, as aforementioned, mechanical and functional properties of metals are strongly related to their microstructures which are themselves inherited from thermal and mechanical processing.

When looking to the so-called FF methods, based on a full description of the microstructure topology and modeling of GBM at mesoscopic scale, and when large deformations have to be considered (common in metal forming context), LS and MPF approaches in context of unstructured FE mesh and FE remeshing strategies, remain the main powerful and generic approaches but with a strong limitation in terms of computational cost.

In this context vertex and front tracking approaches appear as interesting candidates. An explicit description of the interfaces is considered and GBM is imposed at each increment by computing the velocity of the nodes describing the interfaces. While having a deterministic resolution (solving of partial differential equation - PDE), this methodology is very efficient. However, the implementation of algorithms allowing topological events in this context is not straightforward and the fact that these methods do not describe the bulk of the grains could be limiting for some metallurgical mechanisms, such as the appearance of new grains (nucleation is in general taken into account exclusively at the vicinity of multiple junctions) or substructures inside the grains.

The previous chapters have been dedicated to the creation of an improved front-tracking method, solving these weaknesses: the TRM model. This new model maintains the interior of grains meshed, handling with relative ease the topological changes of the grain boundary network and allowing the treatment of in-grain operations. Additionally, the TRM model has proven to have a higher computational performance than classical FE-LS models for the same accuracy. The objective of the present chapter is then to extend and apply the TRM model to handle DRX and PDRX phenomena, namely, reproducing previous methodologies for the modeling of these mechanisms, published in a FE-LS context [7].

5.2 The TRM model: towards the modeling of complex phenomena

The TRM model has been presented in chapter 3, then adapted to a parallel computational environment in chapter 4. This model uses the logic behind front-tracking methods where the discretization of interfaces is the minimal topological information allowing to model 2D-GBM. The TRM model goes a step further by implementing also a discretization of the interior of the grains in the form of simplexes to allow the interaction of the grain boundaries with the bulk of the grains and preventing inconsistencies of the physical domain such as the overlapping of regions. The data structure of the TRM model is then built on top of a mesh with elements and nodes, enabling also the possibility to compute FE problems on it (see section 3.2.1 for more information regarding the data structure of the TRM model).

This data structure contains the pieces of information needed to describe discrete geometrical entities such as *Points*, *Lines* and *Surfaces*. The classification of these entities is helpful when computing geometric properties: the area of surfaces can be computed by adding the contribution of each element of the grain, while the curvature κ and normal $\vec{n}$ of interfaces can be obtained by approximating the interface with a high order mathematical form (higher than the linear discretization of the domain) such as a least square approximation or with piecewise polynomials such as natural parametric splines. We have opted to use the latter in order to obtain such geometrical quantities.

The TRM model allows *physical* mechanisms to be simulated. These physical mechanisms represent how the different geometries are supposed to evolve and interact based on their current state. The TRM model has been developed to move the different nodes of the mesh based on a user-defined velocity field $\vec{v}$ and a time step dt. Once a velocity is defined a new position for each node N_i on the mesh can be obtained using Eq. 3.6. Here, each node displacement can potentially produce an overlap[1] of some of the elements attached to the node. The TRM model hence ensures the local conformity of the mesh employing a "locally-iteratively movement-halving", that finds iteratively the approximated maximal displacement that a node can make in the direction of the velocity v_i before an overlap takes place. This procedure ensures at all times that both, the mesh and the microstructural domain are valid.

The TRM model can be extended to model GBM when SE and capillarity act simultaneously as driving pressures, for which the velocity $\vec{v}$ describing these mechanisms has to be derived. Similarly, ReX can be simulated through the ap-

[1] An overlap in a mesh is produced when an element is partially or completely superposed by another element hence disrupting the 1:1 mapping of the numerical domain to the physical domain, such a mesh can not be used in a Finite Element resolution (see Fig. 3.16).

pearance of new grain boundaries (those delimiting new grains during nucleation), for which a remeshing strategy can be established. The following sections are devoted to the implementation of i. a new velocity term $\vec{v}$ taking into account SE into the dynamics of grain boundaries, along with some necessary changes concerning topological operations that may occur during the modeling of this mechanism (section 5.3) and ii. a remeshing procedure to include new domains (grains) in the microstructure and the laws governing their apparition as a function of the TMT followed by the material (section 5.4), where the methodologies proposed in [7] regarding this aspect were used.

5.3 Grain boundary migration under capillarity and SE driving pressures

The simulation of microstructural evolutions are given by the addition of complex and different phenomena as GG [5, 167, 201, 168, 169], ReX [5, 154, 140, 146, 6, 7, 142] or Zener Pinning (ZP) [98, 10, 11, 12, 147]. In chapter 3, GBM with no influence of SE was used to compare the TRM model to other approaches (LS-FE [140, 167, 168]), the base model used to represent this phenomenon is commonly known as migration by curvature flow. The velocity $\vec{v}$ at every point on the interfaces can be approximated following Eq. 1.3. In an isotropic context as considered in this chapter, the terms M and γ are supposed as invariant in space.

Of course, when post-dynamic phenomena such as SRX or MDRX are considered, the SE will act as another driving pressure of the GBM. Note that the SE within a grain can be variant, as there could be regions on the grain that have accumulated more or fewer dislocations during the considered TMT. At the mesoscopic scale, the SE can be discussed following different hypotheses. Crystal plasticity calculations and EBSD experimental data can bring dislocation density field and so SE field with fine precision until intragranular heterogeneities. While this information is directly usable in pixel/voxel based stochastic approaches such as MS or CA methodologies, generally it is homogenized by considering constant value per grain in deterministic front-capturing (MPF, LS) and front-tracking approaches. If this choice seems quite natural for phenomena where SE gradients and nucleation of new grains are mainly focused on GB like for discontinuous DRX (DDRX), it could be a strong assumption for phenomena where the substructure evolution is important, like for continuous DRX (CDRX). This aspect was for example studied in [159] in the context of SRX with a FE-LS numerical framework. It was concluded that intragranular gradients on the SE could indeed have a big impact on the grain morphology and that simulations taking into account such variations were more in accordance with experimental observations than simulation using a constant value of SE, but with an important numerical cost as the FE mesh must be then adapted at the intragranular heterogeneities

scale. In the following, a constant homogenized energy per grain is assumed. Nonetheless, the approach presented in this chapter to model GG with a SE field can be used in the context of a heterogeneous intragranular energy field, this aspect will be investigated in a forthcoming publication.

Thus, here SE can act on the displacement of the interface by considering the difference of SE at both sides of the interface. We will adopt a slightly modified methodology to the one presented in [154] to quantify it:

$$\vec{v_e} = -M\delta_{(\dot{\epsilon})}[E]_{ij}\vec{n}, \tag{5.1}$$

where the term $[E]_{ij}$ defines the difference of SE E between the grains i and j ($E_i - E_j$), the term $\delta_{(\dot{\epsilon})}$ is a mobility coupling factor whose nature is explained in [7] appendix c[2] and where the direction of the unit normal $\vec{n}$ sets, for a given node of the interface, the order of the indices as: first the index i and then j. Note that this definition holds even if the direction of $\vec{n}$ is ambiguous (in the case of a flat interface with no convex side) as the direction of the velocity $\vec{v_e}$ will be then pointed, in all cases, from the lower to the higher value of SE no matter what the direction of $\vec{n}$ is. Moreover, the value of SE can be computed using the equation:

$$E = \frac{1}{2}\mu b^2 \rho, \tag{5.2}$$

where b corresponds to the norm of the Burgers vector and μ corresponds to the elastic shear modulus of the material.

Finally, the contribution of driving pressures due to SE and capillarity can be accounted by linearly adding the two velocities as in [154, 6]:

$$\vec{v} = -M(\delta_{(\dot{\epsilon})}[E]_{ij}\vec{n} + \gamma\kappa\vec{n}), \tag{5.3}$$

where $\vec{v}$ denotes the final velocity of the interface during GBM when SE effects are included.

5.3.1 Velocity at multiple junctions

Equation 1.3 can only be used in a one-boundary problem, as in a more general context, the presence of multiple junctions (the intersection points of more than 3 interfaces) makes it impossible to compute a curvature κ or a normal $\vec{n}$ at these points. As explained in chapter 3, we have used an alternative methodology to compute the velocity due to capillarity at multiple points: Model II of [14], where the product $\kappa\vec{n}$ is directly obtained from an approximation of the free energy equation of the whole system in a vertex context.

[2] A mobility coupling factor function of the effective strain rate $\dot{\epsilon}$ with $\delta_{(\dot{\epsilon})} = 1$ when $\dot{\epsilon} = 0$.

Similarly, Eq. 5.1 only holds in a one-boundary problem as neither the value of $[E]_{ij}$ nor the value of $\vec{n}$ can be obtained at these points. To solve this, a different approach has been developed to compute a "resultant" velocity due to store energy $\vec{v_e}$ at multiple junctions. This approach is illustrated in Fig. 5.1 where for the sake of clarity, the value of M has been held constant and equal to 1. Fig. 5.1.a shows a typical configuration where the boundaries of three grains converge to a single point, each grain i has its own SE E_i where $E_1 > E_3 > E_2$. The values of the velocity for each normal boundary have been computed with Eq. 5.1 and are shown as white arrows for each node in the boundary of Fig. 5.1.b, here the index on the normal $\vec{n_{ij}}$ term is only representative of their direction and serve to set the indices of each $[E]_{ij}$ terms, these expressions do not follow the Einstein notation summation laws, all summations will be represented by the conventional Σ operator.

If a portion of differential size dr centered at the multiple point is evaluated (see Fig. 5.1 c)) the boundaries between grains will appear as flat, here the difference on the SE can be seen as a distributed difference of potential $[E]_{ij}$ applied on the length of the grain boundary of size dr (analog to a given pressure acting as a resultant force on a given interface). A normal $\vec{n}'_{ij}$ can be obtained and used to compute a velocity of each boundary (Fig. 5.1.d) applied at its center. Note that the direction of $\vec{n}'_{ij}$ can be chosen ambiguously on this linear segment, however, as mentioned before, an eventual ambiguity on the direction of $\vec{n}$ do not represent an ambiguity on the term $-[E]_{ij} \cdot \vec{n_{ij}}$ as $[E]_{ij} \cdot \vec{n_{ij}} = [E]_{ji} \cdot \vec{n_{ji}}$ with $[E]_{ji} = -[E]_{ij}$ and $\vec{n_{ji}} = -\vec{n_{ij}}$. These velocities can be divided and applied at the ends of each boundary and finally added at the junction point (Figures 5.1.e and 5.1.f respectively) to obtain a valid velocity vector field at multiple junctions. The expression on Fig. 5.1.f can be extended to the case where the values of M are neither constant nor equal to 1:

$$\vec{v_e} = \frac{-\Sigma M \delta_{(\dot{e})} [E]_{ij} \vec{n}}{2}, \tag{5.4}$$

of course, this expression can be also used in cases of multiple junctions of any order where more than three interfaces meet. Eq. 5.4 will be used to compute the value of v_e at multiple junctions as an approximation to the yet unknown behavior of such configurations under the influence of SE in a transient state.

5.3.2 Topological changes: capillarity, stored energy

Multiple changes in the topology of the microstructure occur during GG and ReX. In general, the topological changes during GG are given by the disappearance of grains: on a shrinking grain, each of their boundaries evolves until they *collapse* to multiple junctions. Eventually, all boundaries collapse to a single multiple junction and the domain occupied by the grain disappears. This behavior was implemented on the original TRM model presented in chapter 3 by means of the application of the *selective node collapse* operator, where some restrictions were

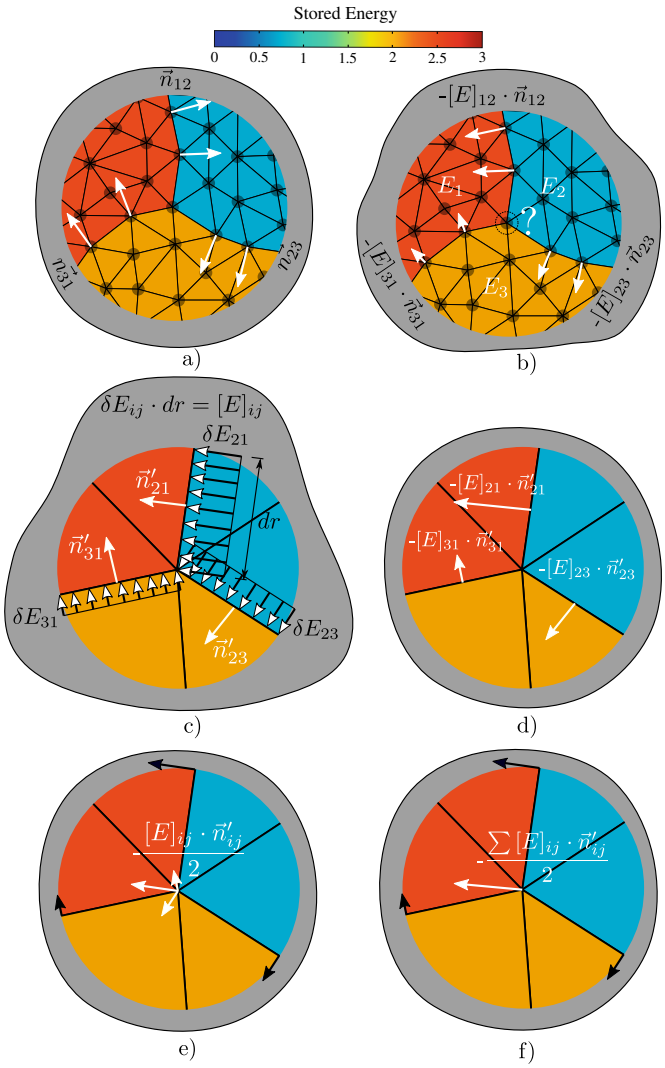


Figure 5.1: Graphical demonstration of the obtention of Eq. 5.4, a) typical triple junction configuration with values of SE homogenized on each grain and the normal vectors n computed at the nodes of the interfaces pointing to their convex side, b) computation of the term $-[E]_{ij} \cdot \vec{n_{ij}}$ for each node of the interface except for the node at the triple junction, c) definition of the same configuration as in a) but in a differential portion of radius dr, d) the resultant driving forces are applied at the center of the segments on the differential portion, e) and f) the driving forces are distributed at the ends of each segment and an expression can be formulated at the triple junction for its resultant driving force.

made regarding the order of collapsing.

In chapter 3 we opted to use this methodology to produce coherent topological changes on the microstructure, leading to a series of rules on the *selective node collapse* operator (see section 3.2.4). These rules gave to the P-Nodes a higher influence over other kind of nodes and prevented the collapsing of non-consecutive

nodes as illustrated in Figures 3.9 and 3.10.

The implementation of such node collapsing strategy allows a high control over the order on which the topological changes occur, unfortunately, this kind of reasoning can only be used on isotropic GG and can not be used when SE or spatial heterogeneities of the mobility/interface energy must be taken into account.

When considering SE, the kinetics of the GB are not only led by the movement of multiple points; flat surfaces can evolve with a given velocity and the velocity of simple boundaries may become much more important than the velocity of multiple junctions. Fig. 5.2 illustrates this behavior with six grains with a specific SE state. The circular grain in the middle grows due to its low SE compared to the SE of its surrounding grains. The circular grain is indeed surrounded by an initially squared grain that starts shrinking by the combined effects of capillarity at their external boundaries and the surface taken away by the circular growing grain. Fig. 5.2 (right), shows the moment when the boundary of the circular grain and the external boundaries of the initially square grain collide, unchaining a series of topological changes on the microstructure. These topological changes are illustrated in Fig. 5.3, where new multiple junctions (Points) appear, grain boundaries (Lines) are split, and grains (Surfaces) are divided.

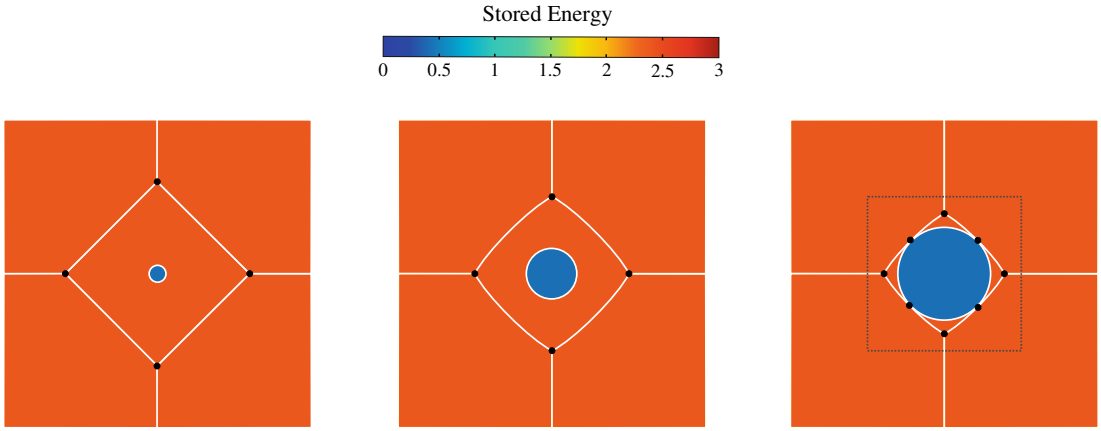


Figure 5.2: Six grains with a specific SE balance, the circular grain in the middle grows due to its low SE compared to the SE of its surrounding grains. The initially squared grain shrinks by the combined effects of capillarity at their external boundaries and the amount of surface taken away by the circular growing grain. Left: initial state, center: the circular grain grows, right: the boundary of the circular grain and the external boundaries of the initially square grain collide.

These several changes on the microstructure (contact of different grain boundaries in non-convex grains) can not be accomplished by the TRM model if the rules described above and illustrated by figures 3.9 and 3.10 are maintained.

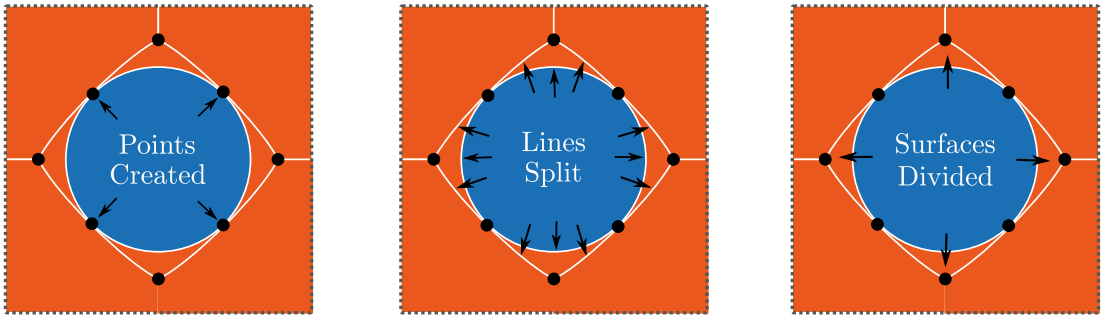


Figure 5.3: Details of the final event of Fig. 5.2, highlighting the changes on the microstructure. Here, Points describe multiple junctions, Lines grain boundaries and Surfaces grains.

This is why these two rules need to be overridden and a new condition implemented: If two non-consecutive nodes (nodes not connected by the microstructural wireframe) collapse, the classification of the remaining node is a P-Node. Additionally, the remaining node is moved to the barycenter of the initial nodes involved in the collapse and the surrounding geometrical entities (points, lines and surfaces) are checked and updated if necessary. Take for example the same configuration shown in Fig. 3.9 now in Fig. 5.4: here the collapsing of nodes N_i and N_j is possible, the remaining node N_i is placed in the middle of the edge $\overline{N_iN_j}$ and its classification is changed from L-Node (blue) to P-Node (red). Now their surroundings need to be checked for possible changes on the topology: all three Lines (grain boundaries in cyan) need to erase node N_k as a final/initial point and put in its place P-Node N_i, similarly, one of the lines has to add N_k as a node in their sequence of L-Nodes hence N_k changes also its classification to L-Node.

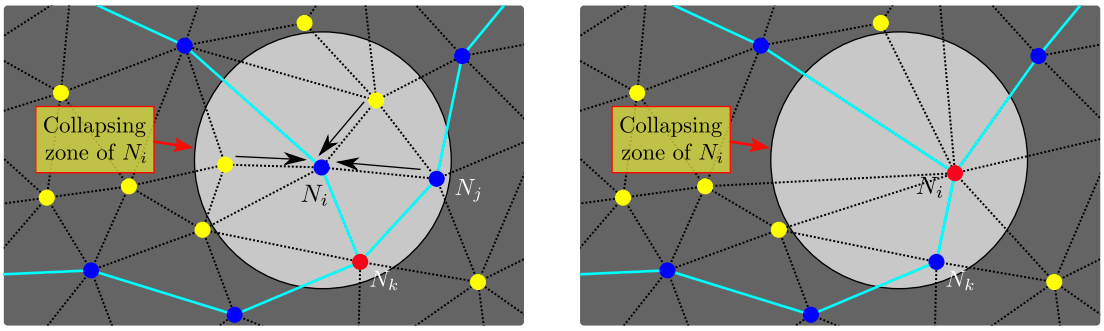


Figure 5.4: Node Collapsing of Fig. 3.9, when allowing the collapse between non consecutive nodes. S-Nodes are displayed in yellow, L-Node in blue and P-Nodes in red. The collapsing of nodes N_i and N_j produces a new P-Node (N_i) while the preexistent P-Node N_k needs to be reclassified as a L-Node. Left: initial state, right: state after collapse.

Similarly, the situation presented in Fig. 3.10 can be reproduced with the new

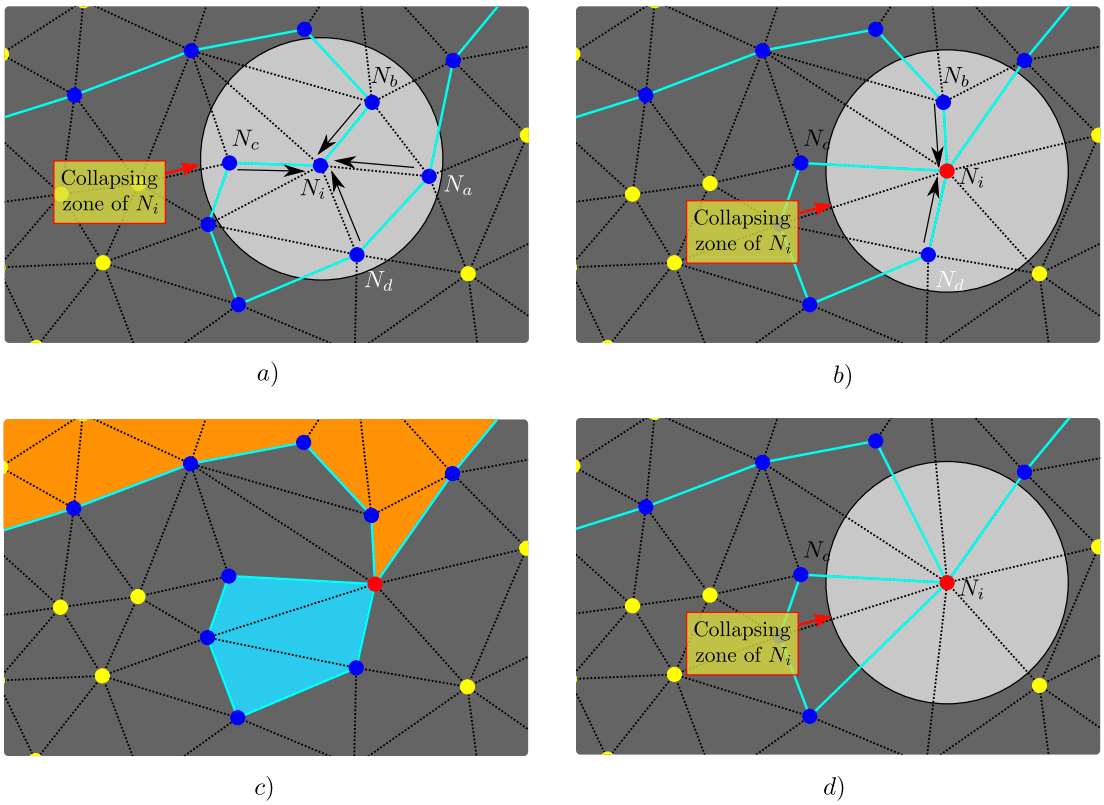


Figure 5.5: Node Collapsing of Fig. 3.10, when allowing the collapse between non consecutive nodes. S-Nodes are displayed in yellow, L-Node in blue and P-Nodes in red. a) Initial state; b) L-Node N_i performs the first collapse with N_a. This produces L-Node N_i to be moved to the center of the edge $\overline{N_iN_a}$ and to become a P-Node. Moreover, the collapsing zone of N_i changes its position and leave Node N_c out; c) The collapse also has produced a Surface to be divided in two (cyan and orange Surfaces); d) Additional collapses are performed between P-Node N_i and L-Nodes N_b and N_d, these collapses are performed in the conventional way without moving N_i as it is a P-Node now.

rules of collapsing: on the initial state (Fig. 5.5.a), the collapsing zone of N_i puts L-Nodes N_a, N_b, N_c and N_d inside. The first node to be collapsed is L-Node N_a. After this initial collapse (Fig. 5.5b)) the collapse produces L-Node N_i to be moved to the center of the edge $\overline{N_iN_a}$ and to become a P-Node. Moreover, the collapsing zone of N_i changes its position and leave L-Node N_c out hence it will not be collapsed. Additionally, a new topological change is identified (Fig. 5.5.c), here the collapse is also responsible for the fact that a Surface is divided in two new surfaces (cyan and orange Surfaces). Finally, the remaining collapses are performed between P-Node N_i and L-Nodes N_b and N_d (Fig. 5.5.d), these collapses are performed in the conventional way following the rules of collapsing between P-Nodes and L-Nodes presented in chapter 3.

The new node collapsing rules have been implemented in the TRM model and will be used from this point forward in the cases where SE is present. This node collapsing technique will be able to perform the majority of the topological changes in the microstructure. Furthermore, for the purpose of this chapter, the creation of boundaries by the decomposition of unstable multiple junctions (multiple junctions with more than 3 meeting boundaries) is handled via the procedure introduced in section 3.3.1 without any further implementation.

5.3.3 The TRM algorithm under the influence of capillarity and stored energy

Finally, the algorithm for a time step of the TRM model in the context of isotropic grain growth under the influence of SE and capillarity is presented in Algorithm 9, where the step "Perform Remeshing and Parallel Sequence" corresponds to the parallel implementation of the TRM model presented in chapter 4.

Algorithm 9 Isotropic Grain Growth TRM Algorithm for capillarity and SE

Perform Remeshing and Parallel Sequence
for all Points: P_i **do**
    while Number of Connections > 3 **do**
        split multiple point P_i.
for all Lines : L_i **do**
    Compute the natural spline approximation of L_i.
for all L-Nodes : LN_i **do**
    Compute curvature and normal ($\kappa\vec{n}$) over LN_i then compute $\vec{v_c}$ for LN_i (Eq. 1.3).
    Compute the $\vec{v_e}$ for LN_i (Eq 5.1)
for all P-Nodes : PN_i **do**
    Compute the product $\kappa\vec{n}$ over PN_i using model II of [14] then compute $\vec{v_c}$ for PN_i (Eq. 1.3).
    Compute $\vec{v_e}$ for PN_i (Eq. 5.4)
Delete Temporal Nodes
for all L-Nodes and P-Nodes: LPN_i **do**
    Compute final velocity $\vec{v}$ of Node LPN_i (Eq. 5.3)
Iterative movement with flipping check in parallel

5.4 Recrystallization

In order to model ReX with the TRM model, two additional components are necessary: the first is a procedure allowing to change the topology of the microstructure and to introduce new grains (i.e. nuclei); the second component

is a model of the apparition of nuclei which depends on thermomechanical conditions. Here DDRX context is considered. Of course, PDRX and subsequent GG phenomena can also be investigated by considering microstructure evolutions when the deformation is completed. The combination of these two mechanisms can describe multiple TMTs that are used today in the material forming industry.

With the purpose of simplicity, in this chapter we will use the same methodology presented in [7] for the laws governing the introduction of new nuclei during the modeling of hot deformation:

In [7], the evolution of the dislocation density is accounted by a Yoshie-Laasraoui-Jonas Law [211] as follows:

$$\frac{\partial \rho}{\partial \epsilon^p_{eff}} = K_1 - K_2 \rho, \tag{5.5}$$

which can be evaluated in a discretized time space with an Euler explicit formulation for the next increment step as :

$$\rho^{(t+\Delta t)} = K_1 \Delta\epsilon + (1 - K_2 \Delta\epsilon)\rho^{(t)}, \tag{5.6}$$

where $\rho^{(t)}$ is the value of the dislocation density at time t and where the value of $\Delta\epsilon$ can be computed as $\dot{\epsilon^p_{eff}} \cdot \Delta t$ with Δt the time step.

As explained in [7], when a grain boundary migrates, the swept area is assumed almost free of dislocations. This aspect is modeled by attributing to these areas a value of dislocation density equal to ρ_0, then, for the grains with part of their domain presenting ρ_0, their dislocation density is homogenized within the grain (as intragranular gradients on the SE are not taken into account either in [7] nor in the present work), the final value of ρ for the growing grains is computed as:

$$\rho^{(t+\Delta t)} = \frac{\rho^t S^{(t)} + \Delta S \rho_0}{S^{(t+\Delta t)}}, \tag{5.7}$$

where S^t and ΔS denote the surface at time t and the change of surface of a given grain.

Additionally to Eq. 5.7, in PDRX the annihilation of dislocations by recovery must be taken into account. This is done thanks to the following evolution law:

$$\frac{d\rho}{dt} = -K_s \rho, \tag{5.8}$$

where K_s is a temperature-dependent parameter representing the static recovery term. This recovery law is only taken into account in PDRX as in DRX, Eq. 5.7 already takes into account the annihilation phenomenon.

5.4.1 Nucleation laws

The procedure consists in introducing volume (surface in 2D) of nuclei at a rate of $\dot{S}$, once the local value of dislocation density has reached a critical value: ρ_c. In [6, 7, 212] this value was obtained by iterating until convergence the following equation:

$$\rho_c^{(i+1)^*} = \left[\frac{-b\gamma\dot{\epsilon}\frac{K_2}{M\delta_{(\dot{\epsilon})}\tau^2}}{ln(1-\frac{K_2}{K_1}\rho_c^i)} \right]^{\frac{1}{2}} \text{ with } \rho_c^i = \rho_c^{i-1} + c\cdot(\rho_c^{(i)^*} - \rho_c^{i-1}), \tag{5.9}$$

where i represents the iteration number, c is a convergence factor ($c < 1$ chosen for the works of this chapter as $c = 0.1$), K_1 and K_2 represents the strain hardening and the material recovery terms in the Yoshie–Laasraoui–Jonas equation discussed in [211], the term $b = 1$ in 2D and $b = 2$ in 3D, τ is the dislocation line energy and $\dot{\epsilon}$ is the effective deformation rate used during the deformation of the material.

When solving this equation, two special cases may produce an erroneous computation: the first is given when $K_1/K_2 * \rho_c > 1$ for which the logarithm is undefined, the solution to this is to limit the value of $\rho_c < K_2/K_1$ whenever this situation occurs. The second is when $\dot{\epsilon} = 0$ which corresponds to the intervals where PDRX is considered. Two solutions may be considered for this situation: the first is to block the nucleation when it is not necessary (metadynamic evolution for example), and the second to supply value of $\dot{\epsilon} > 0$ to Eq. 5.9. Here, we have chosen the latter, for which an apparent effective strain rate $\dot{\epsilon_s}$ is used instead $\dot{\epsilon}$ in PDRX:

$$\dot{\epsilon_s} = \frac{\int_0^t \dot{\epsilon}^2 \, dt}{\int_0^t \dot{\epsilon} \, dt}, \tag{5.10}$$

which accounts for the instant mean value of the real effective strain rate.

Once a value of ρ_c is computed, the surface per unit of time $\dot{S}$ of nuclei to be inserted can be computed with the following equation corresponding to a variant of the proportional nucleation model [70]:

$$\dot{S} = K_g P_c, \tag{5.11}$$

where the term K_g is a probability constant depending on the processing conditions and P_c is the total perimeter of the grains whose dislocation density is greater than ρ_c.

Another constraint is given by the minimal radius r^* of nucleation (the radius at which the nuclei should be inserted in the domain so the capillarity forces would not make it disappear) which can be computed thanks to the following equation [213]:

$$r^* = \omega \frac{\gamma}{(\rho_c - \rho_0)\tau}, \tag{5.12}$$

where $\omega > 1$ is a safety factor ensuring the growth of the nucleus at the moment of its apparition. The term ω accounts for the non-spherical shape of a grain inserted in a discretized domain such as in the TRM model. In section 5.5.1 a value for this factor will be obtained based on numerical tests.

5.4.2 Nucleation approach for the TRM model

Having defined the tools needed to obtain the kinetics of the grain boundaries, where the pressure behind such kinetics can be of different nature: capillarity, SE or both. However, in order to model ReX it is necessary to have a way to introduce new grains into the domain of the TRM model. Nucleation, similarly to boundary migration, is one of the ways of the microstructure to relax the high gradients of the SE appearing during or after a TMT. Nucleation has been addressed by several approaches for each methodology able to simulate such behavior: LS-FE methods, relay on the definition of circular LS fields (different from the already defined LS fields occupying the same spatial domain) to form nuclei [8, 7], CA and MC methods change the crystallographic orientation and SE value of some cells [77, 214, 90, 215] in order to nucleate while vertex models form new grains by redefining new vertex and interfaces in the shape of triangles around the preexistent vertex points [18].

In the present work, a remeshing-reidentification procedure will be performed around a central node N_i in order to introduce nuclei. A circular region with center N_i will be drawn and all edges crossed by this circle will be similarly split at the intersection as in [161] and in chapter 2 by successively applying an edge splitting operation, regardless of the classification of the nodes defining the edge (P-Node/L-Node/S-Node) (see Fig. 5.6). The classification of the new nodes being placed by the splitting algorithm is as L-Nodes unless the split edge represents a grain boundary, in which case the inserted node will be classified as a P-Node (see Figures 5.7.b middle and 5.7.c middle). Once all edges are split, a Surface Identification algorithm will be performed over node N_i (see section 3.2.2), all identified elements and nodes will be inserted into a new empty Surface defining the nucleus, and extracted from their previous Surfaces (grains), new Lines (grain boundaries) will be built with their respective Points (multiple junctions) if any were formed by the nucleation process and all remaining lines and points inside the new surface will be destroyed (remaining lines and points can appear if the nucleation took place near a grain boundary) see Figures 5.7.a, 5.7.b and 5.7.c left. However, in a parallel context an additional constraint has been added: shared nodes can not be involved in the nucleation process, neither as a central node nor one of the nodes of a split edge. This constraint was added because of the lack of information (position of the edges to cut) around shared nodes and the performance of the nucleation process (as a great amount of information would be

necessary to be transferred to other processors). This constraint should not have a great impact on the general behavior of the model as the domain of each processor (and their shared nodes) is changed by the *Unidirectional Element Sending* algorithm presented in chapter 4, every time step, hence constantly unblocking the restriction to nucleate over the same region.

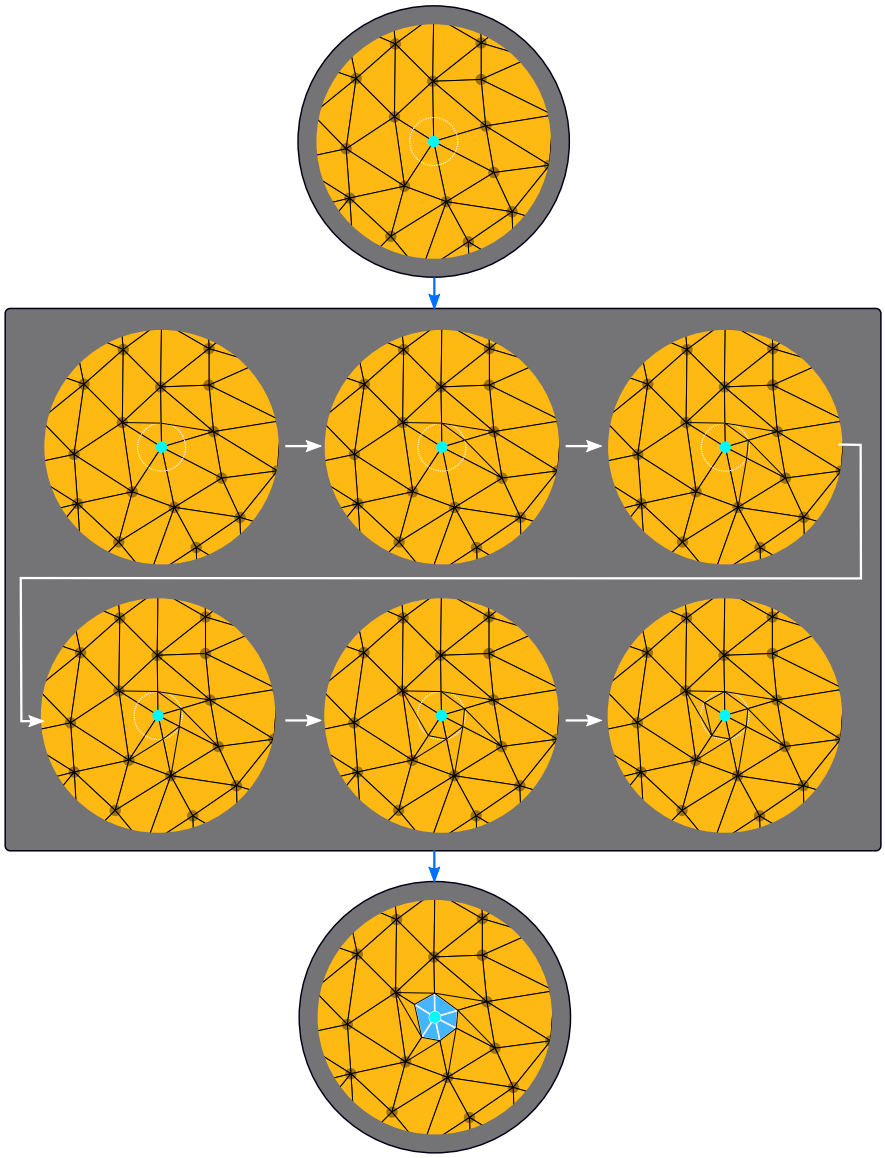


Figure 5.6: Remeshing steps for the nucleation process of the TRM model. Top: initial state with a selected node (cyan) and a circle drawn over the mesh, middle: successive edge splitting steps to form the interfaces of the nucleus, bottom: the elements inside the nucleus are identified and extracted from its previous Surface container.

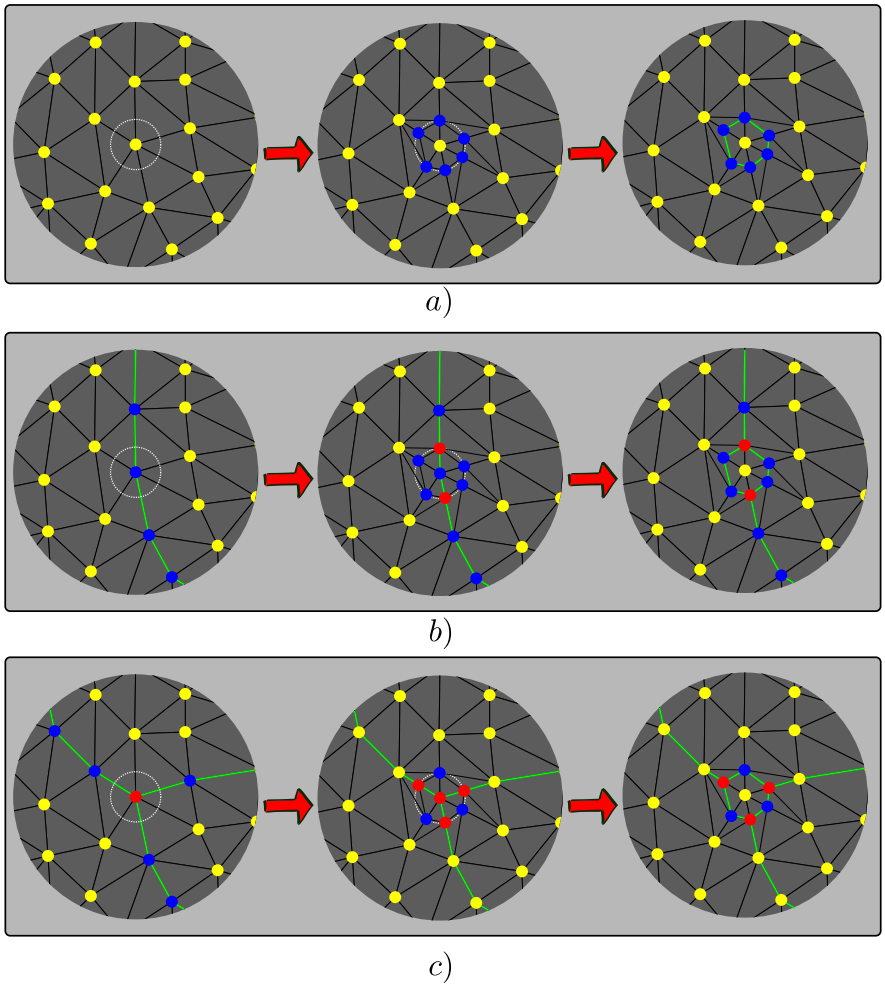


Figure 5.7: Examples of the formation of nucleus over different types of nodes, a) S-Node at the center and no crossed lines, b) L-Node at the center and one crossed line, two P-Nodes are created, the initial line is divided and two new lines are created, c) P-Node at the center, 3 P-Nodes are created, the P-Node at the center is detached from all its lines and converted to S-Node, 3 new lines are created.

5.5 Numerical tests

In this section different academic tests will be performed to evaluate the performance of the TRM model when simulating GBM under the influence of capillarity and SE. For these academic tests, dimensionless simulations will be considered. Moreover, the results of simulations using the DRX and PDRX frameworks described in section 5.4 will be given, these simulations will use the nucleation approach presented in section 5.4.2 specially developed for the TRM model. The different physical parameters will be taken as representative of the 304L stainless steel. Comparisons with LS-FE predictions will be discussed.

5.5.1 Circular Grain: competition between capillarity and stored energy

In this test case, it will be evaluated the accuracy of the model when the geometric configuration leads to a competition between the driving forces given by the capillarity and the SE. Here we will adopt a value of boundary energy and mobility equal to $\gamma = 1$ and $M = 1$ respectively. A circular domain with a value of SE $E = \alpha$ is immersed in a squared domain with an attributed value of SE $E = \beta$ (see Figures 5.8.a and 5.8.b left) where $\beta > \alpha$. The difference on the SE $[E] = \beta - \alpha$ at the boundary will try to make the circle expand at a rate $v_e = M[E] = [E]$ while the capillarity effect will try to make it shrink at a rate $v_c = M\kappa = \kappa$ where κ is the local curvature. The analytical model for this configuration can be put in terms of a non-linear ordinary differential equation in terms of the radius r of the circle as follows:

$$\frac{dr}{dt} = -\frac{1}{r} + [E], \tag{5.13}$$

or in terms of the surface S of the circle:

$$\frac{dS}{dt} = 2(-\pi + \sqrt{\pi S}[E]), \tag{5.14}$$

We have used an Euler explicit approach to solve this equation and the results are used to compare the response of the TRM model for different values of $[E]$ for two cases: the first is given for an initial radius of $r_0 = 0.3$ and the second for $r_0 = 0.025$ (see Figures 5.8.a and 5.8.b left). The initial mesh for each one of the two cases is given in Figures 5.8.a and 5.8.b right respectively. Note how in the first case, the initial circle boundary is discretized by a number of nodes sufficiently capable of capturing precisely the value of its curvature, hence it will serve to evaluate the accuracy on the kinetics of a typically curved boundary, while in the second case, the circle boundary is only defined by a few nodes allowing to evaluate the behavior of a nucleus when it is inserted on the domain.

Results for this first case are given in Fig. 5.9 along with the solution of Eq. 5.14 for different values of $[E]$, Fig. 5.9.left illustrates how the references

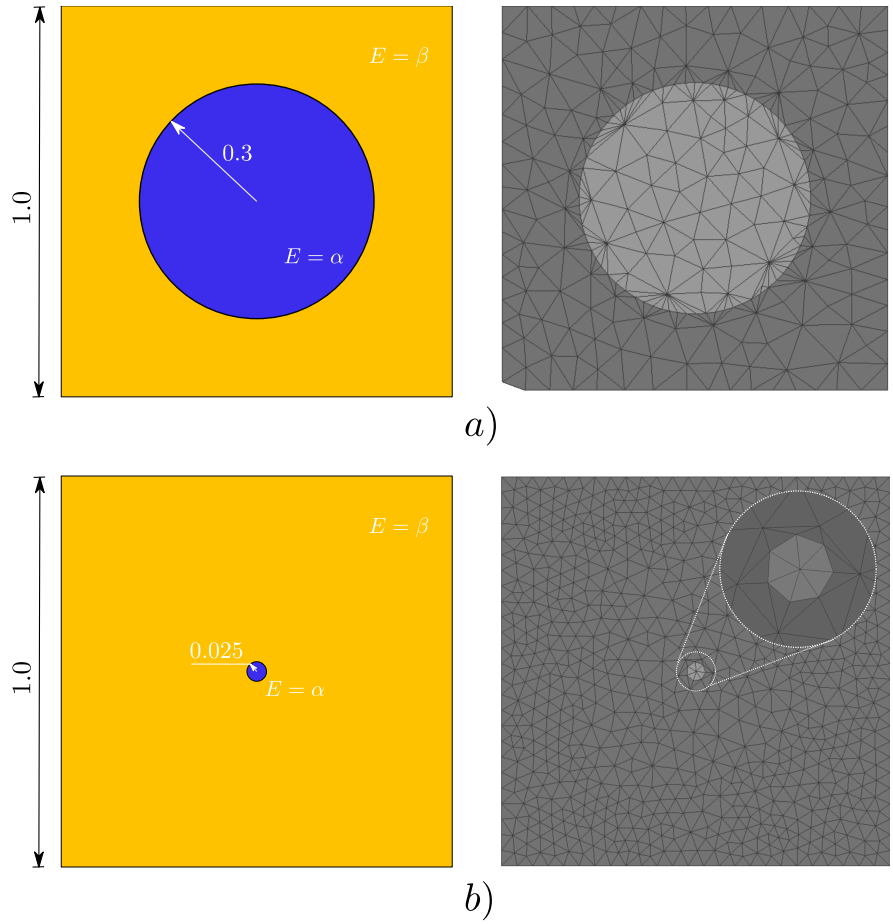


Figure 5.8: Circle Test, left: initial state and right: initial mesh a) $r = 0.3$ radius (Surface=0.287) b) $r = 0.025$ radius (Surface=0.01963)

curves are superposed to the different simulated curves with a very low error (around 2 % max see Fig. 5.9.right). Furthermore, the analytic metastable case (given for $[E] = 10/3$) shows a very good behavior losing only 1.2% of its surface at $t = 0.09$.

Similarly, The results for the second circle case are given in Fig. 5.10. Here it is appreciated how for the cases where the capillarity is the higher driven force ($[E] = 0, 10, 20, 30$), the circle disappears at the good rate. An interesting discussion concerns the case with $[E] = 40$ which corresponds analytically to the metastable configuration. In TRM simulation, the grain disappears. This behavior is due to the low number of nodes at the interface, producing an overestimation on the computed value of its curvature, making it shrink from the very first increment. A value of $[E] \approx 48,5$ was necessary on the simulated side to maintain a metastable position (an increase of 21.2% accordingly to its analytical value). Moreover, for this value, the error on the prediction of the evolution of the surface was also the highest, going up to 92% after $t = 0.003$. Of course, this error is given as the simulated circle maintains its surface, while the analytical solution

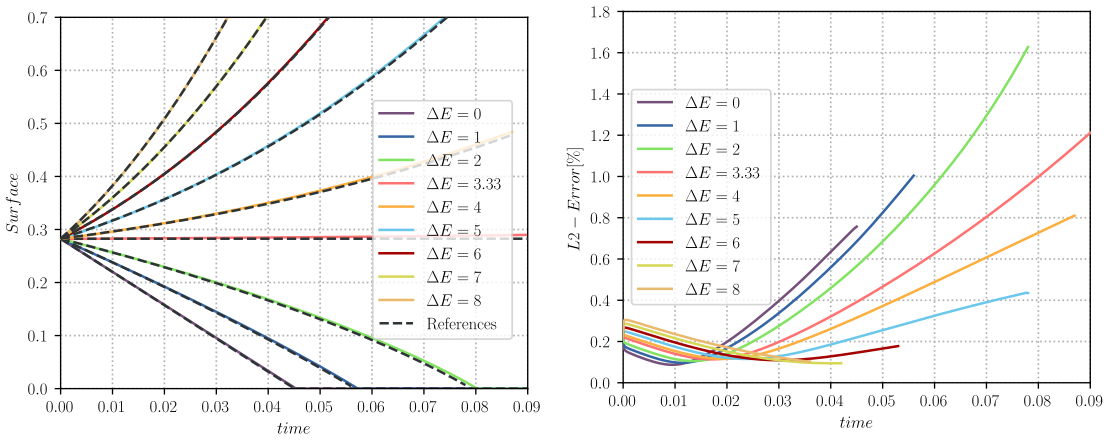


Figure 5.9: Evolution of the surface (left) and L2 error (right) for the circle test case for an initial circle radius $r = 0.3$ (Surface=0.287) a mesh size $h = 0.006$ and a delta time $dt = 3{\cdot}10^{-5}$, the analytical results (References) are shown superposed to the simulated curves in black dashed lines. The expected metastable curve is given for a $[E] = 10/3$ (Red curve).

shows a continuous increase. The curves corresponding to $[E] = 50, 60, 70, 80$ (for which the higher driving force is the SE) show a decreasing error when the value of $[E]$ increases. This result can be used on the determination of factor ω used in Eq. 5.12 where the authors have estimated that a value of $\omega = 1.5$ (which counteracts for an increase of 50% over the analytical value of $[E]$ for a metastable state, see the curve $[E] = 60$ in Fig. 5.10.) is sufficient in order to give the inserted nucleus a growing state and prevent its early disappearance.

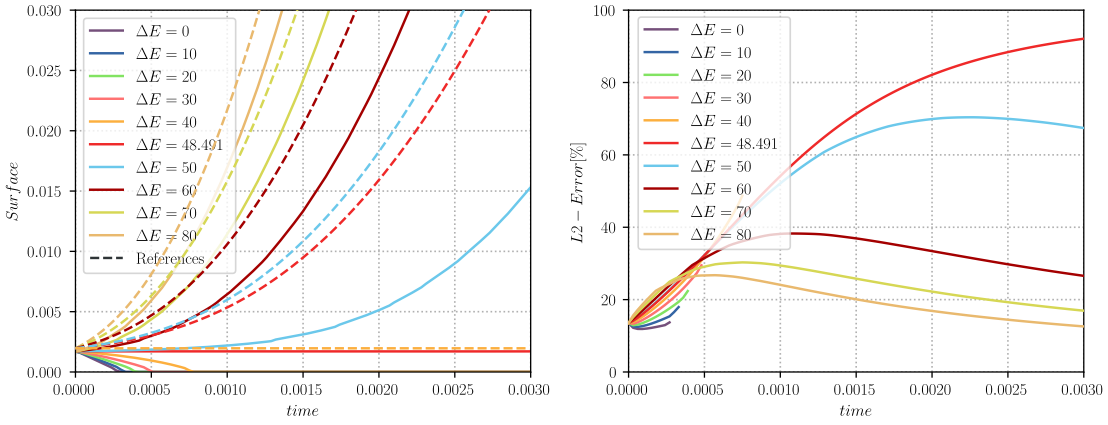


Figure 5.10: Evolution of the surface (left) and L2 error (right) for the circle test case with an initial circle radius of $r = 0.025$ (Surface=0.01963) a mesh size $h = 0.025$ and a delta time $dt = 1 \cdot 10^{-5}$, the analytical results (References) are shown as dashed lines of the same color of their corresponding simulated evolution. The expected metastable curve is given for a $[E] = 40$ (orange curve), but metastability was found for $[E] = 48.491$ (Red curve)

5.5.2 Triple junction: The capillarity effect on the quasi-stable shape of multiple junctions

In [139, 216] analytic solutions for the movement of multiple junctions in a quasi steady-state under the influence of SE were presented. In [139] the so called "Vanishing Surface Tension" (VST) test was introduced to demonstrate the non-uniqueness of the solution presented in [216] hereafter called the "Sharp" solution, this test (the VST test) takes the form of the limit problem given by:

$$\vec{v} \cdot \vec{n} = -M([E]_{ij} + \epsilon\gamma\kappa), \text{ with } \epsilon \to 0, \tag{5.15}$$

which has subjected to several 2D test cases and a perturbation analysis to demonstrate that the VST solution corresponds to one of the solutions when $\epsilon = 0$ and to the unique solution otherwise.

These solutions were later studied in [5, 154] using a LS-FE model to obtain the same behavior both in 2D and 3D. Here we have reproduced with the TRM model two tests that show the same behavior as in [139, 5, 154] for the 2D solutions. For all tests, the "Sharp" solution was obtained when capillarity effects were taken into account (with $\epsilon = 1$) and the VST solution when no capillarity was introduced in the system (hence with a value of $\epsilon = 0$). Furthermore, we have developed analytic equations for the evolution of the growing surface in our specific case (see Fig. 5.11), these analytic evolutions are valid up to the point of contact of the multiple junction with the lower edge of the equilateral triangle (the limits of our domain) and allow us to make a more quantitative comparison in terms of error.

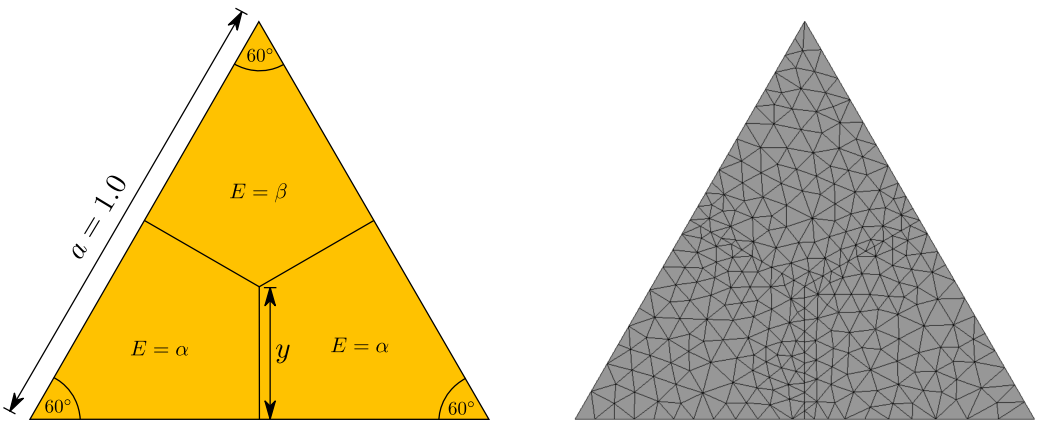


Figure 5.11: Initial state for the triple junction test, three phases immersed in a domain in the shape of an equilateral triangle, this shape is intended to maintain an orthogonal position of the boundaries with respect of its limits while the configuration evolves. a) Initial configuration and b) initial mesh.

For this test case, the initial conditions are those presented in Fig. 5.11.left,

three phases immersed in a domain in the shape of an equilateral triangle, this shape is intended to maintain an orthogonal position of the boundaries with respect of its limits while the configuration evolves. Two of the phases (the two in the lower part of the domain) will have a constant value of SE of α and the third phase a value of $\beta < \alpha$, this configuration will produce a global movement of the triple junction downwards at a constant and normal velocity of the flat interfaces equals to $\alpha - \beta$. Eventually, the triple point will reach the bottom part of the domain making it to split and evolve towards a lower energy state; even though this portion of the simulation is showed in some of the results it is not relevant to our study, hence we will give quantitative results up to the point of splitting. The initial mesh for every test performed is shown in Fig. 5.11.right corresponding to a mesh size parameter of $h_{trm} = 0.006$. Furthermore, values for the boundary energy and mobility have been set to $\gamma = 1$ and $M = 1$ respectively.

The analytic solution for the evolution of the surface of the upper phase (the growing phase) for the Sharp solution is given by:

$$S_{Cap} = \left(\frac{2a}{\sqrt{3}} - y\right)^2 \frac{\sqrt{3}}{4}, \tag{5.16}$$

where a is the length of one of the sides of the equilateral triangle (here $a = 1$) and y is the vertical position of the triple junction measured from the base of the triangle and given by the following expression:

$$y = \frac{a}{2\sqrt{3}} - |\vec{v} \cdot \vec{n}| \frac{2t}{\sqrt{3}}, \tag{5.17}$$

where t is the time and the expression $|\vec{v} \cdot \vec{n}|$ is the instant normal velocity of the flat phase boundaries, i.e. $\alpha - \beta$.

Similarly, the analytic response of the VST solution in terms of surface for the growing phase is given by

$$S_{NoCap} = S_{Cap} + \left(\frac{\pi}{6} - \frac{1}{\sqrt{3}}\right) (|\vec{v} \cdot \vec{n}|t)^2. \tag{5.18}$$

Two test were performed: one with $\beta = 2$ and $\alpha = 4$, i.e. $|\vec{v} \cdot \vec{n}| = [E] = 2$ and one with $\beta = 10$ and $\alpha = 20$, i.e. $|\vec{v} \cdot \vec{n}| = [E] = 10$. The two tests were performed with a time step $\Delta t = 1 \cdot 10^{-5}$. Results for the evolution of the mesh and the surface are given in Figures 5.12 and 5.13 for the first and the second case respectively. It is clear that the accuracy on the scalability of the solution is very good as Figures 5.12.a, b and c are almost equal to the ones of Figures 5.13.a, b and c respectively which were obtained for a velocity 5 times higher. Note that the only different frame is given for Figures 5.12.d right and 5.13.d right as here the capillarity effects over the limits of the domain are not negligible and in Fig. 5.12.d right the configuration have had 5 times more time to evolve to its

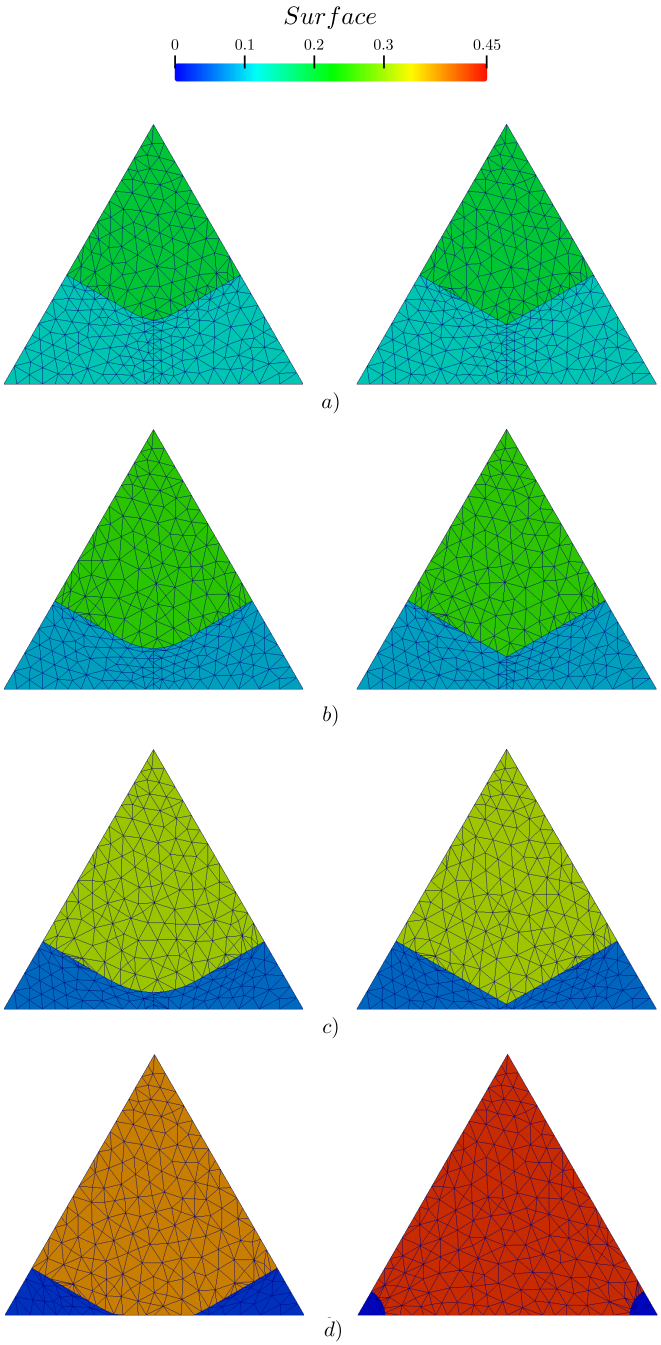


Figure 5.12: States for the triple junction test case with a value of $[E] = 2$, left: with $\epsilon = 0$ and right: with $\epsilon = 1$ at a) $t = 0.02$ b) $t = 0.04$, c) $t = 0.06$, d) $t = 0.08$.

given state.

The evolution of the surface of the growing phase and its error with respect to equations 5.16 and 5.18 is given in Fig. 5.14, where the L2-Error for both cases was lower than 0.8%.

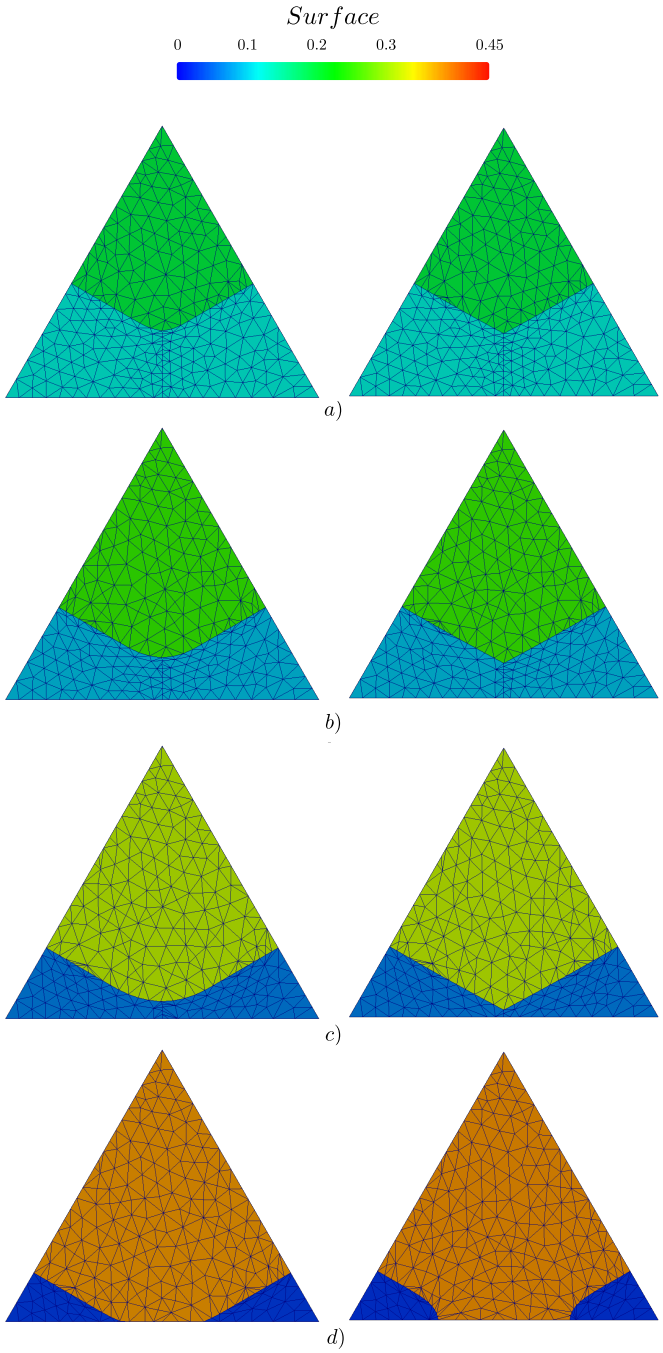


Figure 5.13: States for the triple junction test case with a value of $[E] = 10$, left: with $\epsilon = 0$ and right: with $\epsilon = 1$ at the instant a) $t = 0.004$ b) $t = 0.008$, c) $t = 0.012$, d) $t = 0.016$.

5.5.3 DRX/PDRX case

Here a simulation with a few initial grains will be performed using the recrystallization method mentioned in section 5.4: the initial tessellation will be realized thanks to a Laguerre-Voronoi cells generation procedure [55, 56, 57] over a rect-

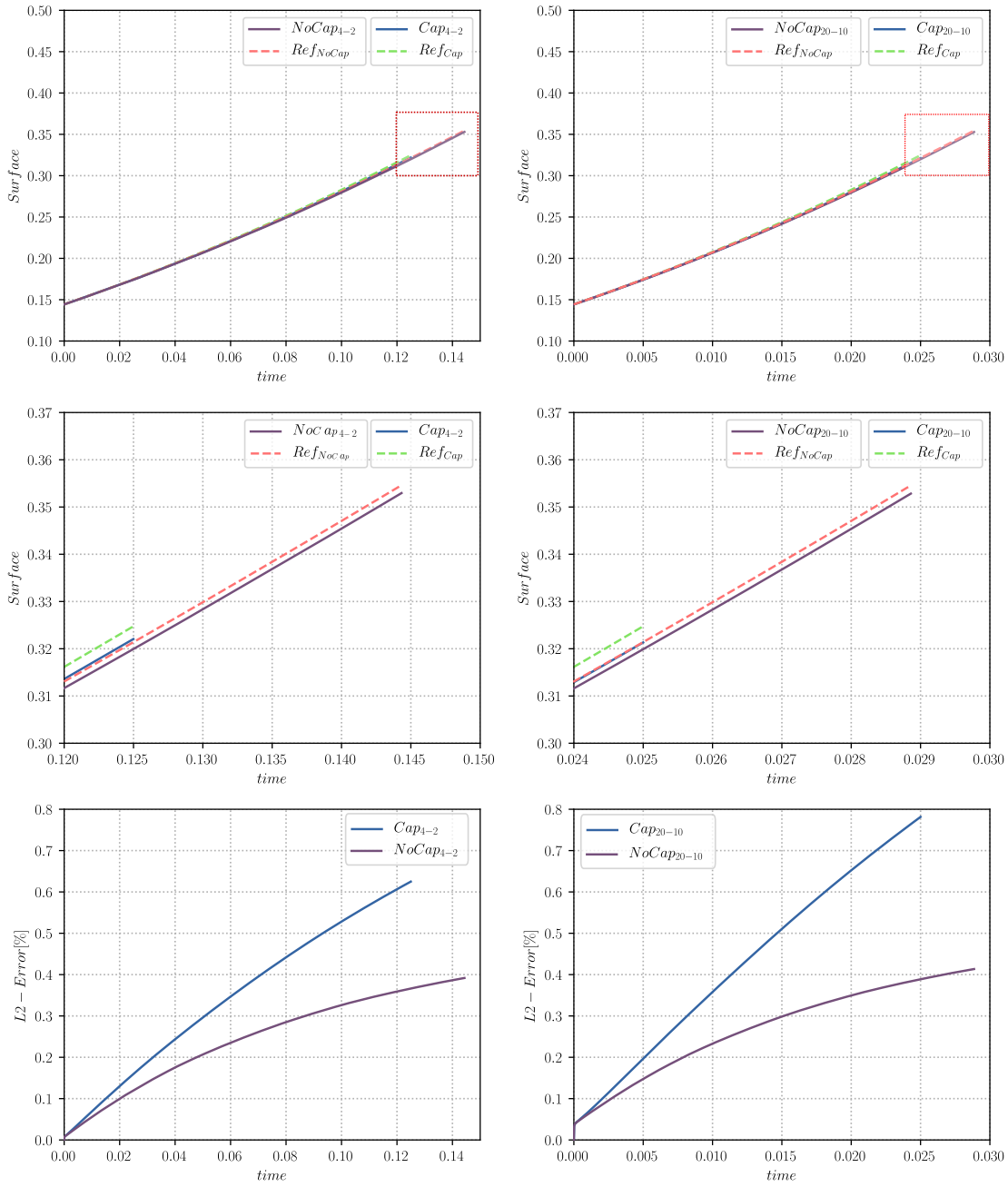


Figure 5.14: Evolution of the surface of the growing phase of the triple junction test case, from top to bottom: (top) Evolution of the surface, (center) Zoom in the red zone, (bottom) L2-Error over the evolution of the surface. Left: results for the test with $[E] = 2$ and right: with $[E] = 10$

angular domain of initial dimensions 0.65×0.328 mm (see Fig. 5.15) and the values for M, γ, τ and k_s are chosen as representative of a 304L stainless steel at 1100 $°C$ (with $M = M_0 * e^{-Q/RT}$ where M_0 is a constant $M_0 = 1.56 \cdot 10^{11}$ mm^4/Js, Q is the thermal activation energy $Q = 2.8 \cdot 10^5$ J/mol, R is the ideal

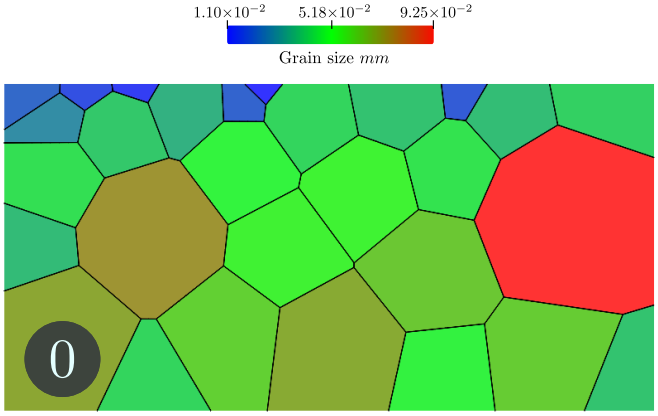


Figure 5.15: Initial State for the DRX/PDRX test case.

gas constant, T is the absolute temperature $T = 1353\ K$, $\gamma = 6 \cdot 10^{-7}\ J/mm^2$, $\tau = 1.28331 \cdot 10^{-12}\ J/mm$ and $k_s = 0.0031\ s^{-1}$ [6, 7]). Additionally, the parameters K_1, K_2, K_g and δ are taken as dependent of the absolute value of the component xx of the strain rate tensor $\dot{\varepsilon}$ ($|\dot{\varepsilon}_{xx}|$) which is defined as corresponding to a plane deformation case. These parameters will be obtained using a linear interpolation of the values presented in Tab. 5.1.

Table 5.1: Parameter data table for the DRX PDRX test case, when in range $|\dot{\varepsilon}_{xx}| = [0.01, 0.1]\ s^{-1}$ the values are interpolated. If $|\dot{\varepsilon}_{xx}| > 0.1$ the value for the corresponding parameter will the same as for $|\dot{\varepsilon}_{xx}| = 0.1\ s^{-1}$, the same strategy applies when $|\dot{\varepsilon}_{xx}| < 0.01$.

$\lvert\dot{\varepsilon}_{xx}\rvert\ s^{-1}$	$K_1\ mm^{-2}$	K_2	$K_g\ mm \cdot s^{-1}$	δ
0.01	$1.105 \cdot 10^9$	9	$1.3 \cdot 10^{-4}$	0.937
0.1	$1.55 \cdot 10^9$	6.9	$9 \cdot 10^{-4}$	2.245

Moreover, during PDRX ($|\dot{\varepsilon}_{xx}| = 0$), the parameter δ will take the value of 9.18 following the findings in [51]. Also, as explained in section 5.4.1, during PDRX the parameter ρ_c will be computed using the apparent effective strain rate $\dot{\epsilon}_s$ (see Eq. 5.10 and Fig. 5.17.right) instead of the effective strain rate $\dot{\epsilon}$ (equals to 0 in this regime). Finally, outside the range of interpolation, the values are computed as follows: if $|\dot{\varepsilon}_{xx}| > 0.1$ the values of all parameters will take the same values as for $|\dot{\varepsilon}_{xx}| = 0.1\ s^{-1}$, similarly, the same strategy applies when $|\dot{\varepsilon}_{xx}| < 0.01$, using the values for $|\dot{\varepsilon}_{xx}| = 0.01\ s^{-1}$ (see Fig. 5.16 for an illustration of the values of $K1$, $K2$, Kg and δ in function of $|\dot{\varepsilon}_{xx}|$).

Four cycles of deformation/coarsening will be applied as illustrated in Fig. 5.17, Fig. 5.17.right shows the computed values of the effective strain rates $\dot{\epsilon}_s$ and $\dot{\epsilon}$,

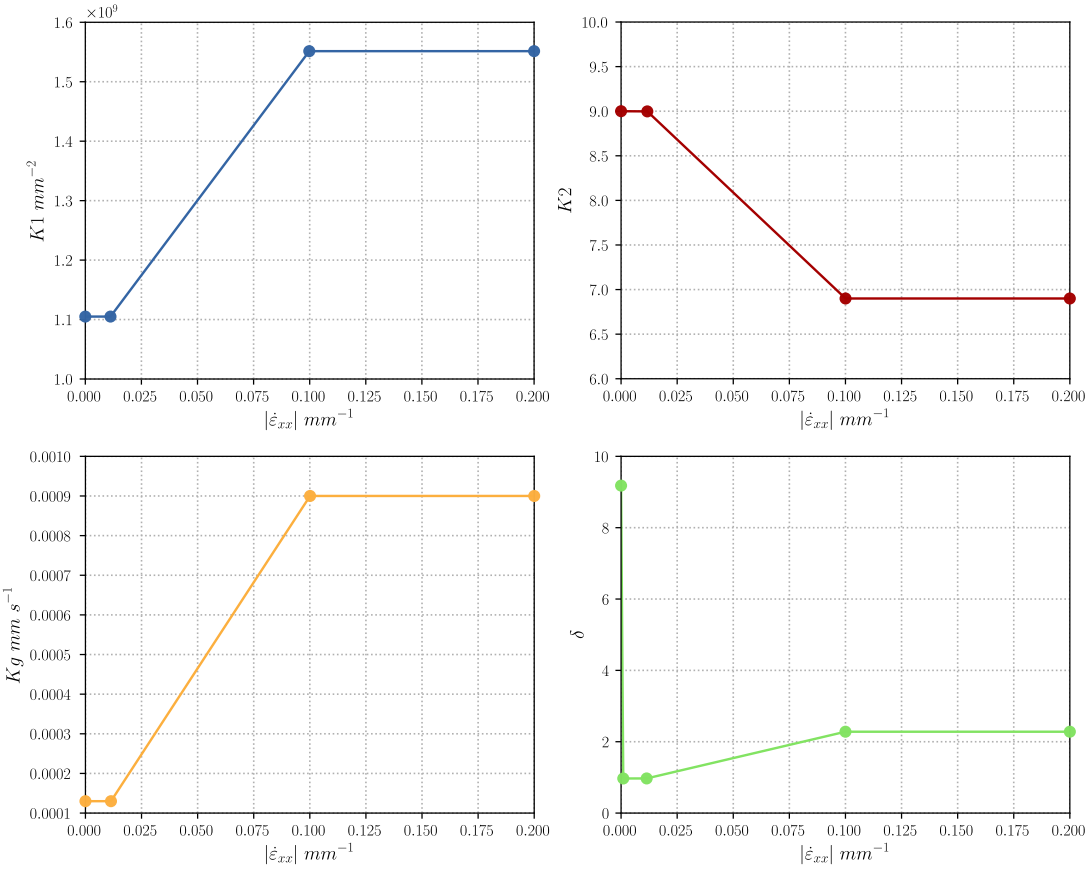


Figure 5.16: Evolution of the parameters of table 5.1 in function of $|\dot{\varepsilon}_{xx}|$.

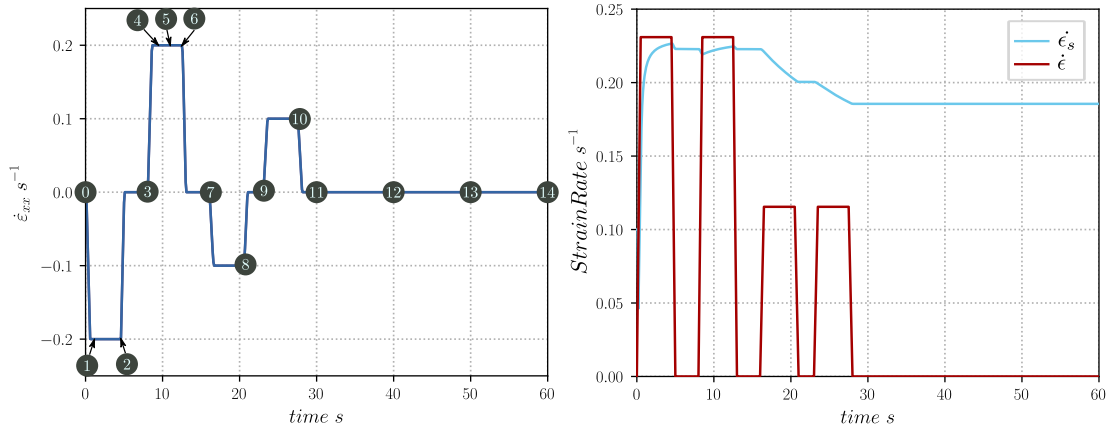


Figure 5.17: Deformation loading strategy for the DRX and PDRX case: right: the computed values of the effective strain rates $\dot{\epsilon_s}$ and $\dot{\epsilon}$, left: the strain deformation component $\dot{\varepsilon}_{xx}$, where multiple markers have been drawn, corresponding to different states during the simulations.

while Fig. 5.17.left shows the strain deformation component $\dot{\varepsilon}_{xx}$, where multiple markers have been displayed, these markers correspond to different states along

with the simulation that will be useful when analyzing the results.

Statistical comparisons of the TRM model and the response obtained by a FE-LS approach presented in [5, 167, 7, 170] will be given. This approach uses a more classic method of mesh adaptation during calculations where the interfaces are captured with an anisotropic non-conform local refined mesh. This methodology will be denoted in the following as the Anisotropic Meshing Adaptation (AMA) model.

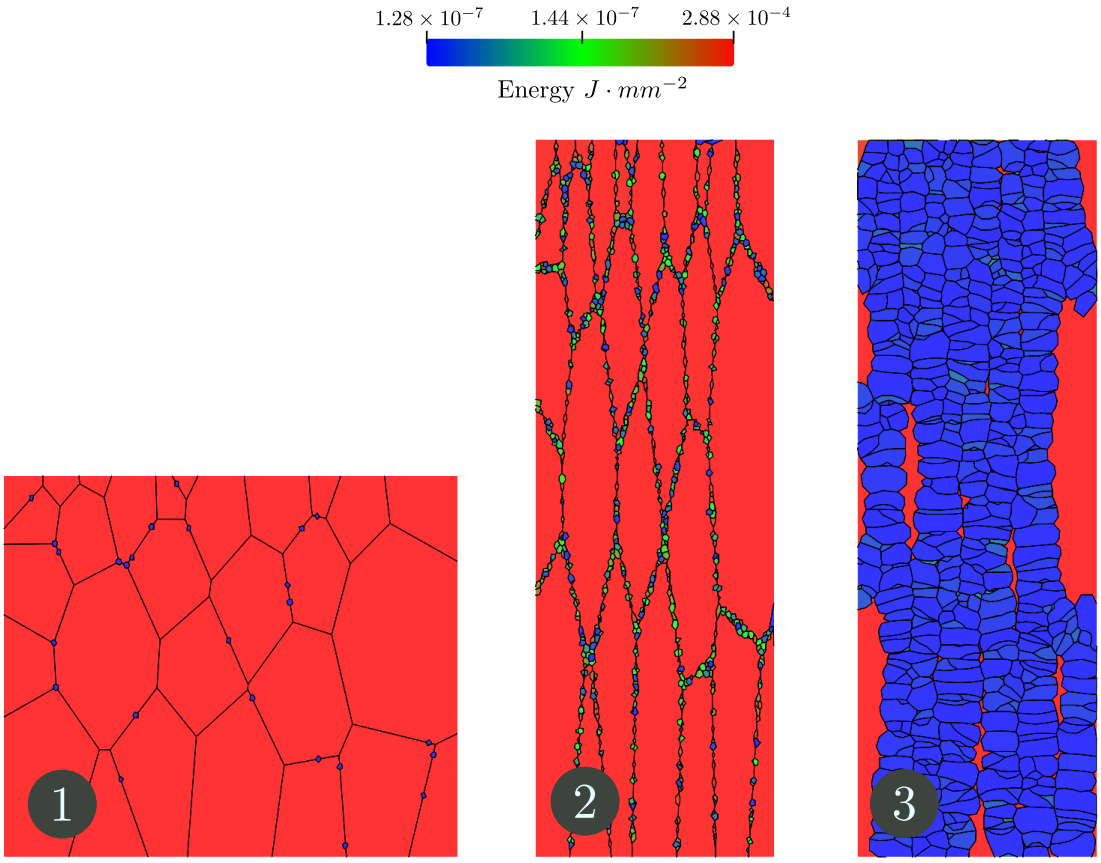


Figure 5.18: States 1 to 3 (see figure 5.17) obtained with the TRM model. 1: the firsts nuclei appear, 2: end of the first stage of deformation, 3: end of the first grain coarsening stage.

A well-known behavior of full-field simulations of microstructural evolutions is that the reduced mobility (γM product) is classically impacted by the choice of the numerical method and is not only a universal physical parameter. In other words, the reduced mobility is a physical parameter that needs to be identified comparatively to experimental data. This identification may lead to different values depending generally on the numerical method used [215]. In chapter 3, the reduced mobility was adjusted in order to minimize the L2-difference between the mean grain size evolution curves considering TRM or AMA numerical

strategies. The same methodology was used here in the global thermomechanical paths leading to an increase of 40% in the optimal identified reduced mobility.

Here, we have chosen the AMA case as a reference even though there is no way to know which model gives the most accurate response to the given physical problem; this choice on the other hand is given as an example of how the TRM model can indeed obtain similar responses to well-established models in the field of microstructural evolutions.

Multiple microstructural states have been retrieved from the results given by the TRM model. These states are marked with numbers corresponding to the states of Fig. 5.17:

States 1 to 3 are given in Fig. 5.18, here state 1 illustrates the apparition of the firsts nuclei in the positions where the dislocation density field reaches its value $\rho \geqslant \rho_c$. Then state 2 gives the end of the first stage of deformation where more nuclei have appeared, note that the value of the SE in some of the small grains is different from others, these grains have been present longer in the domain and consequently have been subjected to strain hardening, contrary to the nucleus that have appeared later, during or at the end of this deformation stage. Finally, state 3 shows the end of the first grain coarsening stage, where nuclei have had time to grow as a product of the high difference in energy with their surroundings.

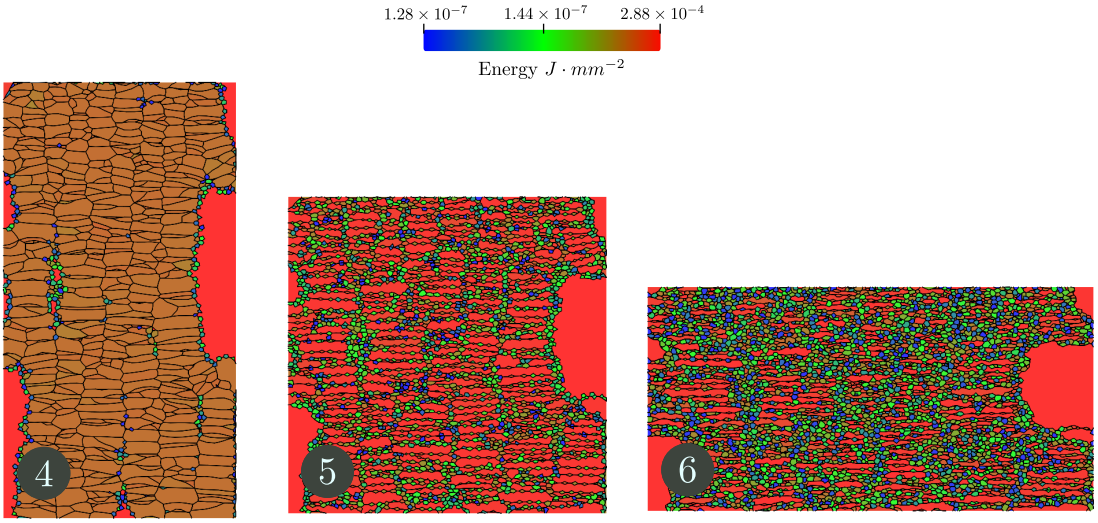


Figure 5.19: States 4 to 6 (see figure 5.17) obtained with the TRM model. 4: the nuclei appear on the regions where $\rho > \rho_c$, 5: all the domain is now above the value of ρ_c hence the nucleation occurs everywhere, 6: end of the second stage of deformation, here the maximum number of grains is reached (4250 grains).

States 4 to 6 are presented in Fig. 5.19. In stage 4 only a small percent of the domain have a dislocation density of at least ρ_c and nucleation is restricted to

these zones, contrary to stage 5, where a bigger part of the domain has reached the value of ρ_c, consequently, new grains appear everywhere. Finally, the end of the second deformation stage is given in state 6 where the first peak of number of grains is reached (4250 grains).

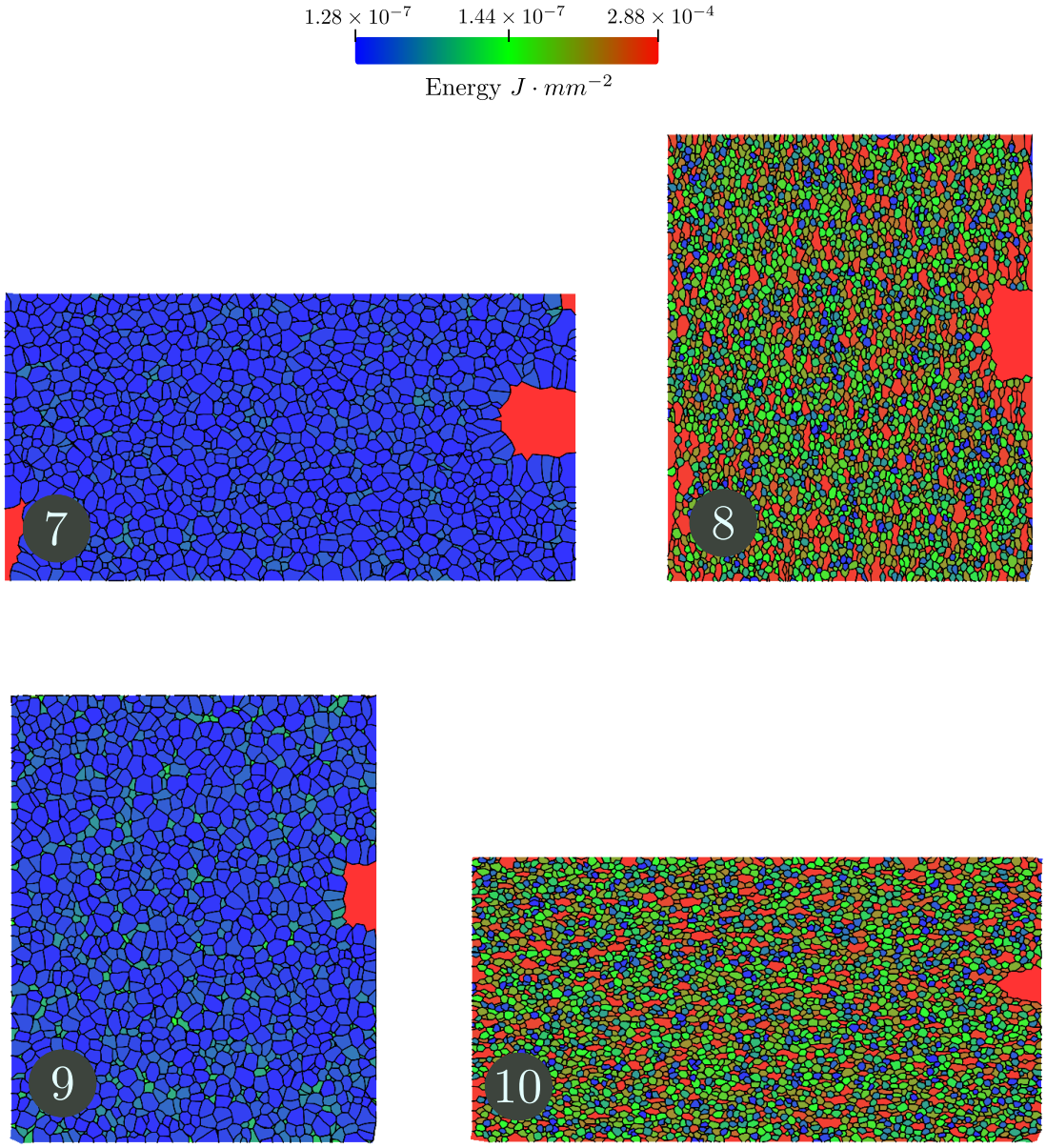


Figure 5.20: States 7 to 10 (see figure 5.17) obtained with the TRM model. 7: end of the second grain coarsening stage, the number of grains drops very quickly given by the increase of the value of δ from 2.245 to 9.18 (its dynamic vs its static value), 8: end of the third deformation stage, 9: end of the third grain coarsening stage, 10: end of the fourth deformation stage.

States 7 to 10 are given in Fig. 5.20, these steps are representative of the ends

of the third and fourth deformation/coarsening cycles, where during the deformation the nucleation process increases the number of grains while in the grain coarsening stages the high value of the parameter δ (2.245 to 9.18 its dynamic vs its static value) makes the grain number decrease rapidly (see Fig. 5.25.a) for the evolution of the number of grains).

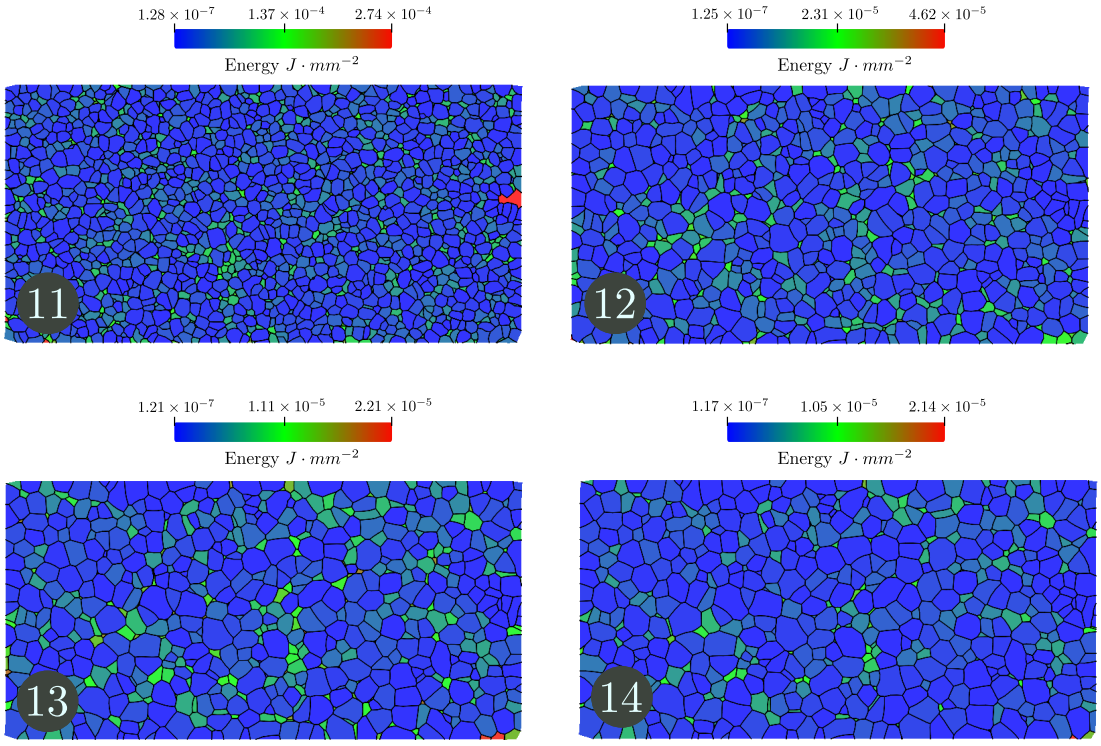


Figure 5.21: States 11 to 14 (see Fig. 5.17) obtained with the TRM model. these state correspond to a value of time t of 30, 40, 50 and 60 seconds respectively.

States 11 to 14 are provided in Fig. 5.21, these states correspond to a value of time t of 30, 40, 50 and 60 seconds respectively. In this range of time no deformation is considered. Note how the limits of the scale in Fig. 5.21 change as a product of the disappearance of high energetic grains and to the annihilation of dislocations simulated through Eq. 5.8.

Statistical values for the states 4 to 6 and 11 to 14 are given in figures 5.23 and 5.24 respectively. The grain size distributions for the TRM model without a mobility increase and with a mobility increase of 40% have been plotted along with the response given by the AMA case. Similarly the evolution of the mean grain size are provided for all simulations in Fig. 5.22.left and the L2-difference to the AMA case is given in Fig. 5.22.right

Finally, the evolution of some representative values are given in Fig. 5.25: the evolution of the number of grains, the recrystallized fraction, the mean value of

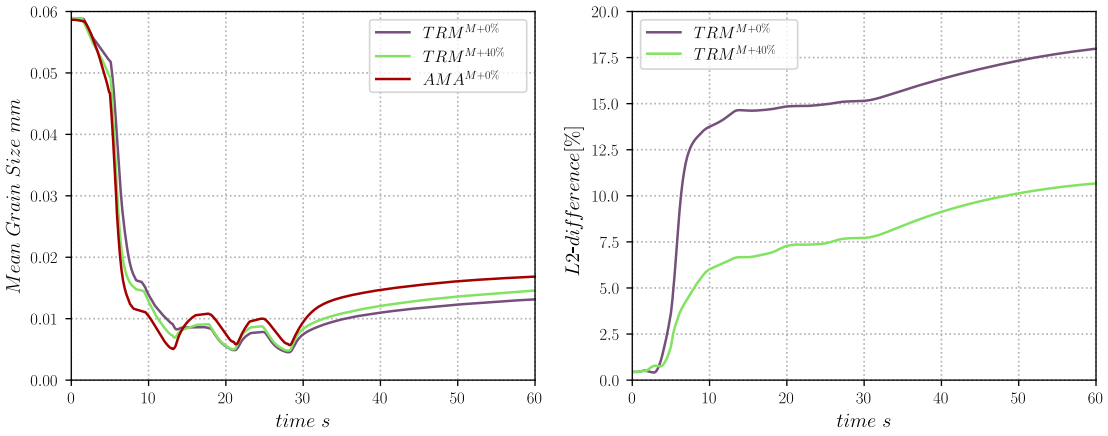


Figure 5.22: Evolution of the Mean grain size (left) for the TRM model and the L2-difference with the AMA simulation (right).

ρ pondered in surface ($\overline{\rho}$) and the total perimeter of the grains whose dislocation density is greater than ρ_c (P_c) are provided.

These results show a good agreement between the general behavior of the TRM model and the behavior of the AMA simulation when an increase of 40% is considered to the reduced mobility $M\gamma$ value (following the findings of chapter 3). The computational cost for the different iterations of the TRM model is given in Fig. 5.26, where for the slower simulation, the time needed for its completion was of 25 min while the fastest took 20 min, a very small CPU-time compared to the time needed for the AMA case (4 hours and 38 min).

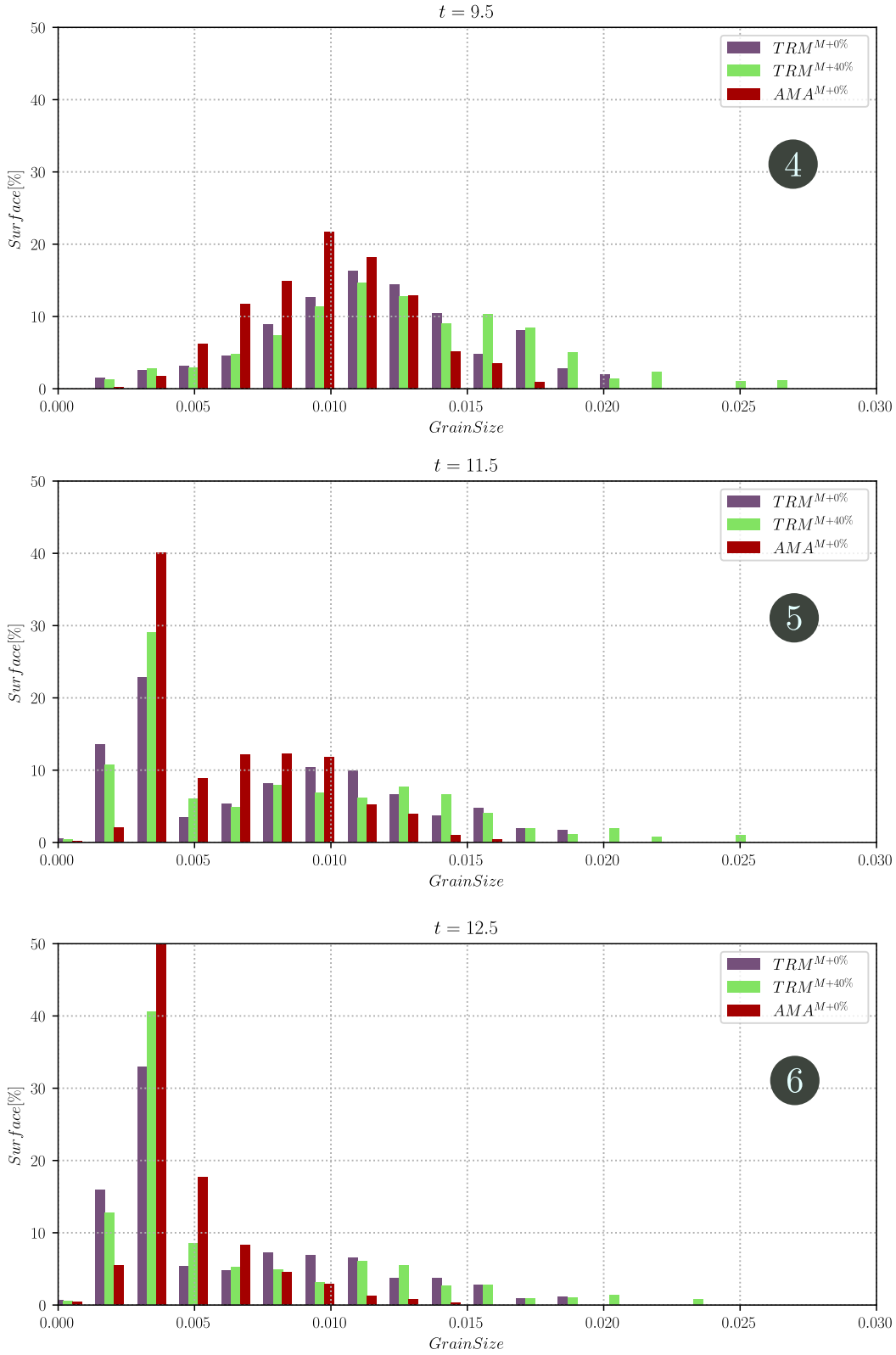


Figure 5.23: Grain size distributions pondered in surface for the states 4 to 6 (example of a deformation Stage). A peek on the nucleus size can be observed.

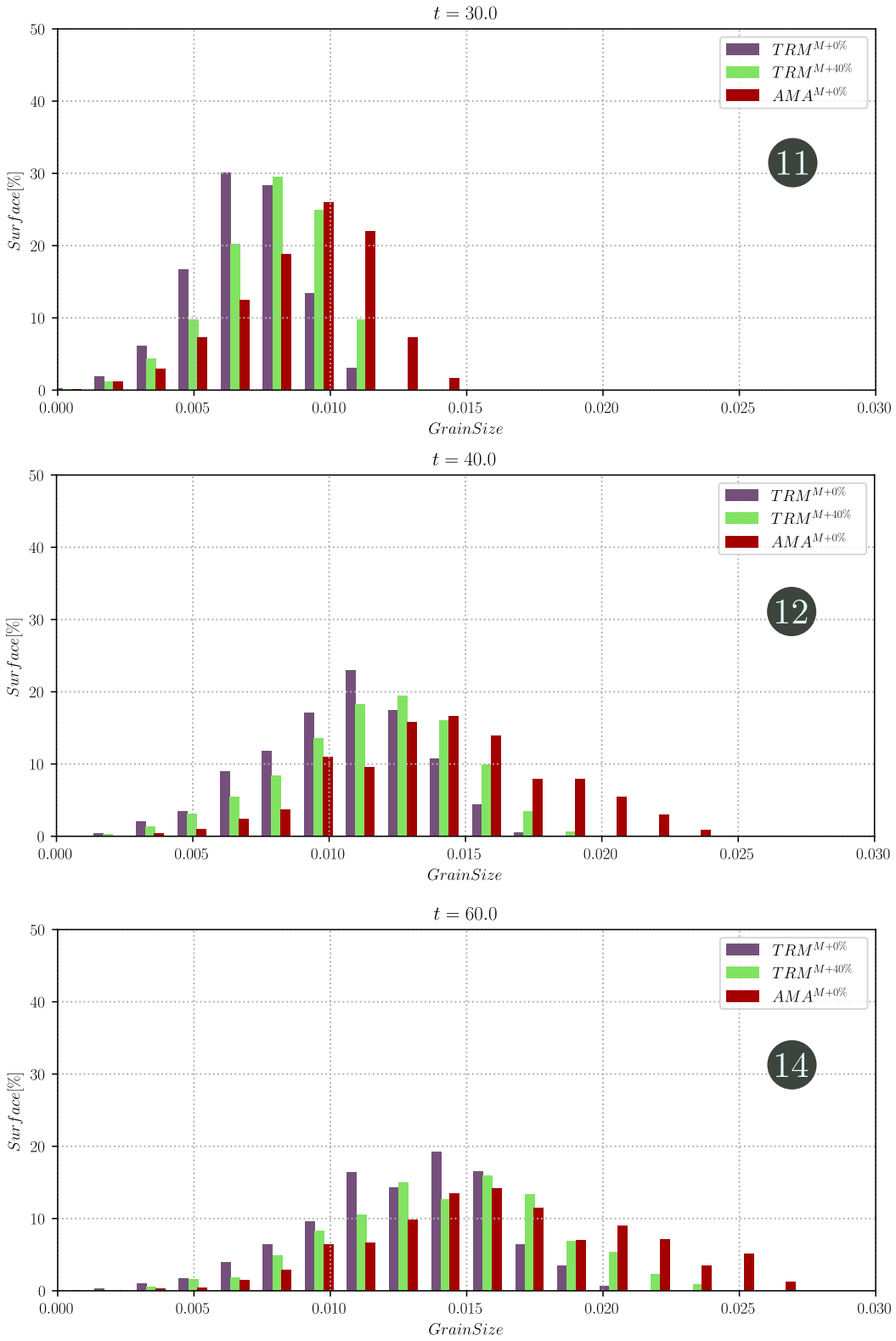


Figure 5.24: Grain size distributions pondered in surface for the states 11, 13 and 14 (example of a grain coarsening stage). The values are distributed more evenly on the size range (x axis) as a product of the grain growth.

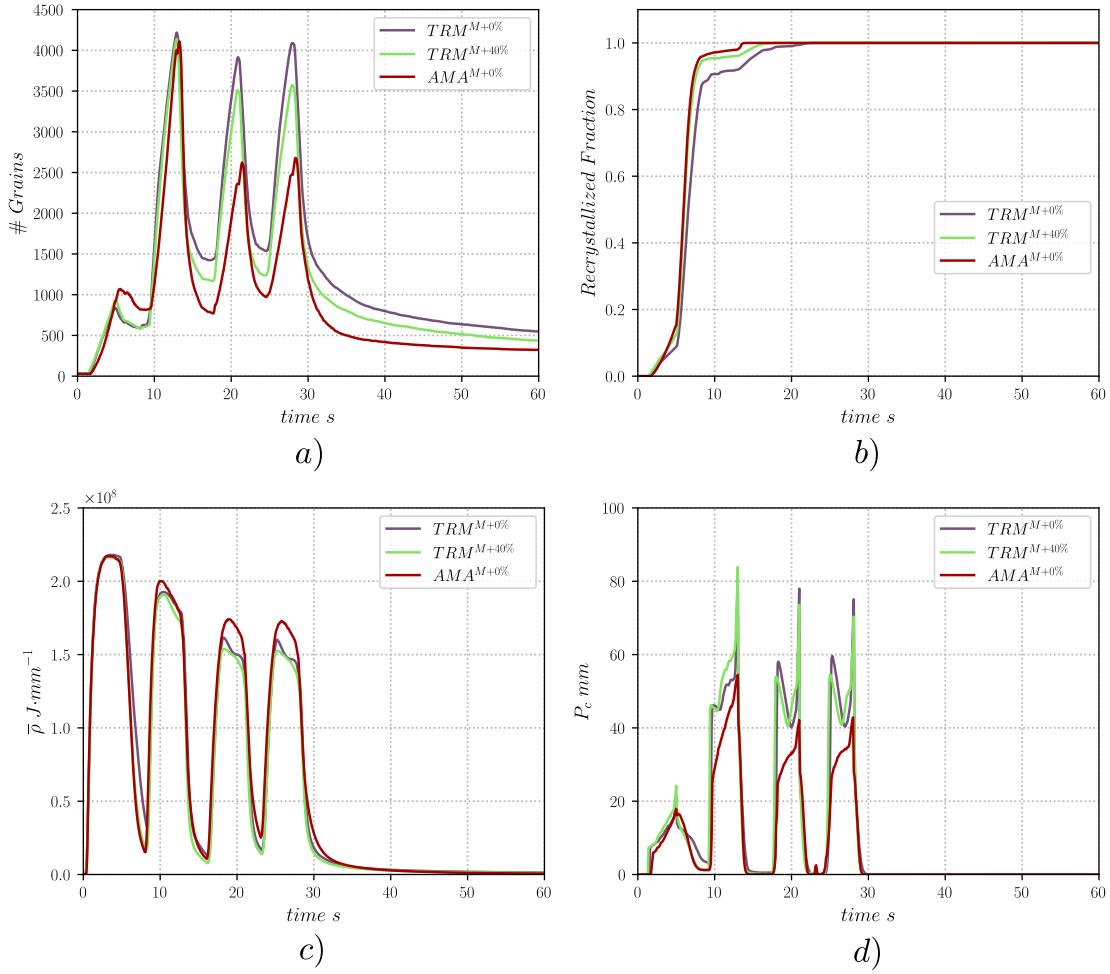


Figure 5.25: different values as a function of time for the DRX and PDRX test case, a) Number of grains, b) Recrystallized fraction, c) Mean value of ρ pondered by surface, and d) Critical perimeter for the computation of the nucleation rate in Eq. 5.11.

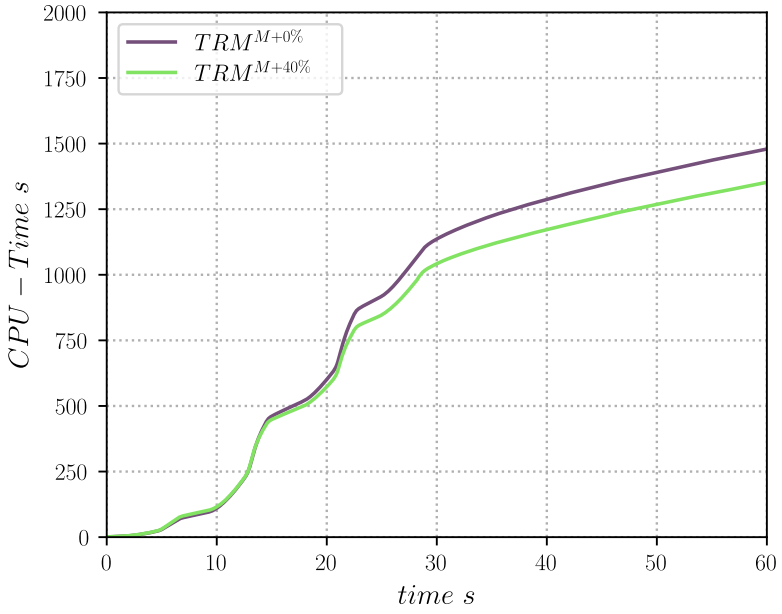


Figure 5.26: CPU-time for the different simulations using the TRM model, the computational cost drops as the number of simulated grains decreases.

5.6 Discussion and conclusion

In this chapter, the TRM model presented in chapters 3 and 4 in the context of GBM by capillarity has been adapted in order to take into account bulk terms due to the SE during plastic deformation. This adaptation has made possible the integration of a recrystallization model to the TRM approach, for which a nucleation procedure has also been presented.

The algorithms presented in section 5.3.1 and represented by Eq. 5.4 for the computation of the velocity at multiple junctions, although intuitive have not been published before to the knowledge of the authors, only [142] shows a similar (more indirect) approach in the context of vertex simulations.

Results for the circle test case and tripe junction case have demonstrated the high accuracy of the TRM model in the modeling of boundary migration due to capillarity and SE, where in the normal context (for typical grain boundaries and multiple junctions), an error no greater than 2% was found. Also, the circle test case showed the typical behavior of a nucleus when subjected to a wide range of SE around its metastable point and helped define the safety factor ω used in Eq. 5.12, defining the minimal radius to nucleate in the context of the TRM model.

Finally, a DRX/PDRX test case was considered in order to test the recrystallization model provided in section 5.4 for 304L stainless steel at 1100 $°C$. A reference test case using the same ReX model but with a FE-LS strategy was also considered (AMA case) [5, 167, 7, 170]. Following the findings of chapter 3, an optimal reduced mobility was calibrated to perform the tests (40% higher than the mobility used in the AMA case context). Results show a very good agreement between the two models. Moreover, the CPU-times of the TRM model were much lower being between 20 to 25 minutes against the 4 hours and 38 minutes needed for the AMA case for its completion.

The works developed in this chapter regarding the new possible topological operations, inherent to GBM governed by different driving pressures can also be used in the context of anisotropic grain boundary properties, as in such a context, boundaries may exhibit similar behaviors (sub-segmentations of preexistent interfaces, see Fig. 5.3). The application of the TRM to an environment with high *heterogeneities* on the definition of grain boundary properties will be presented in chapter 6.

Résumé en Français du Chapitre 5

La nouvelle méthode TRM introduite aux chapitres 3 et 4 est étendue pour étudier des phénomènes relatifs à la recristallisation (ReX) dynamique (DRX) et post-dynamique (PDRX). À cet effet, plusieurs modifications et nouveaux développements ont été réalisés:

- Une nouvelle formulation discrète pour le calcul de la vitesse aux jonctions multiples, issue des pressions exercées par l'énergie stockée, a été proposée.
- Des modifications sur le comportement du remaillage ont été réalisées, afin de prendre en compte un spectre plus large des changements topologiques pouvant apparaître à cause du nouveau terme de vitesse.
- Une procédure de remaillage pour l'apparition de nouveaux grains (germes) a été établie afin de créer les nouveaux joints de grains.
- Des lois de nucléation phénoménologiques, utilisées récemment sur des simulations de type LS-EF, ont été implémentées dans le contexte du modèle TRM.

Ces nouveaux développements permettent la simulation de cas académiques portant sur le comportement du modèle face à des mécanismes de type ReX, où plusieurs forces motrices agissent sur les interfaces (capillarité et énergie stockée). Ces résultats montrent un bon comportement du modèle et une erreur faible sur les différents tests.

Finalement, un chemin thermomécanique réaliste (au sens industriel) a été considéré. Il a permis de mettre en évidence la capacité de l'approche TRM de modéliser, en grandes déformations, des mécanismes de type DRX et PDRX. Les résultats montrent un bon accord avec les prédictions LS-EF, mais en des temps de calculs bien meilleurs (20 à 25 minutes pour l'approche TRM, 4h 38min pour l'approche LS-EF)

Les développements de ce chapitre concernant les nouveaux changements topologiques peuvent aussi être appliqués au contexte où les propriétés de joint des grains ont un caractère anisotrope. Cet aspect est discuté dans le chapitre suivant.

Chapter 6

Towards the modeling of heterogeneous and anisotropic phenomena with the TRM model

This chapter presents the implementation and application of the new front-tracking methodology introduced in chapter 3, to the context of anisotropic grain boundary motion. The new formulation of the boundary migration velocity can take into account any source of heterogeneity both at grain boundaries as well as at multiple junctions. Special attention is given to the decomposition of multiple junctions (MJs) of high order, for which an algorithm is proposed based on the reduction of the local grain boundary energy. Numerical test cases are provided at the end of the chapter in the context of highly *heterogeneous* systems, and comparisons with a recently developed LS-FE strategy are given. Finally, the computational performance of the model will be studied comparing the CPU-times obtained with the same model but in a homogeneous context.

6.1 Introduction

The grain boundary motion of polycrystals has been studied for many decades, both, from an experimental and a numerical point of view. The majority of experimental observations at this scale suggest that the migration of boundaries is in general, a strongly heterogeneous phenomenon, involving complex dynamics and topological transformations of the grain boundary network. However, it is generally accepted in the literature, that the microstructure of given materials (such as 304L stainless steel) behave homogeneously enough to ignore their heterogeneities when polycrystal modeling is considered. This hypothesis is used in numerical environments to propose FF models of microstructural evolution, using isotropic grain boundary properties, e.g. isotropic GG.

This is, however, a strong hypothesis when properties of GBs in a material are known as highly heterogeneous, for example, when a strong texture with particular γ values are involved, or when special GBs (e.g. twin boundaries [25]) are present.

Commonly in the literature [23], the source of the anisotropy of GBs have been considered given as a function of the disorientation angle θ and the inclination of the interface. Typically θ is function of the three Euler angles $(\varphi_{e1}, \Phi, \varphi_{e2})$ defining the crystallographic misorientation between two adjacent grains, and the inclination can be considered through the *local* normal vector $\vec{n}$ of the interface. This gives a total of 5 degrees of freedom (DOF) system defining the values of GBs properties, which can be expressed as a function of the tuple $(M_{lw}, \vec{n})$. These kinds of systems at the polycrystal scale need to be modeled through the use of a numerical approach *able to take into account* this kind of data set. As such, in the same manner as in [217], here we will differentiate three kinds of numerical models in this context: isotropic, heterogeneous, and anisotropic models. **Isotropic** models are those considering GBs properties as constant for its formulation (e.g. the LS model given by Eq. 2.10 is a simplification of Eq. 1.25 where γ and M have been considered as invariant in space). On the contrary, **Anisotropic** models, are those using a formulation where, any assumption regarding the invariability of these quantities in space is discarded, being able to use properties dependent on the tuple $(M_{lw}, \vec{n})$ (i.e. $X_{(M_{lw},\vec{n})}$ where X is either γ or M).

Of course, anisotropic models are much more complex than those using the isotropic hypothesis, since, in this context, special attention must be given, for example, to the meaning of the surface tension component of interfaces, as one must be aware that *torque terms*, derived from the variation of the GB energy γ on the parametric space of a surface may appear [218]. As such, deriving a mathematical model ready to use an anisotropic set of GBs properties is a complex task, and historically, authors in the literature have proposed alternatives: heterogeneous models. **Heterogeneous** models consider within their formulation

the existence of a variation of properties, in function of only the misorientation between two adjacent grains M_{lw} ($X_{(M_{lw})}$), neglecting its dependence on $\vec{n}$. In this context, a given GB is given homogenized intrinsic properties (constant in its parametric space), but not equal to the ones of other GBs. i.e. grain boundary properties only change at multiple junctions (MJ) (or multiple lines in 3D) when crossing from one GB to another.

Several approaches have been proposed in the literature to model heterogeneous/anisotropic GG. Beginning with the Monte Carlo and extending to Phase-Field, Level-Set and Vertex approaches, heterogeneous ($X_{(M_{lw})}$) [63, 143, 144, 130], and anisotropic ($X_{(M_{lw},\vec{n})}$) [23, 145, 17] models have been proposed. However, it has been mentioned in the previous chapters that all these methods are constraint by different reasons each, typically: i. the use of regular grids [152, 153] (difficulty to model deformation), ii. high computational cost [23], iii. the unclear definition of the grain boundary properties and the use of its derivatives in the model formulation (quite common in heterogeneous formulations), and iv. the no-discretization of grain interiors [14, 17] (leading to the difficulty to model intragranular phenomena). Additionally to these aspects, in the context of anisotropic boundary properties modeled using Phase-Field models, although being an appropriate numerical environment, showing interesting results in this context, one should be aware of inherent numerical instabilities, especially for high heterogeneous/anisotropic systems [130, 120].

As an alternative to model anisotropic or heterogeneous microstructural evolutions, we propose the TRM model presented in chapter 2. In this chapter, the needed implementations in order to model fully anisotropic grain properties with the TRM model will be presented. Special attention will be given to the development of a robust high order MJ decomposition algorithm and to the reformulation of the velocity equation at triple junctions extending the methodology presented in [14] to an anisotropic context, using the notions used in [17] for its discrete formulation. Finally, for the purpose of this chapter, the TRM model will be tested in multiple heterogeneous environments identical to the ones presented in [143, 144] while the numerical test in a fully anisotropic environment is a perspective and will be presented in a forthcoming publication.

6.2 Numerical method

In this section, the necessary concepts and new implementations needed by the TRM model in order to model GBM using anisotropic grain boundary properties will be explained. In fact, the topological changes that may occur in this context have the same level of complexity as the ones produced by GBM under the influence of stored energy, hence no supplementary development is needed regarding this aspect. However, notions regarding the anisotropic formulation of a velocity by capillarity in such an anisotropic context, have not been covered in the works presented in the previous chapters. Additionally, in chapter 3 it was mentioned that the decomposition of MJs of high order had been simplified when using isotropic GB properties, and a more developed algorithm was needed to obtain valid predictions in an anisotropic context. These notions will be covered in this section, along with a quick note on the computation of misorientation and disorientation angles.

Hereafter, we will consider that $\gamma_{(M_{lw},\vec{n})}$, while the mobility term M will be considered constant in space. This framework is largely accepted and used at the polycrystal scale. However, it also must be highlighted that the description of γ or the reduced mobility ($M\gamma$) is an active topic of research due to misfits between some experimental works and the actual picture of these parameters [219, 220]. Recent advances concern the introduction of models using tensorial mobility and disconnection models used at the GB scale [221, 222].

6.2.1 Grain boundary motion by capillarity: Anisotropic context for the TRM model

In [17], a formulation for the computation of the velocity of GBs and triple junctions using anisotropic GB properties was proposed in the context of the Vertex model. This formulation uses the tensile character of the capillarity forces exerted at every node based on a discrete analysis, similarly to the one used in this work in chapter 5 for the computation of a velocity from a stored energy field at triple junctions. The model in [17] writes for the velocity at MJs:

$$\vec{v_{ci}} = M_i \left(\sum_j \frac{\gamma_{ij}\vec{t}_{ij} + \tau_{ij}\vec{n}_{ij}}{|\overline{N_iN_j}|} \right), \tag{6.1}$$

where the index i denotes the node representing the MJ N_i and j their connection to node N_j, M_i is the mobility of node N_i, γ_{ij}, $\vec{t}_{ij}$ and $\vec{n}_{ij}$ are respectively the boundary energy, the unit tangent vector and the normal of the segment $\overline{N_iN_j}$. Note that $\gamma_{ij} = \gamma_{ji}$ but $\vec{t}_{ij} = -\vec{t}_{ji}$ and the direction of the normal $\vec{n}_{ij}$ is arbitrary. Finally, note the apparition of the term τ_{ij} which correspond to the torque experienced by the segment $\overline{N_iN_j}$ due to the change of the GB energy given by

its dependence on the inclination angle ω [218], this torque term is defined as follows:

$$\tau_{ij} = -\frac{d\gamma}{d\omega_{ij}}. \tag{6.2}$$

In [14], three formulations were given for the computation of the velocity at triple junctions in the context of isotropic GB properties, from which we have used the so-called model **II** to find our velocity at MJs in chapter 3, 4 and 5. This formulation can be rewritten in the context of heterogeneous grain boundary properties (hence, in the absence of torque terms) and for MJs of arbitrary order, in a very similar way as in Eq. 6.1:

$$\vec{v_{ci}} = M_i \left(\frac{\sum_j \gamma_{ij} \vec{t}_{ij}}{\sum_j |\overline{N_i N_j}|} \right), \tag{6.3}$$

where the only difference with Eq. 6.1 is that the terms in the numerator contribute all in the same amount to the summatory, instead of being escalated each by a the term $|\overline{N_i N_j}|^{-1}$. Indeed, in our experience, the homogenization of the contributions of the numerator term by the separated summatory $\sum_j |\overline{N_i N_j}|$ has proven to be more stable than the one given in Eq. 6.1, especially when the value of any $|\overline{N_i N_j}|$ approaches to zero (or when $|\overline{N_i N_j}| << |\overline{N_i N_k}|$ for all $k \neq j$). For this reason, the use of Eq. 6.3 is preferred here but maintaining the torque terms of Eq. 6.1:

$$\vec{v_{ci}} = M_i \left(\frac{\sum_j \gamma_{ij} \vec{t}_{ij} + \tau_{ij} \vec{n}_{ij}}{\sum_j |\overline{N_i N_j}|} \right). \tag{6.4}$$

This equation can be used along with Eq. 5.4 for the modeling of GBM under the influence of stored energy and anisotropic grain boundary properties.

Finally, note that torque effects do not only have to be accounted at MJs but they also appear at GBs. To formulate this for the velocity at GBs we will use a combination of the discrete approach given in [17] and the standard approach of the TRM model:

$$\vec{v_{ci}} = M_i \left(-\gamma_{ij} \kappa_i \vec{n}_i + \frac{\sum_j \tau_{ij} \vec{n}_{ij}}{\sum_j |\overline{N_i N_j}|} \right) \tag{6.5}$$

where the terms κ_i and $\vec{n}_i$ are computed in the conventional way using the numerical approximation by splines at the node i. Note that $\vec{n}_i \neq \vec{n}_{ij}$, as $\vec{n}_{ij}$ denotes the normal of the segment $\overline{N_i N_j}$ and $\vec{n}_i$ the normal to the numerical approximation evaluated at node i. We acknowledge that this formulation is somewhat unusual and it will be corrected in future works, with a formulation completely based on the approximation of interfaces by high order piece-wise polynomials (splines).

6.2.2 Minimal-state energy of high-order MJs

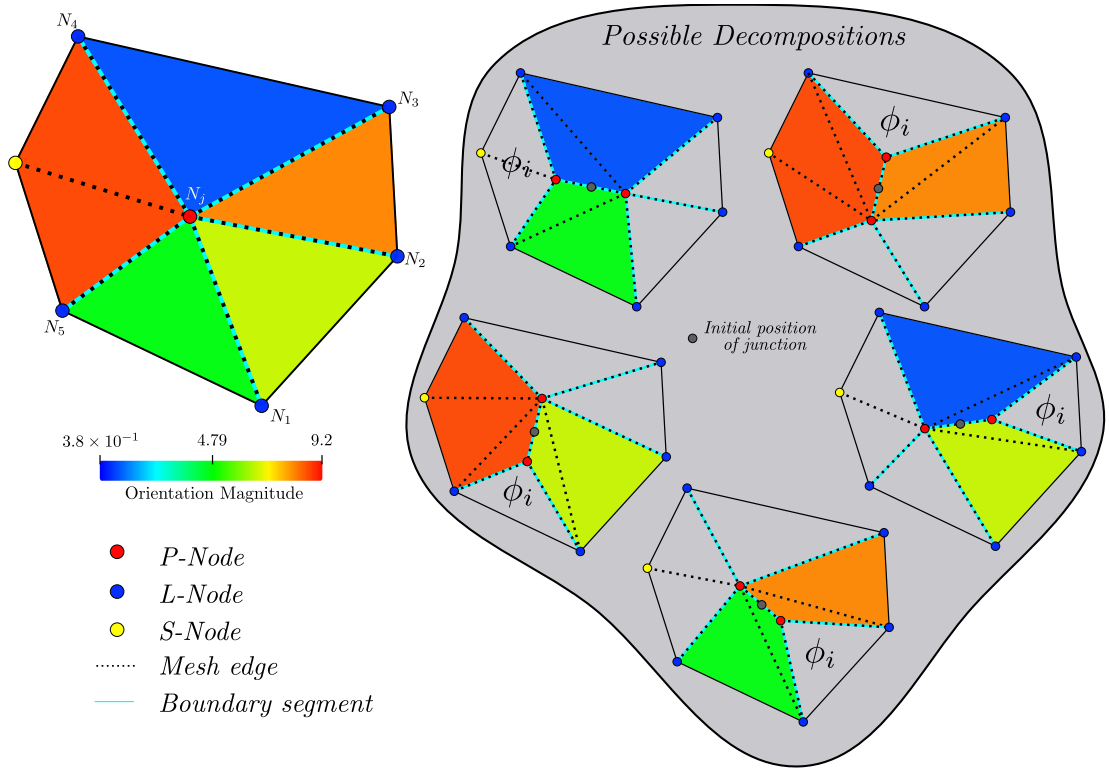


Figure 6.1: Possible decompositions of a MJ of fifth order, data regarding the orientation of each grain is given. One possible decomposition for every phase involved is depicted. The decomposition of a 5th order MJ results in a 4th order MJ and a 3rd order MJ.

In this section, we provide an algorithm for the decomposition of high order MJs when anisotropic GB properties are considered. The main challenge here is to explore all possible configurations that may proceed after a decomposition process. As such, the size of the possibilities set $P(z)$, is only dependent on the order of the MJ z to be decomposed. A 4th order MJ (i.e. 4 grain boundaries meeting in a point) can be decomposed only in two ways (see Fig. 3.17). However, the possibilities set $P(z)$ grows much higher when the MJ increases its order. Consider the configuration given in Fig. 6.1, here a 5th order MJ is given as well as the initial 5 possible decompositions by the separation of *consecutive interface pairs*. Note however that each possibility is conformed by one 4th order MJ and one 3rd order MJ, from which the 4th order MJ can be decomposed in two 3rd order MJ. In total, for a 5th order MJ, 5 possible decompositions are allowed when decomposing all MJ with $z > 3$ ($P^3_{(5)} = 5$ where the upper script means that all final MJ are $z = 3$, see Fig. 6.2). $P^3_{(z)}$ increases rapidly with the order z of the MJ: for $z = (2, 3, 4, 5, 6, 7, 8, ...)$, $P^3_{(z)} = (1, 1, 2, 5, 14, 42, 132, ...)$. In general, the number of possible combinations in this context is obtained by the use of the Catalan numbers [223] formula C_n:

$$P^3_{(z)} = C_{(z-2)} = \frac{(2(z-2))!}{(z-1)!(z-2)!}. \tag{6.6}$$

Of course, the probability of encountering a MJ of order z also decreases as z increases, as for a MJ of order z to form, it would require that all $P_{(z-1)}$ possible decompositions were *stable*. Of course, this notion of stability is related to the *total* minimum energy state able to be reproduced, for a given initial configuration. Note that this notion makes also possible that, one could obtain a *total* minimal energy state for a MJ with $z > 3$, in which case this MJ should not be decomposed [17]. As such, not only the configurations given by $P^3_{(z)}$ need to be considered, but also those intermediate (e.g. the ones given in Fig. 6.1), hence giving $P_{(z)} >> P^3_{(z)}$.

For the purpose of this work, we have simplified this aspect by accepting configurations presenting *local* minimal energy states and by not testing all possible configurations $P_{(z)}$. Algorithm 10 gives the procedure used in the TRM model to decompose a MJ. Here we have used the function $E_b(B)$ which gives the total surface energy of the internal boundary segments B, of a given element patch e_p, obtained thanks to the function $Boundaries(e_p)$, also, we use the function $GL(i, j, N)$ which returns a list of size i with the j_{th} set of consecutive[1] boundary segments attached to Node N (i.e. for the case given in Fig. 6.1, $GL_{(2,1,N_j)} = \{\overline{N_jN_3}, \overline{N_jN_4}\}$, $GL_{(2,2,N_j)} = \{\overline{N_jN_4}, \overline{N_jN_5}\}$, $GL_{(2,3,N_j)} = \{\overline{N_jN_5}, \overline{N_jN_1}\}$, $GL_{(3,1,N_j)} = \{\overline{N_jN_3}, \overline{N_jN_4}, \overline{N_jN_5}\}$...) and finally, the function $Boundaries(e_p)$. The algorithm first searches between each pair of consecutive segments, the one that would reduce the boundary energy the most if it is separated from the MJ (just as depicted in Fig. 6.1), and selects this configuration. Then, if this configuration actually reduce the initial GB energy given by the initial state, the initial configuration is replaced and the algorithm continues to the next MJ. If not, instead of searching between pairs, the algorithm will reiterate between consecutive triplets of lines if the order z of the MJ is sufficiently high (at least $z = 6$) and so on. Finally, if no configuration tested has a lower energy than the initial configuration, the algorithm considers the MJ as stable and continues to the next. Note that the decompositions are made one at a time for a given MJ each time Algorithm 10 is called. This means that a MJ of order $z = 5$ can be entirely decomposed in two increments and one with $z = 6$ in three.

Of course, this procedure accepts decompositions that could not have a *total* minimal energy state (the configuration with the minimum possible boundary energy) specially for MJ of too high order ($z > 6$). However, in practice, such configurations have a very low (almost null) probability of appearance in real microstructures.

[1] Consecutivity is measured in this context following polar coordinates (i.e. the angle made by a given line and the x axis)

Algorithm 10 MJ decomposition algorithm for the TRM model

1: **for all** Points: P_i **do**
2: **if** $z_{(P_i)} > 3$ **then**
3: $N_i \leftarrow$ Node representing P_i
4: $e_{p0} \leftarrow$ Elements(N_i)
5: $B_0 \leftarrow Boundaries(e_{p0})$
6: $E_0 \leftarrow E_b(B_0)$
7: $E_{min} \leftarrow \infty$
8: $S_0 \leftarrow$ tuple (e_{p0}, B_0)
9: $i \leftarrow 2$
10: **for all** number of connections of N_i : j **do**
11: $\{L_j\} \leftarrow GL_{(i,j,N_i)}$
12: Separate $\{L_j\}$ from N_i by adding a new Node N_j
13: Create new boundary (PP-Connection) $\overline{N_i N_j}$
14: $B_j \leftarrow \overline{N_i N_j} \cup B_0$
15: **if** $E_b(B_j) < E_{min}$ **then**
16: $E_{min} \leftarrow E_b(B_j)$
17: $e_{pj} \leftarrow$ Elements(N_j) $\cup\, e_{p0}$
18: $S_{min} \leftarrow$ tuple (e_{pj}, B_j)
19: **if** $E_{min} < E_0$ **then**
20: Replace S_0 by S_{min}
21: **else if** $i < z_{(P_i)}/2$ **then**
22: $i \leftarrow i + 1$
23: **goto** 10:

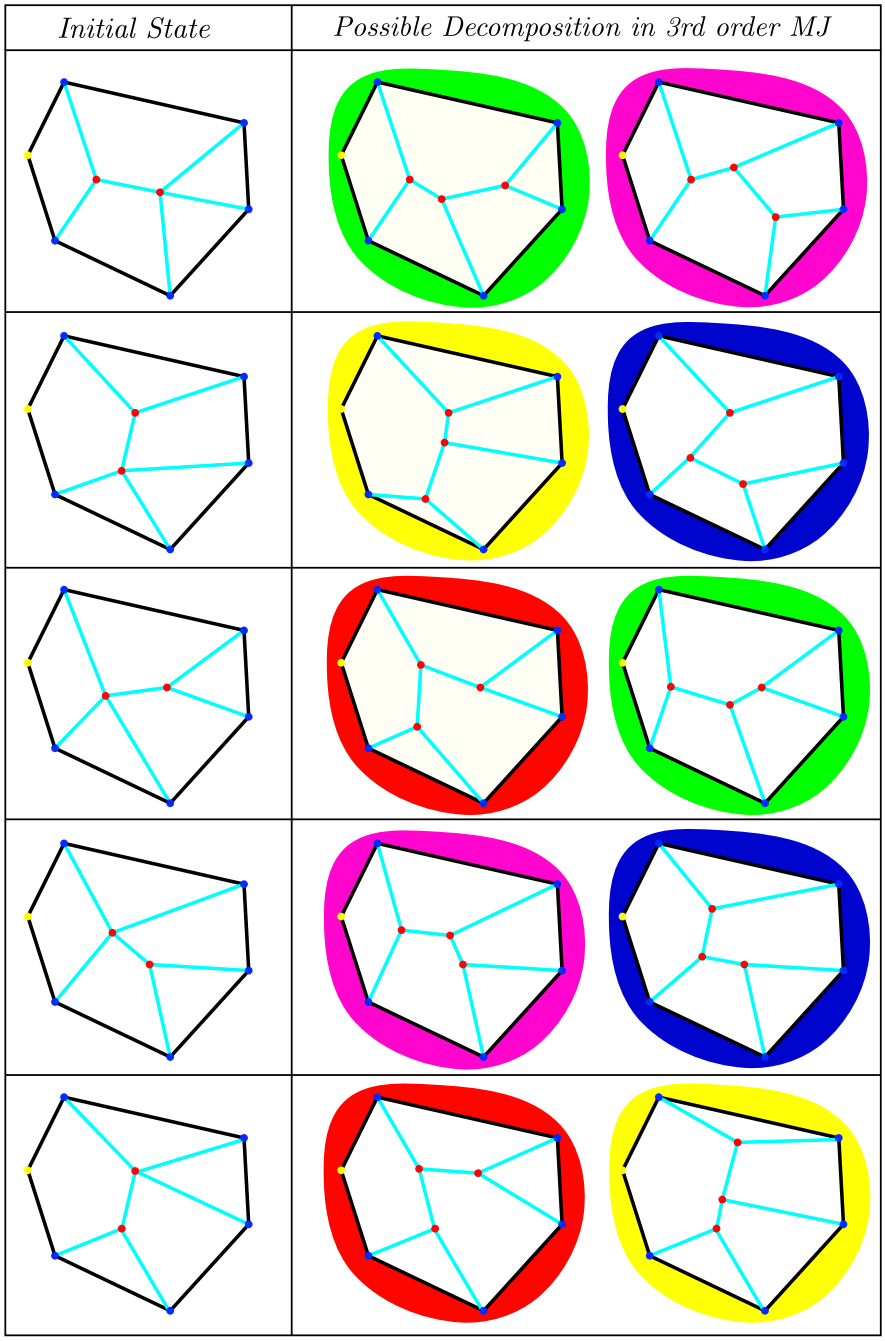


Figure 6.2: Final possible decompositions of the MJ of Fig. 6.1 into MJ of 3rd order, the background color of each final configuration matches similar configurations.

6.2.3 Computation of the disorientation angle

The computation of a misorientation and a disorientation angle in this work will be performed in a similar way as in [144]. A misorientation is computed between two neighbors grains G_l and G_m presenting the orientations O_l and O_m

respectively. Such a set of misorientation can be expressed as:

$$M_{lw}^{*} = O_l^{-1} O_w. \tag{6.7}$$

It is however necessary to compute a misorientation taking into account the minimization of the disorientation angle $\theta(O_l, O_w)$. Hence for all possible symmetric representation of the misorientation M_{lw}^{*}, with $(S_i, S_j) \in \mathbb{H}^2$ the space group of the crystal:

$$M_{lw} = \underset{\min_{i,j}\, \theta(O_l S_i, O_w S_j)}{S_i^{-1} M_{lw}^{*} S_j}. \tag{6.8}$$

The search of the minimal disorientation has been implemented using a brute force algorithm, every misorientation computation needs to iterate over all possible symmetric representations and select the one with the lowest θ. In this work, we will consider only cubic type crystals, hence 24 symmetric representations must be iterated for every misorientation computation.

In the context of the TRM model, these operations need to be done during two stages of the algorithm:

The first misorientation computation is held before the computation of the velocity of nodes v_i as all boundary properties need to be defined at this stage. The computation is done once per *Line*, which attributes all misorientations and disorientation angles for all L-Nodes and segments of the *Lines* entities. Note that a *Line* can only compute one misorientation, hence it is not necessary to compute it for every node belonging to this line. However, some edges still need to define their orientation, these are the ones describing a *PP-Connection*, namely, the edges defining a connection between two *Point* entities (see section 3.2.1 for more information about the data structure used by the TRM model). This computation is necessary as, even though the notion of grain boundary energy do not hold at MJs in the same way as for normal boundaries, the GB properties of all interfaces attached to the MJ are needed.

The second misorientation computation is performed during the decomposition of MJs for all possible new interfaces (see line 13-15: of Algorithm 10). This could be indeed a very demanding procedure as, for instance, each possible decomposition will have to search the minimal disorientation angle between all 24 equivalent symmetries defined for the crystal. We will study the relative cost of the misorientation computation at the end of section 6.3.

6.3 Numerical results

In this section the TRM model will be tested in a *heterogeneous* context, meaning that the influence of the inclination angle ω over the value of γ will be ignored. γ is then considered only dependent on the disorientation angle θ and being for all segments defining the boundary between two given grains but different from all other boundaries. In such a context, the torque term τ is equal to 0 for all boundary segments and the velocity of all nodes can be computed using Eq. 6.3.

All tests performed in this section have been inspired by the ones presented in [143, 144] in the same heterogeneous context. In [143], the classical FE-LS formulation of [137, 5, 146] has been reformulated with the main objective of simulating this kind of phenomena, taking into account the gradients terms produced by a variation of γ, that were otherwise neglected in a homogeneous context (where γ is constant in Ω). Given the fact that this formulation takes into account all variational terms that are relevant in this context, it will be named hereafter the *heterogeneous* FE-LS formulation. The numerical testing of this approach was divided into two parts: firstly, in [143], the numerical analysis is focused on the evolution of multiple junctions, as a mean to test the *heterogeneous* FE-LS formulation presented in the same publication. Secondly, in [144], the same *heterogeneous* FE-LS formulation was tested in the context of heterogeneous GG, using different formulations for the computation of the grain boundary energy γ, as a function of the disorientation angle.

We will reproduce these studies with the TRM model.

6.3.1 Triple junction test case

The first test corresponds to the triple junction test, also used in chapter 5. Here, however, stored energy values are not considered, and the motion of the triple junction is dictated by the GB energies of the interfaces meeting at the central node. Fig. 6.3(left) illustrates this aspect, where the term γ_{ij} denotes the GB energy between grains G_i and G_j and ϕ_i is the angle measured at the junction between the interfaces of grain G_i and the other two grains. For the purpose of this test, $\gamma_{13} = \gamma_{23}$, this will provoke a vertical movement of the junction for any value of $\gamma_{12} \neq \gamma_{23}$, until it arrives at its equilibrium position. As such, we will study the motion and the equilibrium of the junction in function of the ratio $r = \gamma_{23}/\gamma_{12}$. In this test, dimensionless simulations will be considered, the value of the grain boundary energies $\gamma_{13} = \gamma_{23} = 0.1$ will be held constant as well as the mobility term $M = 1$. Moreover, for practical reasons[2], Dirichlet boundary conditions with $\vec{v_i} = 0 \ \forall N_i \in \partial\Omega$, hence impeding the movement of the nodes at

[2]For $r < 1$, the MJ moves downwards, which when using Neumann type boundary conditions, induce a global movement of the interfaces in the same direction, and eventually leads to the contact of the junction with the base of the triangle. This behavior is not wanted in this context.

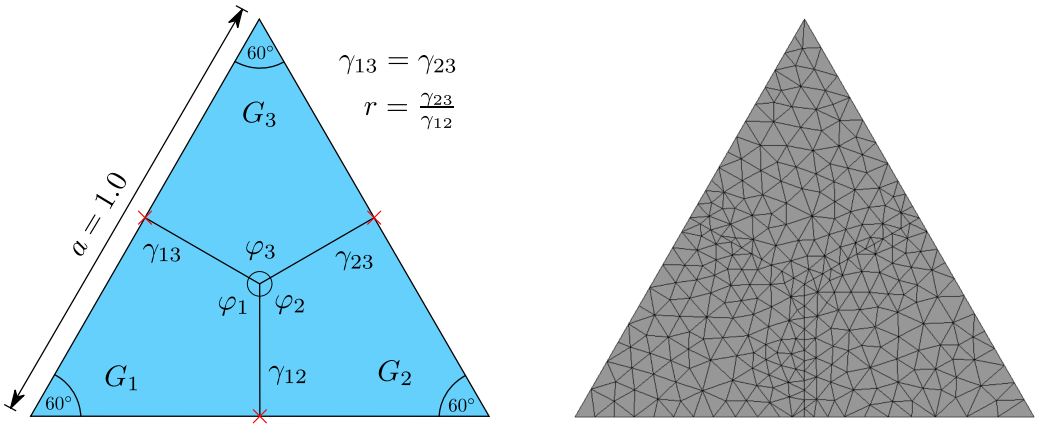


Figure 6.3: Initial state for the triple junction test, three phases immersed in a domain in the shape of an equilateral triangle, the intersection of the domain with the interfaces is fixed (Dirichlet conditions $\vec{v_i} = 0 \ \forall N_i \in \partial\Omega$). a) Initial configuration and b) initial mesh.

the intersection of the GBs and the edges of the triangular domain. Finally, the mesh size parameter $h_{trm} = 0.006$ (see chapter 3. for the description of the metric of the mesh in function of this parameter) and the time step $dt = 5e - 5$ will be used for all tests. These values were selected correspondingly to the limit of the stability region of the TRM model when using piece-wise polynomials (splines) as a mean to obtain values of curvature and normal (see chapter 3). The initial mesh is depicted in Fig. 6.3(right).

While there is not an analytical formulation for the movement of multiple junction during its transient state in this context, the triple junction presents stationary dihedral angles relying on the energies of the grain boundaries meeting at the junction [192]. In the absence of torque terms, i.e. when the energy of each interface is maintained constant, these angles can be computed through Young's equilibrium, which resolves for the dihedral angles ϕ_1, ϕ_2 and ϕ_3 (see Fig. 6.3) the relation:

$$\frac{\sin\phi_1}{\gamma_{23}} = \frac{\sin\phi_2}{\gamma_{13}} = \frac{\sin\phi_3}{\gamma_{21}} \tag{6.9}$$

Similarly to [143], ratios in the range of $r = [0.53, 10]$ have been tested and the obtained equilibrium angles have been compared to the analytical equilibrium state obtained thanks to Eq. 6.9. Fig. 6.4(left) illustrates the evolution of the φ_3 angle for different values of r obtained with the TRM model. These values are compared to the ones obtained in [143] (see Fig. 6.4(right)), where we have found that the TRM model evolves faster to its equilibrium state than the *heterogeneous* FE-LS method for values of $r > 1.67$. Also, the TRM model is able to reproduce more accurately the analytical values of φ_3 for $r < 1.0$. This can also be seen in Fig. 6.5 where the final value of φ_3 is plotted against the grain boundary en-

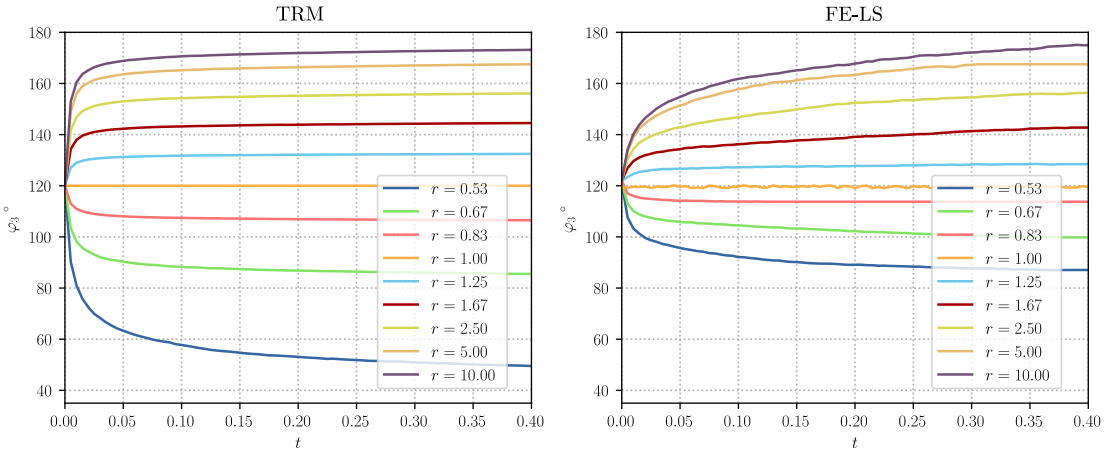


Figure 6.4: Triple junction test case: Evolution of the φ_3 angle for different values of r, (left) TRM and (right) LS-FE models. The plotted data for the LS-FE model was taken from [143].

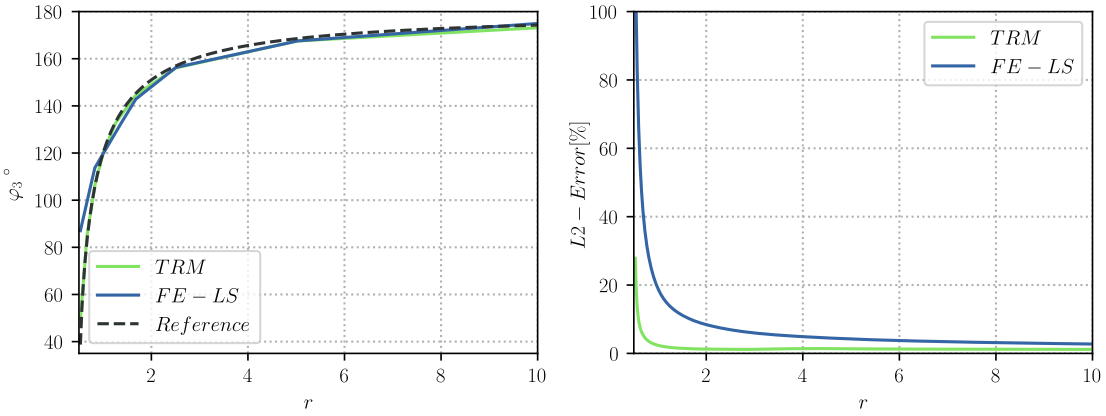


Figure 6.5: Triple junction test case: (left) Final value for the φ_3 angle plotted against the grain boundary energy ratio r and compared to the analytical equilibrium value, and (right) L2-Error. The plotted data for the LS-FE model was taken from [143].

ergy ratio r and compared to the analytical equilibrium value via a L2-Error plot.

Figure 6.6 illustrates the final interface states for both approaches at the end of the simulation. In [143] it was found that, while the equilibrium angles of ϕ_3 were accurately described for values of $r > 2.5$ near the junction, the behavior of the interfaces was strongly affected by the boundary conditions applied to the FE resolution, inducing non-minimal energy configurations. This behavior was not found nor expected with the TRM model as boundary conditions only affect the velocity of nodes belonging to the boundary, and as a result, the TRM model reduces in all cases (until the equilibrium) the total energy of the system. This result can be found in Fig. 6.7 where the evolution of the normalized GB energy

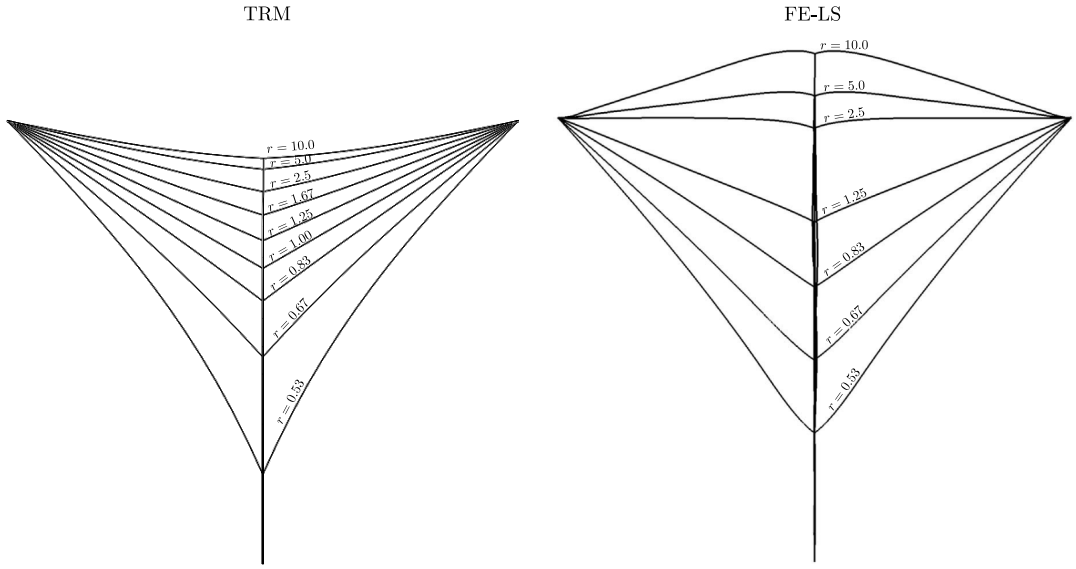


Figure 6.6: Triple junction test case: final interface states for the (left) TRM and (right) LS-FE models. The displayed for the LS-FE model was taken from [143].

$\overline{E}_\Gamma$ (each curve has been escalated to start from a value equals to 1), has been plotted as a function of time.

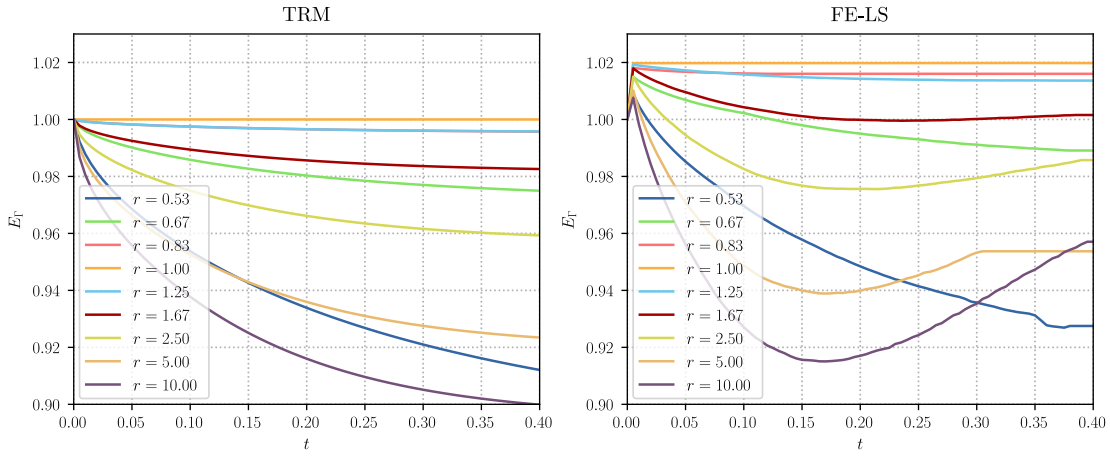


Figure 6.7: Triple junction test case: evolution of the normalized total grain boundary energy E_Γ for the (left) TRM and (right) LS-FE models. The plotted data for the LS-FE model was taken from [143].

6.3.2 2D GG with heterogeneous GB properties: 5000 initial grains

Similarly to the triple junction test case, in this section, we reproduce the testing carried out in [144] in the context of the *heterogeneous* FE-LS formulation

applied to 2D-GG. Therefore, a squared RVE domain of 1.5 mm of side length, containing approximately 5000 initial grains will be used to model annealing. Moreover, even though the statistical data used in [144] for the generation of the initial Laguerre-Voronoi tessellation is given, we were unable to obtain exactly the same subdivision of the domain. Fig. 6.8(top-left) illustrates the initial state of the polycrystal used in all forthcoming simulations. This polycrystal contains exactly 5089 initial grains and its grain size distribution (pondered by surface) is given in Fig. 6.8(top-right). Additionally, in all cases, the mobility term will be held constant and equal to 1.

Of course, before being able to measure a misorientation and grain boundary properties as a function of them, a grain orientation is given to each grain. These orientations are computed at random using a uniform generation random function for each one of the Euler angles (φ_1, Φ, φ_2). Fig. 6.8(bottom-left) shows the initial microstructure colored following the *orientation magnitude* $e = \sqrt{\varphi_1^2 + \Phi^2 + \varphi_2^2}$ of each grain. Disorientation angles (θ) have been computed for each *Line*, *L-Node* and *PP-Connection* (i.e. for all nodes and segments belonging to the GBs) of the interface using the methodology presented in section 6.2.3. Fig. 6.8(bottom-right) gives the disorientation angle distribution obtained for our initial microstructure. The Mackenzie plot [224] for disorientation angles of a cubic sample has been also plotted showing a very good agreement which our statistical initial generation.

The first set of simulations in this section are dedicated to measure the accuracy of the TRM model to reproduce results using different sets of numerical parameters, such as the mesh size and time step (h_{trm}, dt) and in the context of heterogeneous grain boundary properties. A Read-Shockley (RS) type function [64] has been used for this analysis for the determination of the GB energy γ in function of the disorientation angle θ:

$$\gamma = \begin{cases} \gamma_{max} \left(\frac{\theta}{\theta_{max}}\right) \left(1 - \ln\left(\frac{\theta}{\theta_{max}}\right)\right) & \theta < \theta_{max} \\ \gamma_{max} & \theta \geq \theta_{max} \end{cases} \tag{6.10}$$

where γ_{max} is the maximal grain boundary energy equals to 1.012 Jm^{-2} and θ_{max} corresponds to a threshold angle equals to 30°. These values are identical to the ones used in [144], where the authors have explained that contrary to the common usage of the RS function (using values for θ_{max} in the range of [10°, 15°]) the value of $\theta_{max} = 30°$ has been chosen to exaggerate the heterogeneities at the GB properties.

Figure 6.9(top) gives the evolution of various parameters for the first 3 hours of simulated time using a constant mesh size of $h_{trm} = 0.003$ mm and for various time steps dt. The mean grain size, the number of grains and the total GB energy have been plotted, showing a tendency of the data to converge to a fixed solution when the time step decreases. Fig. 6.9(bottom) gives the L2-difference of each iteration taking as a reference the curve using $dt = 5$ s, confirming these results.

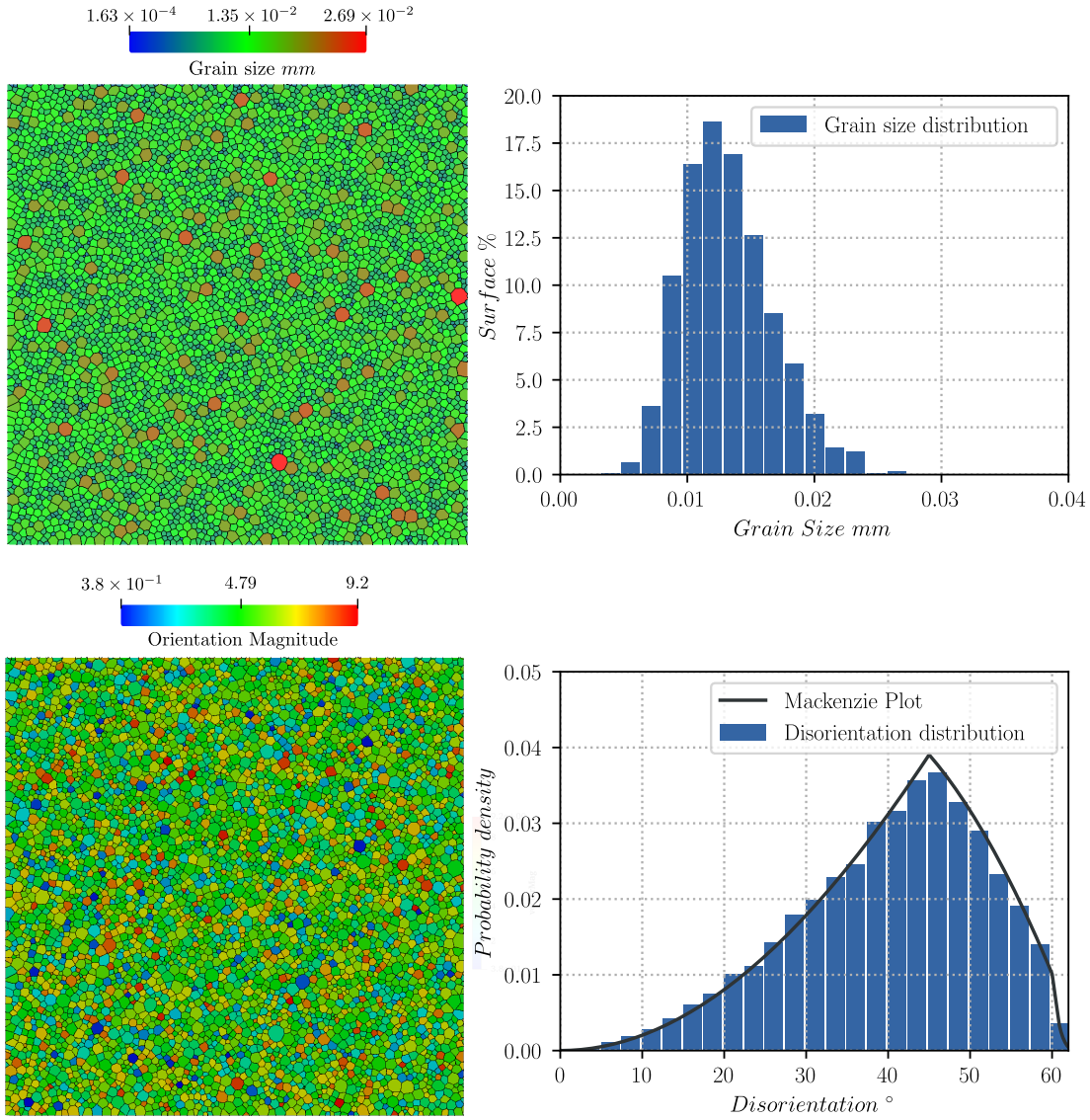


Figure 6.8: Initial state of the 2D heterogeneous GG test case, (top-left) grain size field, (top-right) grain size distribution weighted by surface area, (bottom-left) grain orientation field and (bottom-right) probability density plot of the disorientation angle weighted by length of interface.

Similarly, the simulations were repeated using a constant time step $dt = 50\ s$ and for various mesh sizes, here the tendency of the curves when decreasing the mesh size is also to converge to a given fixed evolution (see Fig. 6.10), reducing the L2-difference to the reference curve (here the one using $h_{trm} = 0.002\ mm$) with every iteration. This study shows that one can expect good accuracy when using a set of parameters (h_{trm}, dt) in the surroundings of ($0.003\ mm$, $50\ s$), hence these values will be used in the subsequent analyses.

Even though we have already given some results for GG under the influence of

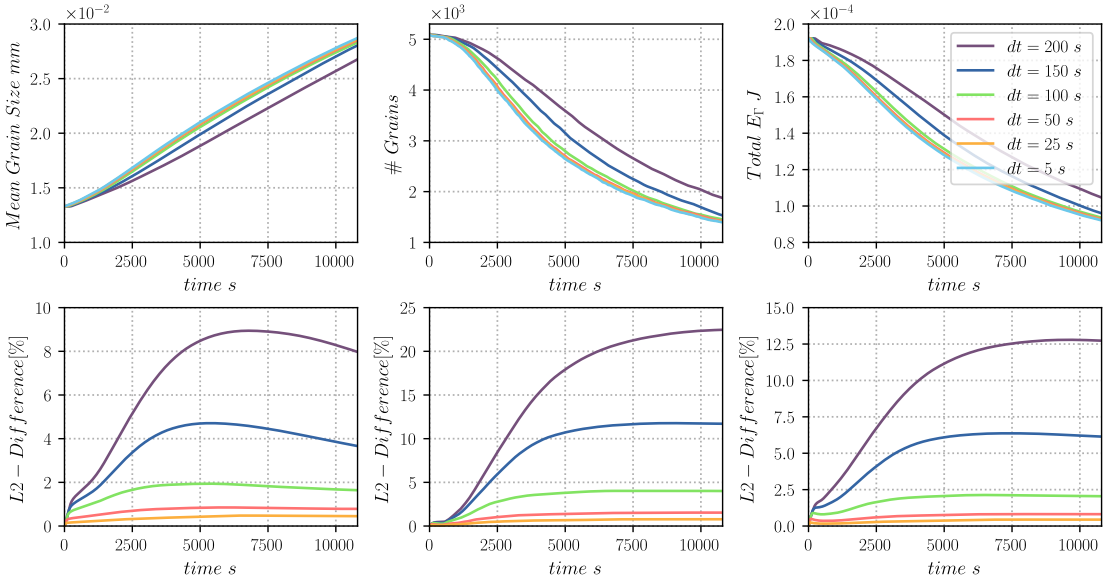


Figure 6.9: Sensitivity to the time step dt for the 2D heterogeneous GG test case using the TRM model and $h_{trm} = 0.003\ mm$. The evolution of different parameters is given when using the RS function for the determination of the GB energy γ in function of the disorientation angle θ: (left) grain Size, (center) number of grains and (right) total GB energy. Each value (top) is compared to the evolution of the smallest dt and the L2-difference to this value is given (bottom).

heterogeneous GB properties, the variation on the heterogeneities using a RS type function is not very high. This is not a surprise as a very low percent of the grain boundaries present a disorientation angle value in the "variational" zone of the RS function (see Fig. 6.8(bottom-right) for the values with a disorientation angle $\theta < 30°$) and the majority of interfaces present a disorientation angle $\theta \geq 30°$ meaning that they will acquire a value of $\gamma = \gamma_{max}$. To give a broader representativity of the *heterogeneous* LS-FE formulation, in [144], multiple functions were used to compute the value of γ as a function of the disorientation angle. In this section, we will test the TRM model using two of the new functions presented in [144] to increase the variability on the GB energy values. As such, the results presented here can be directly compared to those presented in [144]. The functions used correspond to the Read-Shockley+ and the Gaussian functions, being the ones giving the higher heterogeneous configurations, these functions were defined as follows:

RS+

$$\gamma = \begin{cases} \gamma'_{max}\left(\frac{\theta}{\theta_{max}}\right)\left(1 - \ln\left(\frac{\theta}{\theta_{max}}\right)\right) & \theta < \theta_{max} \\ \gamma'_{max} & \theta_{max} \leq \theta \leq \theta_{thresh} \\ 0.1\gamma'_{max} & \theta > \theta_{thresh} \end{cases} \tag{6.11}$$

where $\gamma'_{max} = 1.1\ Jm^{-2}$ and $\theta_{thresh} = 55°$.

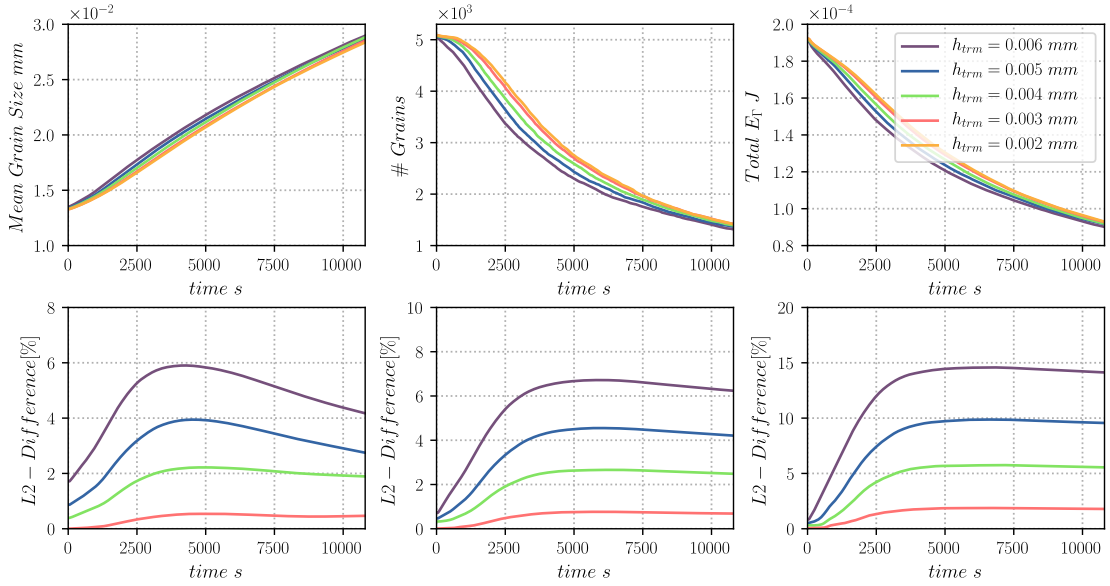


Figure 6.10: Sensitivity to the mesh size h_{trm} for the 2D heterogeneous GG test case using the TRM model and $dt = 50\ s$. The evolution of different parameters is given when using the RS function for the determination of the GB energy γ in function of the disorientation angle θ: (left) grain Size, (center) number of grains and (right) total GB energy. Each value (top) is compared to the evolution of the smallest h_{trm} and the L2-difference to this value is given (bottom).

Gaussian

$$\gamma = \gamma_g \exp \frac{-(\theta - \theta_\mu)^2}{2\theta_\sigma^2} \tag{6.12}$$

where $\gamma'_q = 1.54\ Jm^{-2}$, $\theta_\mu = 40°$ and $\theta_\sigma = 10°$.

These formulations were used along with the RS function and a homogeneous formulation ($\gamma = 1.012\ Jm^{-2}$) in the TRM model for the full field modeling of annealing, using the initial configuration given in Fig. 6.8. Fig. 6.11 shows the initial and the final states of the GB network where the color code is representative of the GB energy of each interface. Here, it is clear how the case using a RS formulation is too "homogeneous", presenting just a few variations in the GB properties (even at the end of the simulation), while the RS+ and Gaussian are more heterogeneous. Additionally, in the RS+ and Gaussian cases, interfaces with a high GB energy seem to be eliminated during the early stages of the simulations, giving a higher predominance to low-energy GBs, which is clearly not the case for the RS configuration. This discussion will be supported below in a more quantitative manner. Finally, another important aspect seen in the final states of the RS+ and Gaussian cases is the appearance of stable high-order multiple junctions predicted in section 6.2.2.

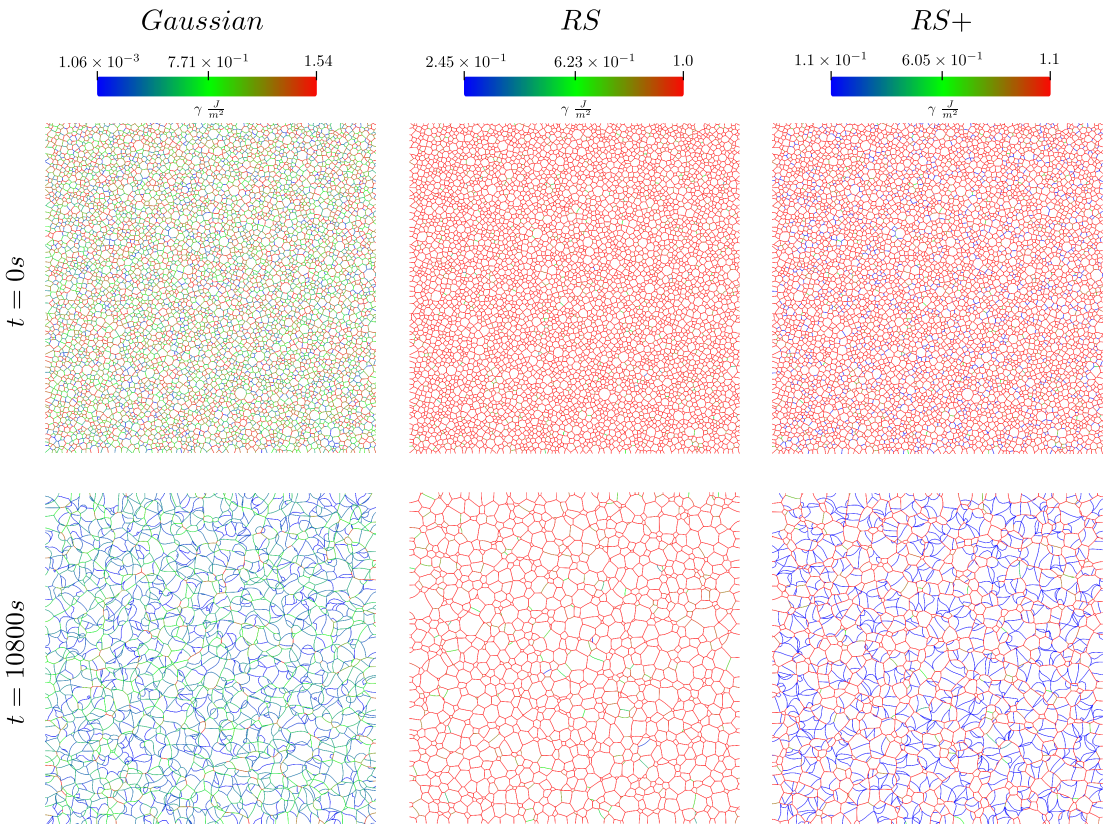


Figure 6.11: States of the microstructure (top) at the beginning of the simulation and (bottom) at the end of the simulation, for the cases from right to left: Gaussian, RS and RS+. A much higher heterogeneity is found in the cases using the Gaussian and RS+ functions, the presence of stable high order multiple junctions can also be remarked for these configurations.

Figure 6.12 illustrates the normalized GB disorientation distributions for the heterogeneous test cases given for the initial state and for every hour of annealing. These data are also compared to the Mackenzie plot. Results show how the RS does not present an evolution of this property but maintains its tendency to approach the Mackenzie plot, hence not giving almost any preference to low energy GBs. Contrarily, the Gaussian and RS+ cases tend to avoid the disorientation angles that produce a high energy GBs, having maximum values at the disorientation angles producing low energy GBs (e.g for the RS+ case a maximum have been found for values of disorientation $\theta > \theta_{thresh} = 55°$). These results can also be observed in terms of the normalized grain boundary energy distributions given in Fig. 6.13, the Gaussian and RS+ cases tend to make disappear high energy GBs, giving a much higher predominance to low energy GBs in only the first hour of treatment and continue to promote their permanency (or their appearance) as times advances, while for the RS case, the changes on the distribution of energy remain negligible.

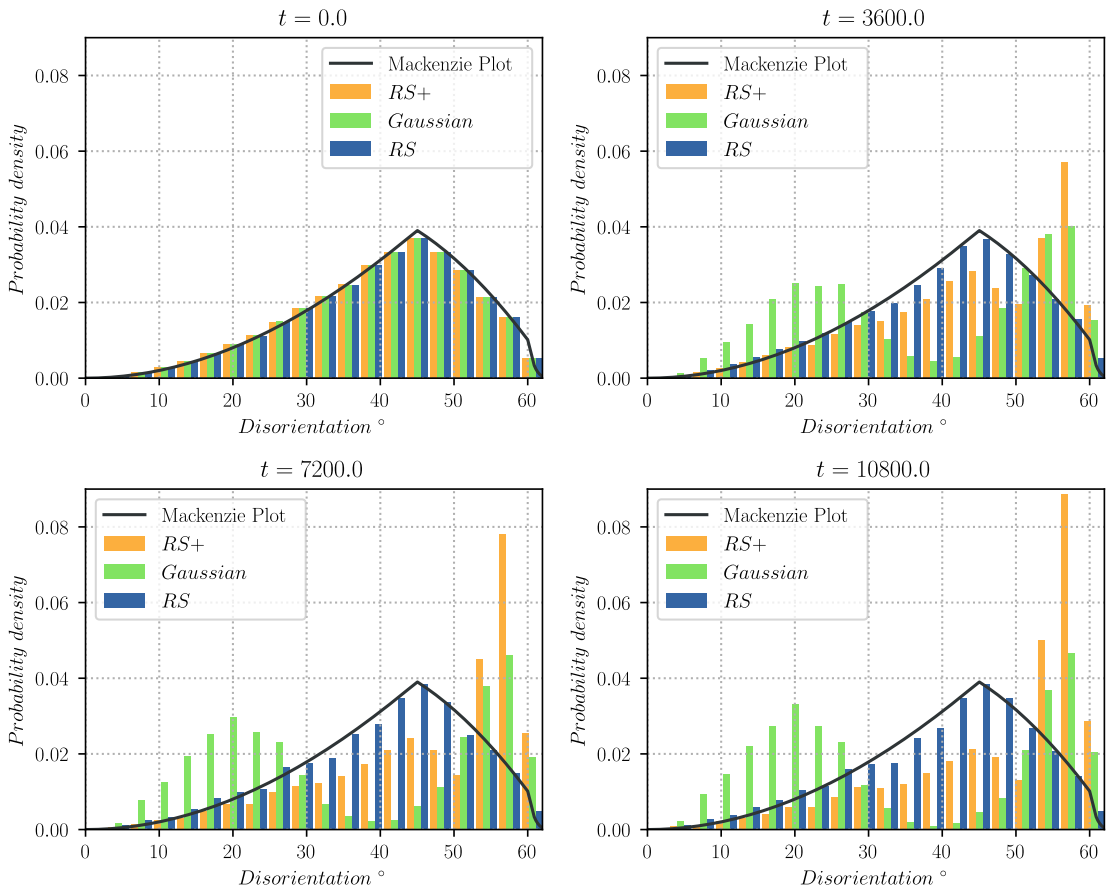


Figure 6.12: Probability density plot of the disorientation angle weighted by GB length, for the cases using the Gaussian, RS and RS+ functions for the computation of the GB energy γ, and for different times.

Low energy GBs predominance may produce a deceleration of the evolution of the grain size in the domain. Fig. 6.14 illustrates the grain size distribution of the different test cases showing how the RS configuration produces a grain size distribution with higher sizes while the RS+ and Gaussian cases promote smaller grains.

Fig. 6.15 gives the evolution of the mean grain size, the total number of grains, the total GB energy and CPU-time of each simulation. Here, it can be seen how the reduction of the GB energy is much higher for the more heterogeneous cases, even though their number of grains and mean size appears to have a "slower" evolution than for the RS and homogeneous cases. It can also be seen that the responses of the homogeneous and the RS cases are very similar, while the CPU-time is much higher in the RS case and in general, the higher the heterogeneity of the case, the higher is its computational cost. This behavior can be anticipated by seeing the evolution of the number of grains, as the more homogeneous cases reduce much faster this quantity. Fig. 6.16(a) gives the CPU-time per increment

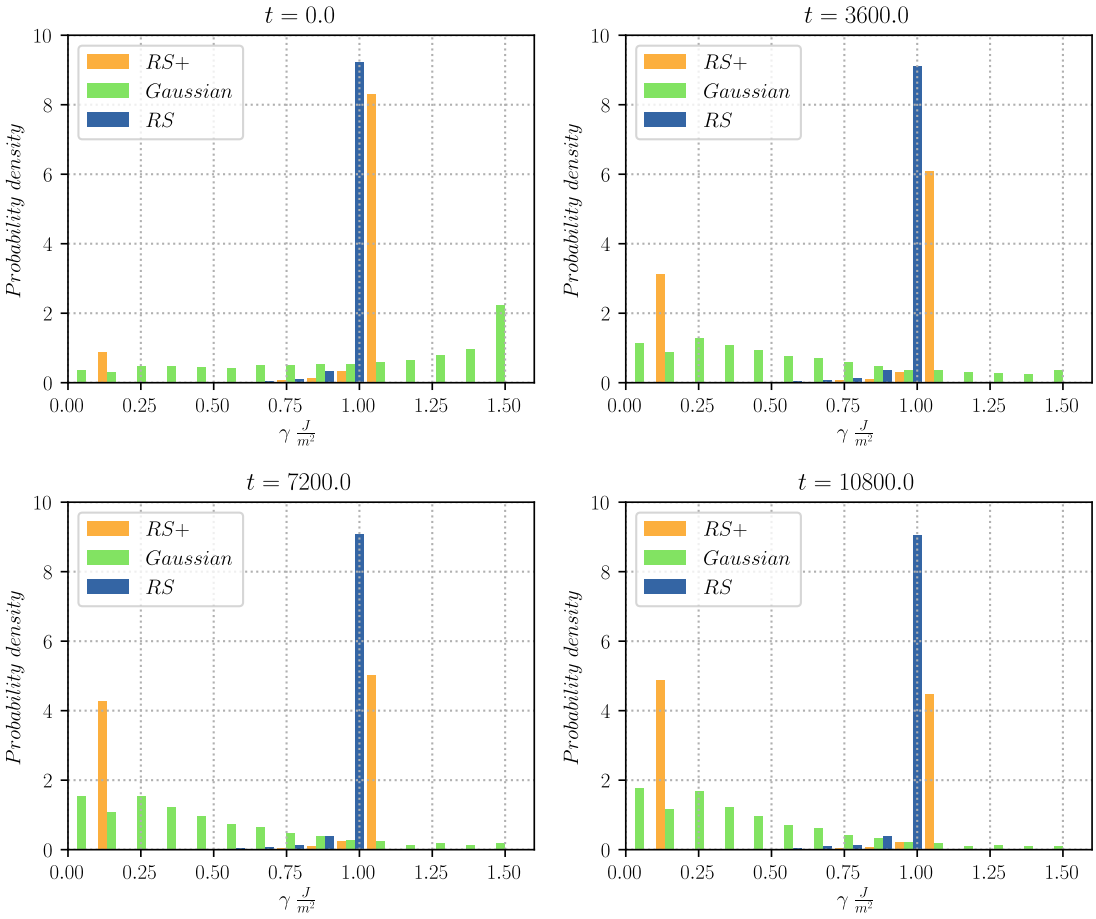


Figure 6.13: Normalized grain boundary energy distribution using various functions for the computation of the GB energy γ. The distributions are given for every hour of thermal treatment.

plotted against the number of grains of the simulation for each case, the results show that for the same number of grains, the Gaussian and RS+ cases are still more time-consuming. One of the reasons for this result is given by the fact that the total length of grain boundaries is not reduced equally for all the configurations. Fig. 6.16(b) gives the total GB length plotted against the number of grains of the simulation showing how the Gaussian and RS+ cases have a much higher total GB length per number of grains than the RS and the homogeneous cases. This result, in turn, can not be anticipated as one could actually have guessed the contrary, by seeing the curves of evolution of the total GB energy given in Fig. 6.15(bottom-left) in function of time and as illustrated in Fig. 6.15(c) in function of the number of grains. Once again, this result is a product of the preference of the higher heterogeneous cases for grain boundaries of low energy, but also by the more frequent apparition of high-order multiple junctions, that decelerate the reduction of the total GBs length.

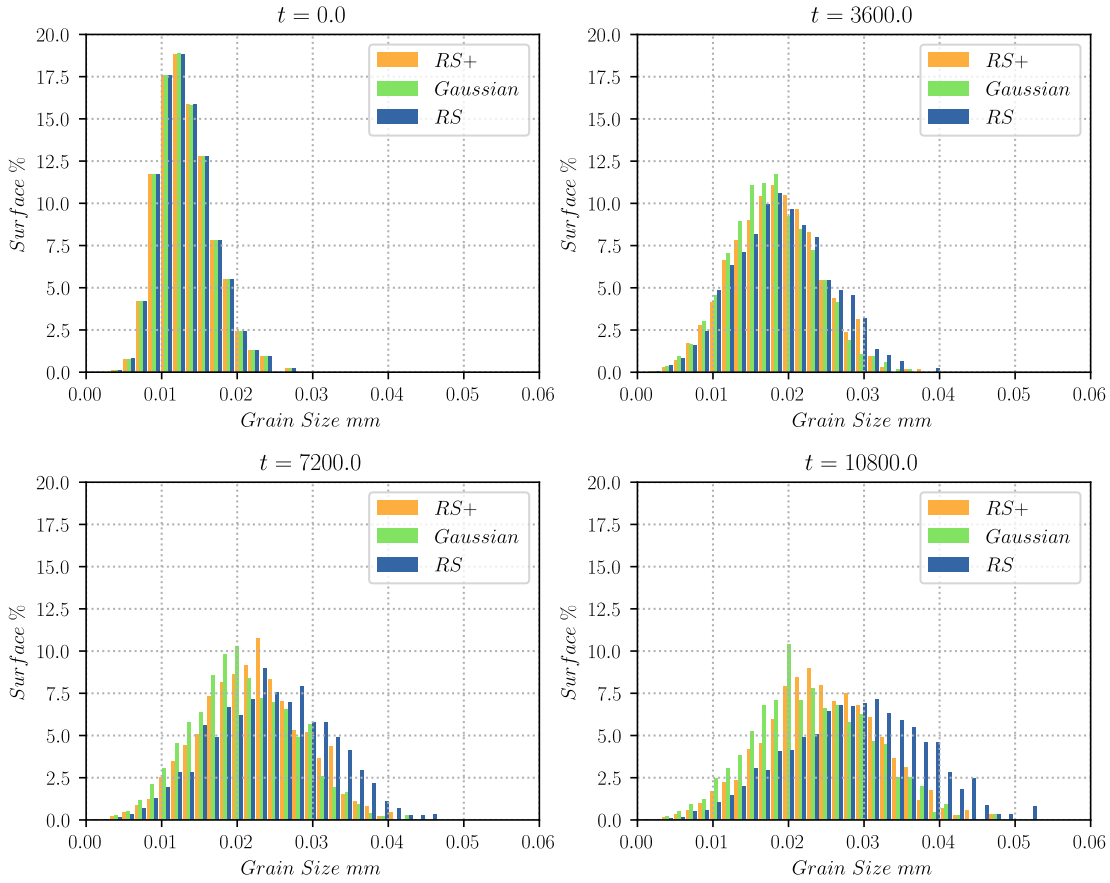


Figure 6.14: Grain size distributions weighted by surface using various functions for the computation of the GB energy γ. The distributions are given for every hour of thermal treatment.

Figure 6.15(d) gives the relation between the CPU-time per increment and the total GB length, showing how the computational cost of the TRM model in this context can be more related to the length of boundaries than to the number of grains. Additionally, it is noticeable that the difference between the computational cost of the homogeneous and the heterogeneous cases. This can be explained by the fact that for the homogeneous case, it is not necessary to compute the misorientation at GBs nor the lowest energy configuration in the event of a separation of MJs. These operations are very demanding as both rely on an iterative computation of the lowest rotation angle between two orientations in a set of 24 possible rotations, where all of them have to be tested.

The results presented in this section regarding the evolution of the microstructural properties are very similar to the ones obtained in [144] in the context of the *heterogeneous* FE-LS formulation. This suggests that both methodologies are valid in order to predict microstructural states in a full field context, as even though the mechanisms behind their evolution are the same, they have been modeled using a completely different numerical scheme, still producing a very similar

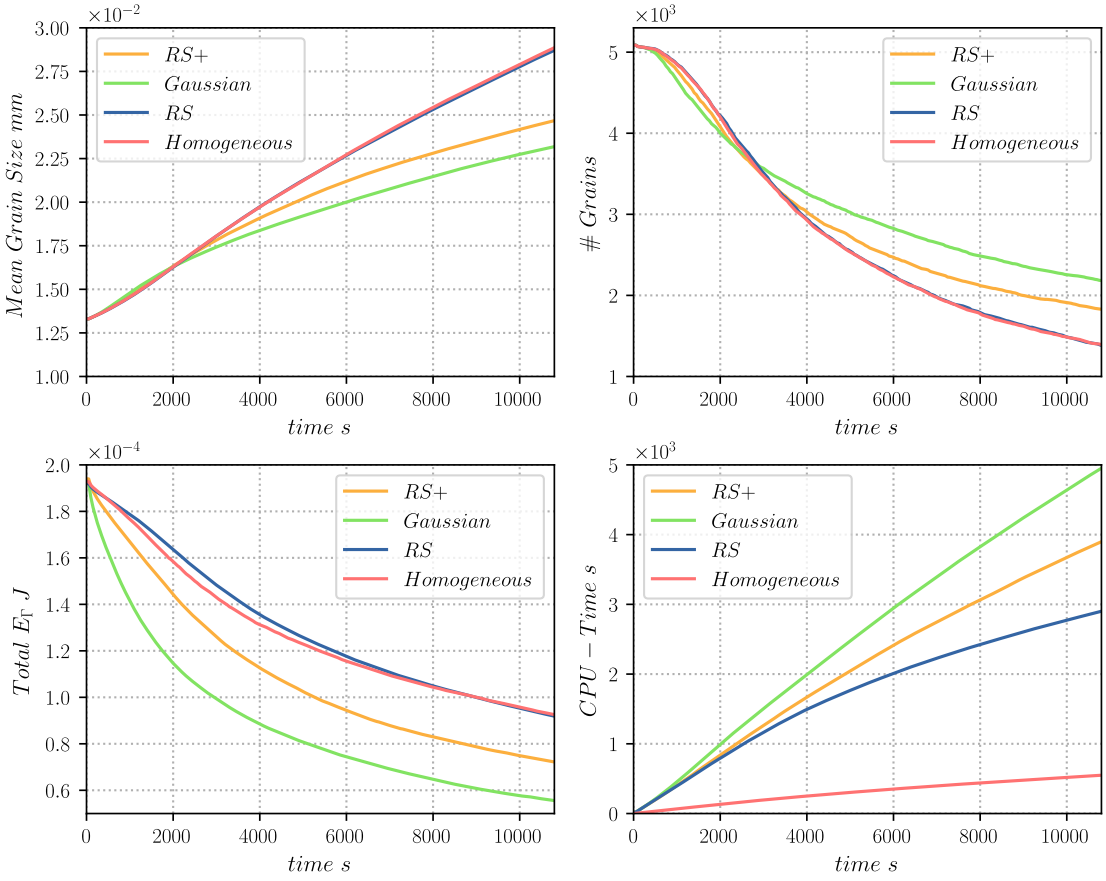


Figure 6.15: Evolution of different parameters as a function of time, for the 2D heterogeneous GG test case simulated with the TRM model: (top-left) mean grain size pondered by surface, (top-right) number of grains, (bottom-left) total grain boundary energy E_Γ and (bottom-right) CPU-time of the simulation

outcome. Moreover, the computational power needed to produce these results using the *heterogeneous* FE-LS may be much higher than the one needed by the TRM. The TRM model was able to perform all simulations in less than 1 hour and 24 minutes in a sequential context an Intel® Core™ i7-7600U processor.

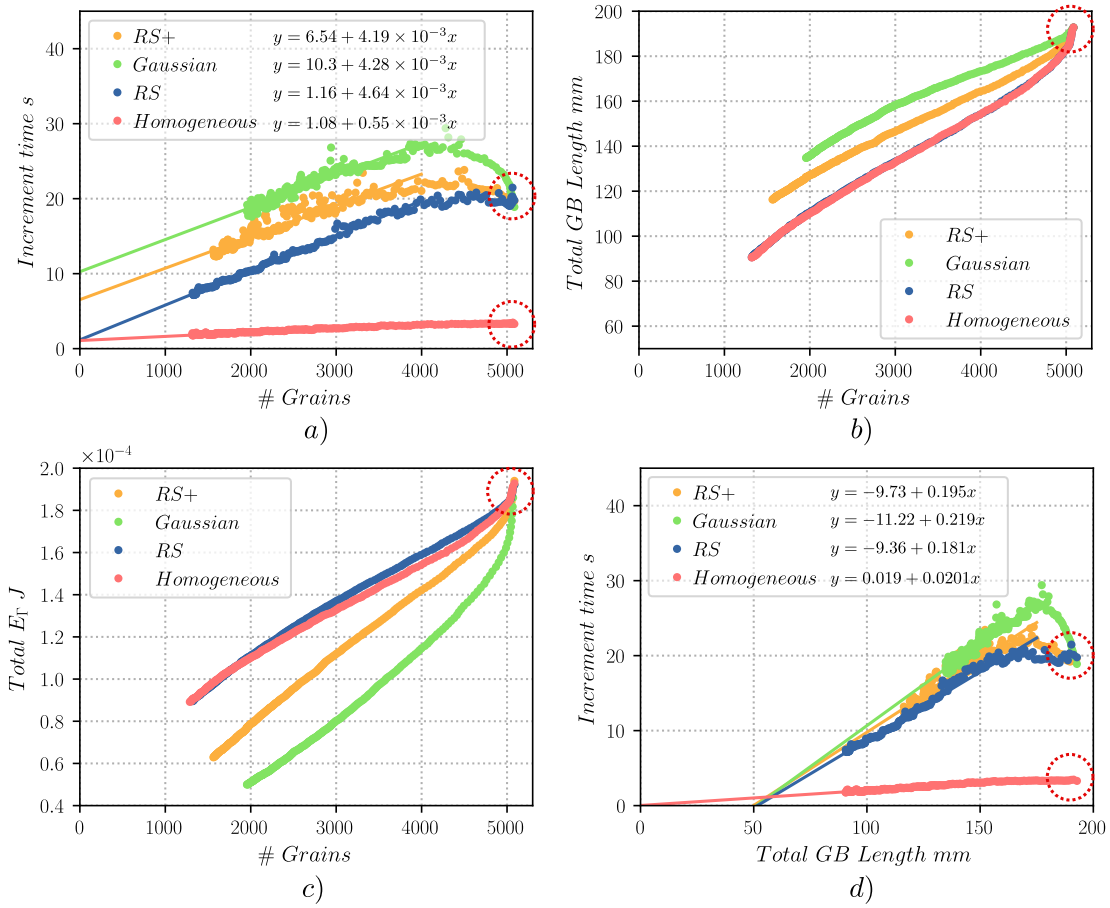


Figure 6.16: Computational cost of the 2D heterogeneous GG test case as a function of a) the number of grains and c) the total GBs length. Additionally, b) gives the total length of GBs as a function of the number of grains. The red circles in every frame give the beginning of the simulation. The heterogeneous cases present a much higher computational cost than the homogeneous case as a consequence of the computation of the misorientation and disorientation angles at the interfaces.

6.4 Discussion, conclusion and perspectives

In this chapter, we have provided the necessary implementation for the modeling of grain growth using heterogeneous grain boundary properties with the TRM model. These implementations consist in the development of a numerical framework developed on top of the TRM base code, able to measure the misorientation between neighbors grains. These measurements have been developed in order to only be calculated at grain interfaces, namely, *L-Nodes* and *PP-Connections*. Additionally, the implementation of a new separation algorithm for high-order multiple junctions was necessary. The new algorithm iterates between all possible decompositions conformed by the subsequent separation of pairs of interfaces from the multiple junction, storing for all iterations the total GB energy change ΔE_Γ and selecting/applying the one with the lowest ΔE_Γ only if its value is lower than zero (as for events with a minimum value of $\Delta E_\Gamma > 0$ the original configuration should remain stable).

The new methodology implemented for the TRM model was tested in the context of heterogeneous grain boundary properties, using identical test cases like the ones presented in [143, 144]. The results show that the TRM model is able to produce more accurate results regarding the equilibrium angles of triple junctions when compared to the analytical values given by Young's equilibrium. Additionally, the TRM model ensures at all times low-energy stable configurations contrary to the *heterogeneous* LS-FE model presented in [143] which may produce stable configurations with non-minimal energy states. Furthermore, the TRM model was tested in a GG context using heterogeneous grain boundary properties in function of the disorientation angle. The initial configuration of all tests was statistically identical to the one presented in [144] with 5089 initial grains. Sensitivity analyses were performed, resulting in a tendency of the model to converge to a fixed solution when decreasing the set of parameters (h_{trm}, dt) controlling the mesh size and the time step respectively. Then, multiple formulations for the determination of the grain boundary energy γ in function of the disorientation angle θ were used, namely the Read-Shockley (RS) [64], the modified Read-Shockley (RS+) and the Gaussian formulations used in [144]. The results obtained in these tests showed a very similar statistical and mean behavior with the results presented in [144] for the same formulations of $\gamma(\theta)$ but used in a LS-FE context, hence validating both approaches at this scale.

A quick study on the computational performance of the TRM model was also performed for the last set of simulations. The results show that the CPU-time per increment is proportional to the total length of GBs. Additionally, the computational needs of the heterogeneous cases are higher than when using a homogeneous configuration, as the computation of the disorientation angles, even if it is only performed at the interfaces, remains very costly, taking up to the 60% of the total CPU-time.

Finally, further analyses need to be performed with the TRM model involving a much higher number of initial grains, in order to measure the impact of the size of the domain on the statistical behavior of grains under the influence of heterogeneous grain boundary properties. This remains a perspective of the present work, along with the application of the TRM model in a fully anisotropic context where the influence of the inclination of the interface on the computation of γ would also be taken into account.

Résumé en Français du Chapitre 6

Ce chapitre a été consacré aux développements du schéma numérique permettant au modèle TRM la prise en compte des anisotropies/hétérogénéités sur la définition des propriétés physiques des joints de grain.

Une nouvelle formulation pour le terme de vitesse issue de la capillarité a été proposée. Ce terme de vitesse permet d'introduire les termes torque présents sur les arêtes des interfaces lors d'une variation dans l'énergie de surface sur la longueur d'un joint de grains.

D'un autre côté, des développements concernant un algorithme de séparation des jonctions multiples d'ordre élevé (points quadruples, quintuples...) ont été introduits. Cet algorithme est basé sur la minimisation locale de l'énergie de surface dans les voisinages des jonctions multiples en testant plusieurs décompositions possibles et en choisissant celle avec l'énergie totale la plus faible.

Le modèle a été testé sur un environnement *hétérogène*: l'énergie des interfaces varie uniquement en fonction de l'angle de désorientation des joints de grain. Étant donné que cette désorientation est constante sur le long d'un joint de grain, les termes torque sont nuls. Cette considération a permis d'effectuer un cas test quantifiant l'erreur sur les angles à l'équilibre (équilibre de Young), obtenues aux jonctions multiples. La méthode TRM s'avère très précise.

Un deuxième cas test hétérogène a été effectué, reproduisant les conditions initiales (même distribution de taille de grains et même nombre de grains), utilisées pour la caractérisation d'un modèle LS-EF publié dans un article récent. Les résultats du modèle TRM sous ces conditions montrent un bon accord avec ceux obtenus avec le modèle hétérogène LS-EF.

Finalement, la performance numérique du modèle TRM sous conditions hétérogènes a été comparé à celle avec des conditions homogènes. Cette comparaison montre une augmentation significative des ressources nécessaires pour des conditions hétérogènes (un temps de calcul global 5 fois plus important). Cela s'explique, en partie, par le calcul de la désorientation entre grains voisins.

La méthode TRM est ainsi validée sur un contexte hétérogène, elle est en cours de test sur un environnement complètement anisotrope (dépendance à l'angle de désorientation et à l'inclinaison) et les résultats seront présentés dans une future publication.

Perspectives

Even though we have provided an accurate and efficient framework for the modeling of microstructural evolutions in a 2D context, the work regarding the TRM model is far from being finished. Indeed we have covered aspects such as grain coarsening during annealing, ReX mechanisms, and the influence of heterogeneous grain boundary properties during GG, all of them in a 2D context.

Other microstructural mechanisms involve the application of the TRM model in a fully anisotropic (5 parameters dependence) context, as an extension of the works presented in chapter 6. These aspects are intended to be studied in the subsequent PhD work of [225]. Another perspective of the present work is the development of the TRM approach for the modeling of the Smith-Zener pinning phenomenon by taking into account static or evolving SPPs, this study is currently being developed in the context of the FE-LS method in [226] and is expected to be developed also in the context of the TRM model in future works.

A great interest of the present work is the ability of the TRM model to simulate all these mechanisms using a very high number of grains over small CPU-times. This aspect was illustrated in chapter 4 of this work, where the TRM model was used to simulate annealing for a test case made of 545000 initial grains, to the point where only 1/5 of the initial grains remained (1 hour of thermal treatment) under 40 minutes of CPU-time, in a cluster station with 140 processors. These works can be extended to study the influence of the size and the initial subdivision of the domain on the reliability of the obtained results. i.e, the TRM model can help answer the question of, for simulations using the same initial grain size distribution, how big must a full field simulation be, to provide accurate statistical and morphological predictions regardless of the initial subdivision of the domain (i.e. the algorithm or random generator used to create the initial microstructural state)?. This question has been formally expressed in the recent PhD works of [23], and it may have different answers in function of the considered microstructural attributes or mechanisms being studied (GG, the anisotropy of GBs properties, ReX...). Other questions regarding this aspect may be risen, such as, is it the same to simulate 100 different microstructures all with the same initial grain boundary distribution than 1 simulation 100 times bigger? or the analogous question, now in the context of heterogeneous GBs, regarding the initial orientations of grains.

Moreover, another high perspective of these works is the extension of the TRM method to a 3D context, which represents a great challenge but not an impossible one. Many of the algorithms implemented and presented in this PhD work have been developed to be used in a 3D context. Additionally, the fact that the model has been built upon unstructured meshes based on simplexes (which need to maintain a correct geometrical description of the domain they represent) is an advantage of the TRM model over other Front-Tracking methods regarding their 3D implementation, as all topological changes of the microstructure can be reduced to simple local remeshing operations performed at the boundaries between grains.

Similarly, the use of unstructured FE meshes carries the benefit of using the FE method for the resolution of more complex formulations. For instance, the TRM model produces meshes being directly applicable to the context of crystal plasticity, a subject of great interest studied in the PhD works of [227], that could refine the stored energy data used during ReX and give a more realistic displacement field of an RVE subjected to a thermomechanical loading (see chapter 5 where plane deformation hypotheses were used).

Indeed, refining the stored energy field description during full-field simulations can lead to the study of the substructures (subgrains) and to propose more precise nucleation models.

Moreover, until now, full-field models have been limited to the study of RVEs that, even though they contain a high number of grains, they are usually representative of very small portions of the material. The works presented in this thesis can make it to take full-field simulations to another scale, where 10^7 or 10^8 grains would be considered, hence being able to predict the microstructural behavior of whole (small) macroscopic parts, and thus allowing the study of aspects such as the influence of a temperature gradient and its evolution over time, or the use of the real deformation field of components being formed. Such simulations are not possible today and multiscale computations are the norm. Theoretically, the TRM model could perform simulations at the scale of a small part in a matter of days (around 6 days in a cluster station with 140 cores for a GG simulation involving 10^8 initial grains and 1 hour of TMT). Such computations will also enable to discuss the limits of RVE computations.

Furthermore, there is a high interest in the application of a deep learning (DL) (multilayered neural network) protocol to enrich the remeshing strategy employed by the TRM model. In fact, in chapter 2 the star-connection algorithm is introduced as a tool for metric-based remeshing. Then in chapter 3, we defined the remeshing procedure of the TRM model, based on the use of remeshing operators (edge-splitting, node-collapse...), and we mentioned that our decision was

based on the fact that we wanted to maintain a complete control over the remeshing procedure (which may not be the case when the star-connection algorithm is preferred), as from it depends the topological changes occurring in the grain boundary network. However, this decision may decrease the ability of the TRM model to generate metric-based unstructured meshes. Two potential candidates may solve this issue: i. the use of the star-connection algorithm in the bulk of the grains, blocking all operations modifying the segments of grain boundaries (Lines), and ii. the use of a reconnection algorithm (such as the star-connection algorithm) as a generator for a database containing a set of the **best possible reconnection** (result) of a set of **element patches** (entry); the idea would be to use this database in a DL environment, able to access/interpolate a result in, theoretically, a fraction of the time that would employ the original reconnection algorithm[3] to generate the same (or a very similar) result. In this scenario, the database can also contain data regarding the position of the internal node (see Fig. 2.1.h) that would maximize the element quality of a given patch in a given metric space. The development of such an environment will be the subject of my postdoctoral research in 2021.

Finally, Even though all applications of the TRM model in this manuscript have been performed in the context of microstructural evolutions, the TRM model can be applied to other solid-state phenomena in the context of multidomain simulations, using other boundary motion mechanisms and another definition of the boundary properties.

[3]Reconnection algorithms perform iteratively the same operations even if the input data (element patch and quality criteria) is the same (or very similar) to already treated patches. This could make the mesh adaptation process very costly in terms of CPU-time

Bibliography

[1] A. M. Turing. On computable numbers with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937.

[2] A. M. Turing. On computable numbers with an application to the Entscheidungsproblem. A Correction. *Proceedings of the London Mathematical Society*, 42(2):544–546, 1937.

[3] A. Church. An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics*, 58(2):345, 1936.

[4] D. Hilbert. *Grundzüge Der Theoretischen Logik*. Berlin, G. Springer, 1928.

[5] M. Bernacki, Y. Chastel, T. Coupez, and R. E. Logé. Level set framework for the numerical modelling of primary recrystallization in polycrystalline materials. *Scripta Materialia*, 58(12):1129–1132, 2008.

[6] B. Scholtes, R. Boulais-Sinou, A. Settefrati, D. Pino Muñoz, I. Poitrault, A. Montouchet, N. Bozzolo, and M. Bernacki. 3D level set modeling of static recrystallization considering stored energy fields. *Computational Materials Science*, 122:57–71, 2016.

[7] L. Maire, B. Scholtes, C. Moussa, N. Bozzolo, D. P. Muñoz, A. Settefrati, and M. Bernacki. Modeling of dynamic and post-dynamic recrystallization by coupling a full field approach to phenomenological laws. *Materials and Design*, 133:498–519, 2017.

[8] B. Scholtes. *Development of an efficient level set framework for the full field modeling of recrystallization in 3D.* PhD thesis, MINES ParisTech, 2016.

[9] K. A. Brakke. The surface evolver. *Experimental Mathematics*, 1(2):141–165, 1992.

[10] G. Couturier, C. Maurice, and R. Fortunier. Three-dimensional finite-element simulation of Zener pinning dynamics. *Philosophical Magazine*, 83(30):3387–3405, 2003.

[11] G. Couturier, C. Maurice, R. Fortunier, R. Doherty, and J. H. Driver. Finite element simulations of 3D Zener pinning. In *Materials Science Forum*, volume 467-470, pages 1009–1018, 2004.

[12] G. Couturier, R. Doherty, C. Maurice, and R. Fortunier. 3D finite element simulation of the inhibition of normal grain growth by particles. *Acta Materialia*, 53(4):977–989, 2005.

[13] J. K. Becker, P. D. Bons, and M. W. Jessell. A new front-tracking method to model anisotropic grain and phase boundary motion in rocks. *Computers and Geosciences*, 34(3):201–212, 2008.

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

[15] D. Weygand, Y. Bréchet, and J. Lépinoux. A vertex dynamics simulation of grain growth in two dimensions. *Philosophical Magazine B*, 78(4):329–352, 1998.

[16] J. Lépinoux, D. Weygand, and M. Verdier. Modeling grain growth and related phenomena with vertex dynamics. *Comptes Rendus Physique*, 11(3-4):265–273, 2010.

[17] L. A. Barrales Mora. 2D vertex modeling for the simulation of grain growth and related phenomena. *Mathematics and Computers in Simulation*, 80(7):1411–1427, 2010.

[18] Y. Mellbin, H. Hallberg, and M. Ristinmaa. A combined crystal plasticity and graph-based vertex model of dynamic recrystallization at large deformations. *Modelling and Simulation in Materials Science and Engineering*, 23(4), 2015.

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

[20] S. Florez, K. Alvarado, D. P. Muñoz, and M. Bernacki. A novel highly efficient Lagrangian model for massively multidomain simulation applied to microstructural evolutions. *Computer Methods in Applied Mechanics and Engineering*, 367:113107, 2020.

[21] S. Florez, J. Fausty, K. Alvarado, B. Murgas, and M. Bernacki. A novel highly efficient lagrangian model for massively multidomain simulations: parallel context. *Under review at Modelling and Simulation in Materials Science and Engineering*, preprint at arXiv:2009.04424, 2020.

[22] S. Florez, K. Alvarado, and M. Bernacki. A new front-tracking lagrangian model for the modeling of dynamic and post-dynamic recrystallization. *Under review at Modelling and Simulation in Materials Science and Engineering*, preprint at arXiv:2009.08368, 2020.

[23] J. Fausty. *Towards the full field modeling and simulation of annealing twins using a Finite Element Level Set method.* PhD thesis, PSL, MinesParisTech, 2020.

[24] D. Schwarzenbach. The success story of crystallography. *Acta Crystallographica Section A*, 68(1):57–67, 2012.

[25] J. Humphreys, G. S. Rohrer, and A. Rollett. *Recrystallization and Related Annealing Phenomena (Third Edition).* Elsevier, 2017.

[26] T. Sutoki. On the mechanism of crystal growth by annealing. *Scientific Reports of Tohoku. Imperial University*, 17(2):857–876, 1928.

[27] M. Alam, M. Blackman, and D. W. Pashley. High-angle kikuchi patterns. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 221(1145):224–242, 1954.

[28] C. M. Hefferan, J. Lind, S. F. Li, U. Lienert, A. D. Rollett, and R. M. Suter. Observation of recovery and recrystallization in high-purity aluminum measured with forward modeling analysis of high-energy diffraction microscopy. *Acta Materialia*, 60(10):4311–4318, 2012.

[29] R. Pokharel, J. Lind, S. F. Li, P. Kenesei, R. A. Lebensohn, R. M. Suter, and A. D. Rollett. In-situ observation of bulk 3D grain evolution during plastic deformation in polycrystalline Cu. *International Journal of Plasticity*, 67:217–234, 2015.

[30] M. Syha, W. Rheinheimer, M. Bäurer, E. M. Lauridsen, W. Ludwig, D. Weygand, and P. Gumbsch. Three-dimensional grain structure of sintered bulk strontium titanate from X-ray diffraction contrast tomography. *Scripta Materialia*, 66(1):1–4, 2012.

[31] L. Nervo, A. King, A. Fitzner, W. Ludwig, and M. Preuss. A study of deformation twinning in a titanium alloy by X-ray diffraction contrast tomography. *Acta Materialia*, 105:417–428, 2016.

[32] T. Werz, M. Baumann, U. Wolfram, and C. E. Krill. Particle tracking during Ostwald ripening using time-resolved laboratory X-ray microtomography. *Materials Characterization*, 90:185–195, 2014.

[33] H. Carpenter and E. C.F. Crystal growth and recrystallization in metals. *Journal of the Institute of Metals*, 24:83–131, 1920.

[34] D. Harker and E. R. Parker. Grain shape and grain growth. *Transactions of the American Society for Metals*, 34:156–201, 1945.

[35] J. E. Burke. Some factors affecting the rate of grain growth in metals. *Aime Trans*, 180(February):73–91, 1949.

[36] J. E. Burke and D. Turnbull. Recrystallization and grain growth. *Progress in Metal Physics*, 3(C), 1952.

[37] C. S. Smith. Grains, phases, and interfaces: An introduction of microstructure. *Trans. Metall. Soc. AIME*, 175:15–51, 1948.

[38] J. Evers, P. Klüfers, R. Staudigl, and P. Stallhofer. Czochralski's Creative Mistake: A Milestone on the Way to the Gigabit Era. *Angewandte Chemie - International Edition*, 42(46):5684–5698, 2003.

[39] J. Czochraiski. *Internationale Zeitschrift fur Metallographie*, 6:289, 1914.

[40] G. Kugler and R. Turk. Modeling the dynamic recrystallization under multi-stage hot deformation. *Acta Materialia*, 52(15):4659–4668, 2004.

[41] M. J. Luton and C. M. Sellars. Dynamic recrystallization in nickel and nickel-iron alloys during high temperature deformation. *Acta Metallurgica*, 17(8):1033–1043, 1969.

[42] C. M. Sellars and J. A. Whiteman. Recrystallization and grain growth in hot rolling. *Metal Science*, 13(3-4):187–194, 1979.

[43] C. W. Price. Use of Kolmogorov-Johnson-Mehl-Avrami kinetics in recrystallization of metals and crystallization of metallic glasses. *Acta Metallurgica Et Materialia*, 38(5):727–738, 1990.

[44] S. F. Medina and C. A. Hernandez. Modelling of the dynamic recrystallization of austenite in low alloy and microalloyed steels. *Acta Materialia*, 44(1):165–171, 1996.

[45] M. Militzer, E. B. Hawbolt, and T. R. Meadowcroft. Microstructural model for hot strip rolling of high-strength low-alloy steels. *Metallurgical and Materials Transactions A: Physical Metallurgy and Materials Science*, 31(4):1247–1259, 2000.

[46] S. I. Kim and Y. C. Yoo. Dynamic recrystallization behavior of AISI 304 stainless steel. *Materials Science and Engineering A*, 311(1-2):108–113, 2001.

[47] A. M. Elwazri, P. Wanjara, and S. Yue. Dynamic recrystallization of austenite in microalloyed high carbon steels. *Materials Science and Engineering A*, 339(1-2):209–215, 2003.

[48] N. Nakata and M. Militzer. Modelling of microstructure evolution during hot rolling of a 780 MPa high strength steel. *ISIJ International*, 45(1):82–90, 2005.

[49] D. Liu, F. Fazeli, M. Militzer, and W. J. Poole. A microstructure evolution model for hot rolling of a Mo-TRIP steel. *Metallurgical and Materials Transactions A: Physical Metallurgy and Materials Science*, 38(4):894–909, 2007.

[50] X. Hou, Y. Xu, and D. Wu. Hot deformation behavior of a new-generation 980 MPa grade TRIP steel for automobiles. *Advanced Materials Research*, 261-263:775–779, 2011.

[51] L. Maire. *Full field and mean field modeling of dynamic and post-dynamic recrystallization in 3D – Application to 304L steel*. PhD thesis, PSL, Mines-ParisTech, 2019.

[52] M. P. Anderson, D. J. Srolovitz, G. S. Grest, and P. S. Sahni. Computer simulation of grain growth-I. Kinetics. *Acta Metallurgica*, 32(5):783–791, 1984.

[53] W. Ludwig, A. King, P. Reischig, M. Herbig, E. M. Lauridsen, S. Schmidt, H. Proudhon, S. Forest, P. Cloetens, S. R. du Roscoat, J. Y. Buffière, T. J. Marrow, and H. F. Poulsen. New opportunities for 3D materials science of polycrystalline materials at the micrometre lengthscale by combined use of X-ray diffraction and X-ray imaging. *Materials Science and Engineering: A*, 524(1-2):69–76, 2009.

[54] H. Proudhon, J. Li, P. Reischig, N. Guéninchault, S. Forest, and W. Ludwig. Coupling Diffraction Contrast Tomography with the Finite Element Method. *Advanced Engineering Materials*, 18(6):903–912, 2016.

[55] H. Imai, M. Iri, and K. Murota. Voronoi diagram in the Laguerre geometry and its applications. *SIAM Journal on Computing*, 14(1):93–105, 1985.

[56] K. Hitti, P. Laure, T. Coupez, L. Silva, and M. Bernacki. Precise generation of complex statistical Representative Volume Elements (RVEs) in a finite element context. *Computational Materials Science*, 61:224–238, 2012.

[57] D. N. Ilin and M. Bernacki. Advancing layer algorithm of dense ellipse packing for generating statistically equivalent polygonal structures. *Granular Matter*, 18(3):43, 2016.

[58] F. S. de Sousa, N. Mangiavacchi, L. G. Nonato, A. Castelo, M. F. Tomé, V. G. Ferreira, J. A. Cuminato, and S. McKee. A front-tracking/front-capturing method for the simulation of 3D multi-fluid flows with free surfaces. *Journal of Computational Physics*, 198(2):469–499, 2004.

[59] F. Haessner and S. Hoffman. *Recrystallization of metallic materials edited by Frank Haessner*. Riederer-Verlag, Stuttgart, [2nd edition] edition.

[60] D. Turnbull. Theory of grain boundary migration rates. *Jom*, 3(8):661–665, 1951.

[61] D. J. Srolovitz, M. P. Anderson, G. S. Grest, and P. S. Sahni. Computer simulation of grain growth-III. Influence of a particle dispersion. *Acta Metallurgica*, 32(9):1429–1438, 1984.

[62] D. J. Srolovitz, G. S. Grest, and M. P. Anderson. Computer simulation of grain growth-V. Abnormal grain growth. *Acta Metallurgica*, 33(12):2233–2247, 1985.

[63] G. S. Grest, D. J. Srolovitz, and M. P. Anderson. Computer simulation of grain growth-IV. Anisotropic grain boundary energies. *Acta Metallurgica*, 33(3):509–520, 1985.

[64] W. T. Read and W. Shockley. Dislocation Models of Crystal Grain Boundaries. *Physical Review*, 78(3):275–289, 1950.

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

[66] D. J. Srolovitz, G. S. Grest, M. P. Anderson, and A. D. Rollett. Computer simulation of recrystallization-II. Heterogeneous nucleation and growth. *Acta Metallurgica*, 36(8):2115–2128, 1988.

[67] A. D. Rollett, M. J. Luton, and D. J. Srolovitz. Microstructural simulation of dynamic recrystallization. *Acta Metallurgica Et Materialia*, 40(1):43–55, 1992.

[68] J. Jonas, C. Sellars, and W. M. Tegart. Strength and structure under hot-working conditions. *Metallurgical Reviews*, 14(1):1–24, 1969.

[69] P. Peczak and M. J. Luton. A Monte Carlo study of the influence of dynamic recovery on dynamic recrystallization. *Acta Metallurgica Et Materialia*, 41(1):59–71, 1993.

[70] P. Peczak and M. J. Luton. The effect of nucleation models on dynamic recrystallization i. homogeneous stored energy distribution. *Philosophical Magazine B*, 68(1):115–144, 1993.

[71] P. Peczak. A Monte Carlo study of influence of deformation temperature on dynamic recrystallization. *Acta Metallurgica Et Materialia*, 43(3):1279–1291, 1995.

[72] B. Radhakrishnan and T. Zacharia. Simulation of curvature-driven grain growth by using a modified monte carlo algorithm. *Metallurgical and Materials Transactions A*, 26(1):167–180, 1995.

[73] A. D. Rollett and D. Raabe. A hybrid model for mesoscopic simulation of recrystallization. *Computational Materials Science*, 21(1):69–78, 2001.

[74] J. Von Neumann et al. The general and logical theory of automata. *1951*, pages 1–41, 1951.

[75] S. Wolfram. Theory and applications of cellular automata: Advanced series on complex systems. *Singapore*, 1986.

[76] S. Wolfram. *A new kind of science*, volume 5. Wolfram media Champaign, IL, 2002.

[77] H. W. Hesselbarth and I. R. Göbel. Simulation of recrystallization by cellular automata. *Acta Metallurgica Et Materialia*, 39(9):2135–2143, 1991.

[78] W. Johnson and R. Mehl. Reaction kinetics in processes of nucleation and growth, tech. publ. 1089. *Am. Inst. Min. Eng*, pages 1–27, 1939.

[79] M. Avrami. Kinetics of phase change. i general theory. *The Journal of chemical physics*, 7(12):1103–1112, 1939.

[80] M. Avrami. Transformation-time relations for random distribution of nuclei kinetics of phase change, ii. *The Journal of chemical physics*, 8:212, 1940.

[81] M. Avrami. Granulation, phase change, and microstructure kinetics of phase change. iii. *The Journal of chemical physics*, 9(2):177–184, 1941.

[82] A. N. Kolmogorov. On the statistical theory of the crystallization of metals. *Bull. Acad. Sci. USSR, Math. Ser*, 1(3):355–359, 1937.

[83] C. F. Pezzee and D. C. Dunand. The impingement effect of an inert, immobile second phase on the recrystallization of a matrix. *Acta Metallurgica Et Materialia*, 42(5):1509–1524, 1994.

[84] Y. Liu, T. Baudin, and R. Penelle. Simulation of normal grain growth by cellular automata. *Scripta Materialia*, 34(11):1679–1683, 1996.

[85] D. Raabe. Discrete mesoscale simulation of recrystallization microstructure and texture using a stochastic cellular automation approach. In *Texture and Anisotropy of Polycrystals*, volume 273 of *Materials Science Forum*, pages 169–174. Trans Tech Publications Ltd, 1998.

[86] D. Raabe. Introduction of a scalable three-dimensional cellular automaton with a probabilistic switching rule for the discrete mesoscale simulation of recrystallization phenomena. *Philosophical Magazine*, 79(10):2339–2358, 1999.

[87] K. Kremeyer. Cellular Automata Investigations of Binary Solidification. *Journal of Computational Physics*, 142(1):243–263, 1998.

[88] Y. C. Lin, Y. X. Liu, M. S. Chen, M. H. Huang, X. Ma, and Z. L. Long. Study of static recrystallization behavior in hot deformed Ni-based superalloy using cellular automaton model. *Materials and Design*, 99:107–114, 2016.

[89] H. Hallberg, M. Wallin, and M. Ristinmaa. Simulation of discontinuous dynamic recrystallization in pure Cu using a probabilistic cellular automaton. *Computational Materials Science*, 49(1):25–34, 2010.

[90] L. Sieradzki and L. Madej. A perceptive comparison of the cellular automata and Monte Carlo techniques in application to static recrystallization modeling in polycrystalline materials. *Computational Materials Science*, 67:156–173, 2013.

[91] L. Rauch, L. Madej, P. Spytkowski, and R. Golab. Development of the cellular automata framework dedicated for metallic materials microstructure evolution models. *Archives of Civil and Mechanical Engineering*, 15(1):48–61, 2015.

[92] H. Li, X. Sun, and H. Yang. A three-dimensional cellular automata-crystal plasticity finite element model for predicting the multiscale interaction among heterogeneous deformation, DRX microstructural evolution and mechanical responses in titanium alloys. *International Journal of Plasticity*, 87:154–180, 2016.

[93] L. Madej, M. Sitko, A. Legwand, K. Perzynski, and K. Michalik. Development and evaluation of data transfer protocols in the fully coupled random cellular automata finite element model of dynamic recrystallization. *Journal of Computational Science*, 26:66–77, 2018.

[94] A. Soares, A. Ferro, and M. Fortes. Computer simulation of grain growth in a bidimensional polycrystal. *Scripta Metallurgica*, 19(12):1491–1496, 1985.

[95] D. Weaire and J. P. Kermode. Computer simulation of a two-dimensional soap froth. *Philosophical Magazine B*, 48(3):245–259, 1983.

[96] R. Fullman. Boundary migration during grain growth. *Metal Interfaces*, pages 179–207, 1952.

[97] D. Weygand, Y. Brechett, and J. Lépinoux. On the nucleation of recrystallization by a bulging mechanism: A two-dimensional vertex simulation. *Philosophical Magazine B: Physics of Condensed Matter; Statistical Mechanics, Electronic, Optical and Magnetic Properties*, 80(11):1987–1996, 2000.

[98] D. Weygand, Y. Bréchet, and J. Lépinoux. Zener pinning and grain growth: a two-dimensional vertex computer simulation. *Acta Materialia*, 47(3):961–970, 1999.

[99] D. Weygand, Y. Bréchet, and J. Lépinoux. Influence of a reduced mobility of triple points on grain growth in two dimensions. *Acta Materialia*, 46(18):6559–6564, 1998.

[100] D. Weygand, Y. Bréchet, and J. Lépinoux. Reduced mobility of triple nodes and lines on grain growth in two and three dimensions. *Interface Science*, 7(3):285–295, 1999.

[101] D. Weygand, J. Lépinoux, and W. Gust. Three-dimensional grain growth: A vertex dynamics simulation. *Philosophical Magazine B: Physics of Condensed Matter; Statistical Mechanics, Electronic, Optical and Magnetic Properties*, 79(5):703–716, 1999.

[102] H. J. Frost, C. V. Thompson, C. L. Howe, and J. Whang. A two-dimensional computer simulation of capillarity-driven grain growth: Preliminary results. *Scripta Metallurgica*, 22(1):65–70, 1988.

[103] J. W. Cahn and J. E. Hilliard. Free energy of a nonuniform system. I. Interfacial free energy. *The Journal of Chemical Physics*, 28(2):258–267, 1958.

[104] J. B. Collins and H. Levine. Diffuse interface model of diffusion-limited crystal growth. *Physical Review B*, 31(9):6119–6122, 1985.

[105] J. S. Langer. Models of Pattern Formation in First-Order Phase Transitions. pages 165–186, 1986.

[106] L. Q. Chen. Phase-field models for microstructure evolution. *Annual Review of Materials Science*, 32:113–140, 2002.

[107] J. W. Cahn. On spinodal decomposition. *Acta Metallurgica*, 9(9):795–801, 1961.

[108] S. M. Allen and J. W. Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metallurgica*, 27(6):1085–1095, 1979.

[109] L. Q. Chen and W. Yang. Computer simulation of the domain dynamics of a quenched system with a large number of nonconserved order parameters: The grain-growth kinetics. *Physical Review B*, 50(21):15752–15756, 1994.

[110] I. Steinbach, F. Pezzolla, B. Nestler, M. Seeßelberg, R. Prieler, G. J. Schmitz, and J. L. Rezende. A phase field concept for multiphase systems. *Physica D: Nonlinear Phenomena*, 94(3):135–147, 1996.

[111] I. Steinbach. Phase-field models in materials science. *Modelling and Simulation in Materials Science and Engineering*, 17(7):073001, 2009.

[112] D. Fan and L. Q. Chen. Diffuse-interface description of grain boundary motion. *Philosophical Magazine Letters*, 75(4):187–196, 1997.

[113] D. Fan and L. Q. Chen. Topological evolution during coupled grain growth and Ostwald ripening in volume-conserved 2-D two-phase polycrystals. *Acta Materialia*, 45(10):4145–4154, 1997.

[114] D. Fan, C. Geng, and L.-Q. Chen. Computer simulation of topological evolution in 2-D grain growth using a continuum diffuse-interface field model. *Acta Materialia*, 45(3):1115–1126, 1997.

[115] D. Fan and L.-Q. Chen. Computer simulation of grain growth using a continuum field model. *Acta Materialia*, 45(2):611–622, 1997.

[116] C. E. Krill and L. Q. Chen. Computer simulation of 3-D grain growth using a phase-field model. *Acta Materialia*, 50(12):3057–3073, 2002.

[117] A. Kazaryan, Y. Wang, S. A. Dregia, and B. R. Patton. Generalized phase-field model for computer simulation of grain growth in anisotropic systems. *Physical Review B*, 61(21):14275–14278, 2000.

[118] A. Kazaryan, Y. Wang, S. A. Dregia, and B. R. Patton. Grain growth in systems with anisotropic boundary mobility: Analytical model and computer simulation. *Physical Review B*, 63(18):184102, 2001.

[119] N. Moelans, B. Blanpain, and P. Wollants. Quantitative analysis of grain boundary properties in a generalized phase field model for grain growth in anisotropic systems. *Physical Review B - Condensed Matter and Materials Physics*, 78(2), 2008.

[120] K. Chang, L. Q. Chen, C. E. Krill, and N. Moelans. Effect of strong nonuniformity in grain boundary energy on 3-D grain growth behavior: A phase-field simulation study. *Computational Materials Science*, 127:67–77, 2017.

[121] K. Chang and N. Moelans. Effect of grain boundary energy anisotropy on highly textured grain structures studied by phase-field simulations. *Acta Materialia*, 64:443–454, 2014.

[122] N. Moelans, A. Godfrey, Y. Zhang, and D. Juul Jensen. Phase-field simulation study of the migration of recrystallization boundaries. *Physical Review B - Condensed Matter and Materials Physics*, 88(5):1–10, 2013.

[123] L. Chen, J. Chen, R. A. Lebensohn, Y. Z. Ji, T. W. Heo, S. Bhattacharyya, K. Chang, S. Mathaudhu, Z. K. Liu, and L. Q. Chen. An integrated

fast Fourier transform-based phase-field and crystal plasticity approach to model recrystallization of three dimensional polycrystals. *Computer Methods in Applied Mechanics and Engineering*, 285:829–848, 2015.

[124] J. Tiaden, B. Nestler, H. Diepers, and I. Steinbach. The multiphase-field model with an integrated concept for modelling solute diffusion. *Physica D: Nonlinear Phenomena*, 115(1-2):73–86, 1998.

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

[126] I. Steinbach and M. Apel. Multi phase field model for solid state transformation with elastic strain. *Physica D: Nonlinear Phenomena*, 217(2):153–160, 2006.

[127] J. Eiken, B. Böttger, and I. Steinbach. Multiphase-field approach for multicomponent alloys with extrapolation scheme for numerical application. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 73(6):1–9, 2006.

[128] H. Garcke, B. Nestler, and B. Stoth. On anisotropic order parameter models for multi-phase systems and their sharp interface limits. *Physica D: Nonlinear Phenomena*, 115(1-2):87–108, 1998.

[129] H. Garcke, B. Nestler, and B. Stoth. A Multiphase Field Concept : Numerical Simulations of Moving Phase Boundaries and Multiple Junctions. *Applied Mathematics*, 60(1):295–315, 1999.

[130] E. Miyoshi and T. Takaki. Multi-phase-field study of the effects of anisotropic grain-boundary properties on polycrystalline grain growth. *Journal of Crystal Growth*, 474(November 2016):160–165, 2017.

[131] T. Takaki, T. Hirouchi, Y. Hisakuni, A. Yamanaka, and Y. Tomita. Multi-phase-field model to simulate microstructure evolutions during dynamic recrystallization. *Materials Transactions*, 49(11):2559–2565, 2008.

[132] T. Takaki, Y. Hisakuni, T. Hirouchi, A. Yamanaka, and Y. Tomita. Multi-phase-field simulations for dynamic recrystallization. *Computational Materials Science*, 45(4):881–888, 2009.

[133] T. Takaki, C. Yoshimoto, A. Yamanaka, and Y. Tomita. Multiscale modeling of hot-working with dynamic recrystallization by coupling microstructure evolution and macroscopic mechanical behavior. *International Journal of Plasticity*, 52:105–116, 2014.

[134] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[135] M. Sussman, P. Smereka, and S. Osher. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *Journal of Computational Physics*, 114(1):146–159, 1994.

[136] J. Bruchon, D. Pino-Muñoz, F. Valdivieso, and S. Drapier. Finite Element Simulation of Mass Transport During Sintering of a Granular Packing. Part I. Surface and Lattice Diffusions. *Journal of the American Ceramic Society*, 95(8):2398–2405, aug 2012.

[137] B. Merriman, J. K. Bence, and S. J. Osher. Motion of Multiple Junctions: A Level Set Approach. *Journal of Computational Physics*, 112(2):334–363, 1994.

[138] H. K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127(1):179–195, 1996.

[139] F. Reitich and H. M. Soner. Three-phase boundary motions under constant velocities. I: The vanishing surface tension limit. *Royal Society of Edinburgh - Proceedings A*, 126(4):837–865, 1996.

[140] M. Bernacki, R. E. Logé, and T. Coupez. Level set framework for the finite-element modelling of recrystallization and grain growth in polycrystalline materials. *Scripta Materialia*, 64(6):525–528, 2011.

[141] B. Scholtes, M. Shakoor, N. Bozzolo, P. O. Bouchard, A. Settefrati, and M. Bernacki. Advances in Level-Set modeling of recrystallization at the polycrystal scale - Development of the Digi-μ software. *Key Engineering Materials*, 651-653:617–623, 2015.

[142] H. Hallberg. A modified level set approach to 2D modeling of dynamic recrystallization. *Modelling and Simulation in Materials Science and Engineering*, 21(8):85012, 2013.

[143] J. Fausty, N. Bozzolo, D. Pino Muñoz, and M. Bernacki. A novel level-set finite element formulation for grain growth with heterogeneous grain boundary energies. *Materials & Design*, 160:578–590, 2018.

[144] J. Fausty, N. Bozzolo, and M. Bernacki. A 2D level set finite element grain coarsening study with heterogeneous grain boundary energies. *Applied Mathematical Modelling*, 78:505–518, 2020.

[145] H. Hallberg and V. V. Bulatov. Modeling of grain growth under fully anisotropic grain boundary energy Modeling of grain growth under fully anisotropic grain boundary energy. *Modelling and Simulation in Materials Science and Engineering*, 27(045002), 2019.

[146] B. Scholtes, M. Shakoor, A. Settefrati, P.-O. Bouchard, N. Bozzolo, and M. Bernacki. New finite element developments for the full field modeling of microstructural evolutions using the level-set method. *Computational Materials Science*, 109:388–398, 2015.

[147] A. Agnoli, M. Bernacki, R. Logé, J. M. Franchet, J. Laigo, and N. Bozzolo. Selective Growth of Low Stored Energy Grains During δ Sub-solvus Annealing in the Inconel 718 Nickel-Based Superalloy. *Metallurgical and Materials Transactions A: Physical Metallurgy and Materials Science*, 46(9):4405–4421, 2015.

[148] D. A. Ruiz Sarrazola, D. Pino Muñoz, and M. Bernacki. A new numerical framework for the full field modeling of dynamic recrystallization in a CPFEM context. *Computational Materials Science*, 179(January):109645, 2020.

[149] D. A. Ruiz Sarrazola, L. Maire, C. Moussa, N. Bozzolo, D. Pino Muñoz, and M. Bernacki. Full field modeling of dynamic recrystallization in a CPFEM context – Application to 304L steel. *Computational Materials Science*, 184(May):109892, 2020.

[150] M. Elsey, S. Esedoglu, and P. Smereka. Diffusion generated motion for grain growth in two and three dimensions. *Journal of Computational Physics*, 228(21):8015–8033, 2009.

[151] M. Elsey, S. Esedoglu, and P. Smereka. Large-scale simulation of normal grain growth via diffusion-generated motion. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467(2126):381–401, 2011.

[152] M. Elsey, S. Esedog¯lu, and P. Smereka. Simulations of anisotropic grain growth: Efficient algorithms and misorientation distributions. *Acta Materialia*, 61(6):2033–2043, 2013.

[153] C. Mießen, N. Velinov, G. Gottstein, and L. A. Barrales-Mora. A highly efficient 3D level-set grain growth algorithm tailored for ccNUMA architecture. *Modelling and Simulation in Materials Science and Engineering*, 25(8), 2017.

[154] M. Bernacki, H. Resk, T. Coupez, and R. E. Logé. Finite element model of primary recrystallization in polycrystalline aggregates using a level set framework. *Modelling and Simulation in Materials Science and Engineering*, 17(6):64006, 2009.

[155] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[156] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences*, 95(15):8431–8435, 1998.

[157] J. A. Sethian and A. Vladimirsky. Fast methods for the Eikonal and related Hamilton- Jacobi equations on unstructured meshes. *Proceedings of the National Academy of Sciences*, 97(11):5699–5703, 2000.

[158] M. Shakoor, B. Scholtes, P.-O. Bouchard, and M. Bernacki. An efficient and parallel level set reinitialization method – Application to micromechanics and microstructural evolutions. *Applied Mathematical Modelling*, 39(23-24):7291–7302, 2015.

[159] D. N. Ilin, N. Bozzolo, T. Toulorge, and M. Bernacki. Full field modeling of recrystallization: Effect of intragranular strain gradients on grain boundary shape and kinetics. *Computational Materials Science*, 150(March):149–161, 2018.

[160] H. Resk, L. Delannay, M. Bernacki, T. Coupez, and R. Logé. Adaptive mesh refinement and automatic remeshing in crystal plasticity finite element simulations. *Modelling and Simulation in Materials Science and Engineering*, 17(7):75012, 2009.

[161] M. Shakoor, P.-O. Bouchard, and M. Bernacki. An adaptive level-set method with enhanced volume conservation for simulations in multiphase domains. *International Journal for Numerical Methods in Engineering*, 109(4):555–576, 2017.

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

[163] P. G. Young, T. B. H. Beresford-West, S. R. L. Coward, B. Notarberardino, B. Walker, and A. Abdul-Aziz. An efficient approach to converting three-dimensional image data into highly accurate computational models. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 366(1878):3155–3173, 2008.

[164] Y. Zhang, C. Bajaj, and B.-S. Sohn. 3D Finite Element Meshing from Imaging Data. *Computer methods in applied mechanics and engineering*, 194(48-49):5083–5106, 2005.

[165] A. Rollett, D. Saylor, J. Fridy, B. El-Dasher, A. Brahme, S.-B. Lee, C. Cornwell, and R. Noack. Modeling Polycrystalline Microstructures in 3D. In S. Ghosh, J. Castro, and J. Lee, editors, *American Institute of Physics Conference Series*, volume 712 of *American Institute of Physics Conference Series*, pages 71–77, 2004.

[166] A. Brahme, M. H. Alvi, D. Saylor, J. Fridy, and A. D. Rollett. 3D reconstruction of microstructure in a commercial purity aluminum. *Scripta Materialia*, 55(1):75–80, 2006.

[167] A. L. Cruz-Fabiano, R. Logé, and M. Bernacki. Assessment of simplified 2D grain growth models from numerical experiments based on a level set framework. *Computational Materials Science*, 92:305–312, 2014.

[168] L. Maire, B. Scholtes, C. Moussa, D. Pino Muñoz, N. Bozzolo, and M. Bernacki. Improvement of 3-D mean field models for pure grain growth based on full field simulations. *Journal of Materials Science*, 51(24):10970–10981, 2016.

[169] J. Furstoss, M. Bernacki, C. Ganino, C. Petit, and D. Pino-Muñoz. 2D and 3D simulation of grain growth in olivine aggregates using a full field model based on the level set method. *Physics of the Earth and Planetary Interiors*, 283:98–109, 2018.

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

[171] F. Alauzet. A changing-topology moving mesh technique for large displacements. *Engineering with Computers*, 30(2):175–200, 2014.

[172] R. Loubère, P. H. Maire, M. Shashkov, J. Breil, and S. Galera. ReALE: A reconnection-based arbitrary-Lagrangian-Eulerian method. *Journal of Computational Physics*, 229(12):4724–4761, 2010.

[173] M. Shakoor. *Three-dimensional numerical modeling of ductile fracture mechanisms at the microscale*. PhD thesis, MINES ParisTech, 2016.

[174] C. Dapogny, C. Dobrzynski, and P. Frey. Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *Journal of Computational Physics*, 262:358–378, 2014.

[175] C. Gruau. *Metric generation for anisotropic mesh adaption with numerical applications to material forming simulation*. PhD thesis, MINES ParisTech, 2004.

[176] L. A. Freitag and C. Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. *International Journal for Numerical Methods in Engineering*, 40(21):3979–4002, 1997.

[177] X. Li. *Mesh modification procedures for general 3d non-manifold domains*. PhD thesis, Renselear Polytechnic Institute, 2003.

[178] X. Li, J.-F. Remacle, N. Chevaugeon, and M. S. Shephard. Anisotropic mesh gradation control. In *IMR*, pages 401–412, 2004.

[179] X. Li, M. S. Shephard, and M. W. Beall. 3D anisotropic mesh adaptation by mesh modification. *Computer Methods in Applied Mechanics and Engineering*, 194(48-49):4915–4950, 2005.

[180] J. F. Remacle, X. Li, M. S. Shephard, and J. E. Flaherty. Anisotropic adaptive simulation of transient flows using discontinuous Galerkin methods. *International Journal for Numerical Methods in Engineering*, 62(7):899–923, 2005.

[181] C. Geuzaine and J. F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[182] F. Alauzet. *Adaptation de maillage anisotrope en trois dimensions: applications aux simulations instationnaires en mécanique des fluides*. PhD thesis, Montpellier 2, 2003.

[183] F. Alauzet. Size gradation control of anisotropic meshes. *Finite Elements in Analysis and Design*, 46(1-2):181–202, 2010.

[184] M. Shakoor, M. Bernacki, and P.-O. Bouchard. A new body-fitted immersed volume method for the modeling of ductile fracture at the microscale: Analysis of void clusters and stress state effects on coalescence. *Engineering Fracture Mechanics*, 147:398–417, 2015.

[185] M. Shakoor, M. Bernacki, and P.-O. Bouchard. A new body-fitted immersed volume method for the modeling of ductile fracture at the microscale: Analysis of void clusters and stress state effects on coalescence. *Engineering Fracture Mechanics*, 147:398–417, 2015.

[186] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery (SPR) and adaptive finite element refinement. *Computer Methods in Applied Mechanics and Engineering*, 101(1-3):207–224, 1992.

[187] O. C. Zienkiewicz and J. Z. Zhu. Superconvergence and the superconvergent patch recovery. *Finite Elements in Analysis and Design*, 19(1-2):11–23, 1995.

[188] N.-E. Wiberg. Superconvergent Patch Recovery — a key to quality assessed FE solutions. *Advances in Engineering Software*, 28(2):85–95, 1997.

[189] T. Coupez, H. Digonnet, and R. Ducloux. Parallel meshing and remeshing. *Applied Mathematical Modelling*, 25(2):153–175, 2000.

[190] C. Gruau and T. Coupez. 3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric. *Computer Methods in Applied Mechanics and Engineering*, 194(48-49):4951–4976, 2005.

[191] J. Bruchon, H. Digonnet, and T. Coupez. Using a signed distance function for the simulation of metal forming processes: Formulation of the contact condition and mesh adaptation. From a Lagrangian approach to an Eulerian approach. *International Journal for Numerical Methods in Engineering*, 78(8):980–1008, may 2009.

[192] C. Herring. Surface tension as a motivation for sintering. In *Fundamental Contributions to the Continuum Theory of Evolving Phase Interfaces in Solids*, pages 33–69. Springer, 1999.

[193] R. I. Saye and J. a. Sethian. Cozzarelli Prize Winner: The Voronoi Implicit Interface Method for computing multiphase physics. *Proceedings of the National Academy of Sciences*, 108(49):19498–19503, 2011.

[194] R. I. Saye and J. A. Sethian. Analysis and applications of the Voronoi Implicit Interface Method. *Journal of Computational Physics*, 231(18):6051–6085, 2012.

[195] J. C. Cantrell. *Geometric topology*. Academic Press, 1979.

[196] C. de Boor. *A Practical Guide to Splines*, volume 27 of *Applied Mathematical Sciences*. Springer, New York, NY, 1978.

[197] M. Amirfakhrian and H. Mafikandi. Approximation of parametric curves by Moving Least Squares method. *Applied Mathematics and Computation*, 283:290–298, 2016.

[198] Q. Wang, W. Zhou, Y. Cheng, G. Ma, X. Chang, Y. Miao, and E. Chen. Regularized moving least-square method and regularized improved interpolating moving least-square method with nonsingular moment matrices. *Applied Mathematics and Computation*, 325:120–145, 2018.

[199] G. Compère, J. F. Remacle, and E. Marchandise. Transient mesh adaptivity with large rigid-body displacements. *Proceedings of the 17th International Meshing Roundtable, IMR 2008*, (iMMC):213–230, 2008.

[200] G. Compère, J.-F. Remacle, J. Jansson, and J. Hoffman. A mesh adaptation framework for dealing with large deforming meshes. *International Journal for Numerical Methods in Engineering*, 82(7):843–867, 2010.

[201] H. Hallberg. Influence of anisotropic grain boundary properties on the evolution of grain boundary character distribution during grain growth - A 2D level set study. *Modelling and Simulation in Materials Science and Engineering*, 22(8), 2014.

[202] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. *OSDI 2004 - 6th Symposium on Operating Systems Design and Implementation*, pages 137–149, 2004.

[203] D. W. Walker. Standards for message-Passing in a Distributed Memory Environment. Technical report, Center for Research on Parallel Computing (CRPC), Oak Ridge National Lab., TN (United States), 1992.

[204] É. Laucoin and C. Calvin. A parallel front-tracking method for two-phase flow simulations. In *Parallel Computational Fluid Dynamics 2004*, pages 289–296. Elsevier, 1996.

[205] A. da Silveira Neto, M. Villar, A. Roma, R. Serfaty, B. van Wachem, and M. Pivello. A parallel front-tracking algorithm for the simulation of rising bubbles. *9th International Conference on Multiphase Flow, ICMF-2016*, (February):1–6, 2016.

[206] K. L. Pan and G. C. Yin. Parallel strategies of front-tracking method for simulation of multiphase flows. *Computers and Fluids*, 67:123–129, 2012.

[207] D. M. Sussman. cellGPU: Massively parallel simulations of dynamic vertex models. *Computer Physics Communications*, 219:400–406, 2017.

[208] P. Madhikar, J. Åström, J. Westerholm, and M. Karttunen. CellSim3D: GPU accelerated software for simulations of cellular growth and division in three dimensions. *Computer Physics Communications*, 232:206–213, 2018.

[209] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal of Scientific Computing*, 20(1):359–392, 1998.

[210] Y. Mesri, H. Digonnet, and T. Coupez. Advanced parallel computing in material forming with CIMLib. *European Journal of Computational Mechanics*, 18(7-8):669–694, 2009.

[211] A. Laasraoui and J. J. Jonas. Prediction of steel flow stresses at high temperatures and strain rates. *Metallurgical Transactions A*, 22(7):1545–1558, 1991.

[212] A. Cruz-Fabiano. *Modelling of crystal plasticity and grain boundary migration of 304L steel at the mesoscopic scale*. PhD thesis, MINES ParisTech, 2014.

[213] J. Bailey and P. B. Hirsch. The recrystallization process in some polycrystalline metals. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 267(1328):11–30, 1962.

[214] C. H. Davies. Growth of nuclei in a cellular automaton simulation of recrystallisation. *Scripta Materialia*, 36(1):35–40, 1997.

[215] F. Villaret, B. Hary, Y. de Carlan, T. Baudin, R. Logé, L. Maire, and M. Bernacki. Probabilistic and deterministic full field approaches to simulate recrystallization in ODS steels. *Computational Materials Science*, 179(March):109646, 2020.

[216] J. E. Taylor. The motion of multiple-phase junctions under prescribed phase-boundary velocities. *Journal of Differential Equations*, 119(1):109–136, 1995.

[217] J. Fausty, B. Murgas, S. Florez, N. Bozzolo, and M. Bernacki. A new level set-finite element formulation for anisotropic grain boundary migration. jun 2020.

[218] G. DeWit and J. S. Koehler. Interaction of dislocations with an applied stress in anisotropic crystals. *Phys. Rev.*, 116:1113–1120, 1959.

[219] J. Zhang, W. Ludwig, Y. Zhang, H. H. B. Sørensen, D. J. Rowenhorst, A. Yamanaka, P. W. Voorhees, and H. F. Poulsen. Grain boundary mobilities in polycrystals. *Acta Materialia*, 191:211–220, 2020.

[220] A. Bhattacharya, Y. F. Shen, C. M. Hefferan, S. F. Li, J. Lind, R. M. Suter, and G. S. Rohrer. Three-dimensional observations of grain volume changes during annealing of polycrystalline Ni. *Acta Materialia*, 167:40–50, 2019.

[221] K. Chen, J. Han, X. Pan, and D. J. Srolovitz. The grain boundary mobility tensor. *Proceedings of the National Academy of Sciences*, 117(9):4533–4538, mar 2020.

[222] J. Han, S. L. Thomas, and D. J. Srolovitz. Grain-boundary kinetics: A unified approach. *Progress in Materials Science*, 98(February):386–476, 2018.

[223] R. Stanley. *Enumerative combinatorics*. Cambridge University Press, Cambridge, 1999.

[224] J. K. Mackenzie. Second paper on statistics associated with the random disorientation of cubes. *Biometrika*, 45(1-2):229–240, 1958.

[225] B. Murgas. *Towards a fine description of the grain boundary mobility on the modeling of microstructural evolutions*. PhD thesis, MINES ParisTech, 2021.

[226] K. Alvarado. *Influence of Smith-Zener pinning phenomena on the grain growth behaviour: multiscale approach and application to nickel-based superalloys*. PhD thesis, MINES ParisTech, 2021.

[227] D. Ruiz. *Full field modeling of discontinuous dynamic recrystallization in a CPFEM context*. PhD thesis, MINES ParisTech, 2020.

RÉSUMÉ

Les industries stratégiques ont un besoin toujours plus croissant dans l'utilisation des matériaux métalliques. Il y ainsi aujourd'hui une demande forte dans le fait d'être capable de prédire l'évolution des microstructures de ces matériaux lors de leur mise en forme car leurs caractéristiques microstructurales sont intrinsèquement liées à leurs propriétés en service. Dans ce contexte de problèmes massivement multi-domaines, de nombreuses méthodes dites à champ complet et qui décrivent les réseaux de joints de grains à l'échelle de la microstructure ont été développées ces quarante dernières années. Dans un contexte de grandes à très grandes déformations comme c'est le cas pour les procédés industriels de mise en forme à chaud, l'approche level-set (LS) couplée à une formulation éléments finis (EF) et des méthodes de remaillage reste l'approche la plus générique et la plus efficace.
Si des améliorations récentes ont été rapportées (logiciel DIGIMU par exemple), la principale faiblesse de cette approche reste son coût numérique qui limite le nombre de grains considérés dans les simulations et implique des temps de calculs importants, principalement en 3D.
Dans ces travaux, les performances réelles de l'approche LS-FE sont étudiées et une alternative, dénommée ToRealMotion, capable de réaliser des simulations massivement multi-domaines en 2D, est introduite. Cette nouvelle approche, appartenant à la famille des méthodes de type « suivi de front », inclut différentes innovations et a été parallélisée. Les propriétés géométriques des interfaces intervenant dans le calcul des cinétiques sont évaluées uniquement aux interfaces et la migration du réseau de joints de grains est réalisée en lagrangien tout en conservant un maillage EF conforme et global (sous-entendant que le coeur des grains est également maillé). Cette méthodologie permet ainsi une meilleure adaptabilitée aux mécanismes intragranulaires que les approches de type « suivi de front » classiques. Bien sur, une des ambitions principales de ce travail réside dans l'amélioration des performances numériques de l'état de l'art tout en conservant la précision et le côté générique (multi-mécanismes) de l'approche LS-FE en grandes déformations. Ainsi, de nombreux cas tests 2D en croissance de grains (GG) et recristallisation (ReX) sont réalisés pour prouver l'efficacité de la méthode. Les résultats s'illustrent par une réduction importante des temps de calcul et offrent d'importantes perspectives dans le contexte de la métallurgie numérique.

MOTS CLÉS

Eléments Finis, Modèles Lagrangiens, Maillage des interfaces, Level-set, Simulations multidomaines.

ABSTRACT

Strategic industries make extensive use of metallic materials. Today, there is a strong demand from these industries to predict, during hot metal forming processes, the microstructural evolutions of these materials, which are of prime importance concerning their final in-use properties.
In this context of massive multi-domain problems, numerous full-field approaches that describe grain boundary (GB) network motion at the mesoscopic scale have been developed for forty years. When very large deformations are investigated, as in the context of realistic industrial thermomechanical treatments, the level-set (LS) approach in the context of finite element (FE) formulations and meshing/remeshing algorithms remains the most powerful and versatile numerical tool.
Even if recent improvements were realized (context of DIGIMU software), the main weakness of this approach remains its numerical cost, which limits the number of grains considered (small representative volume elements) and still implies long calculation times, especially in 3D.
In these works, the performance of FE-LS models is studied and a new method denominated ToRealMotion, capable to perform 2D massive multi-domain simulations is introduced. This new method, belonging to the family of front-tracking methods, includes various innovations and has been parallelized. Geometrical properties used in the kinetics are only computed at the interfaces, and GBs migration is defined thanks to a Lagrangian scheme, keeping a FE discretization of the bulk of the grains through the concept of body-fitted unstructured FE meshes. This aspect allows for higher adaptability than traditional Front-Tracking models. Of course, one of the main ambitions of this new approach is the improvement of the computational performance when simulating evolving microstructures, while keeping the precision and versatility of the FE-LS approach. As such, 2D numerical cases in the context of grain growth (GG) and recrystallization (ReX) are provided to prove the efficiency of this new approach. These results show impressive reductions in the computational costs and offer promising perspectives on the modeling of massive multi-domain simulations in terms of numerical performance and precision in the modeling of numerous solid-state phenomena.

KEYWORDS

Finite Elements, Lagrangian Models, Body-Fitted Mesh, Level-set, Multidomain simulations.