



A novel highly efficient Lagrangian model for massively multidomain simulation applied to microstructural evolutions

Sebastian Florez*, Karen Alvarado, Daniel Pino Muñoz, Marc Bernacki

Mines-ParisTech, PSL-Research University, CEMEF – Centre de mise en forme des matériaux, CNRS UMR 7635, CS 10207 rue Claude Daunesse, 06904 Sophia Antipolis Cedex, France

Received 24 February 2020; received in revised form 10 April 2020; accepted 29 April 2020

Available online xxxx

Abstract

In this paper, 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 Finite Element (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 grain growth mechanism (GG) and it will be compared to a more classical front capturing approach based on a Level-Set (LS) description of grain interfaces which have been proved to be a powerful method for this kind of simulations.

© 2020 Elsevier B.V. All rights reserved.

Keywords: Front-tracking; Body-fitted; Massive-multidomain; Grain-growth.

1. Introduction

Multiple numerical models have been developed to simulate multidomain problems. In the context of the Finite Element (FE) Method, the Level-set (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 phenomena as Grain Growth (GG) [1] and Recrystallization (ReX) could appear [2–5]. 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.

Some optimizations could be applied to the FE-LS method with the objective of reduce the computational cost (hence the CPU-Time) without reducing its accurateness [6,7]. However, these optimizations have a certain limit

* Corresponding author.

E-mail address: sebastian.florez@mines-paristech.fr (S. Florez).

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 [8] and further discussed in [9] 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 context of vertex approaches [10–14] or front-tracking methodology [15–19]. These methods explicitly describe the interfaces in terms of vertices. For 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 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 them.

In this paper, a new method for the simulation of evolving microstructures by curvature flow is presented. The method is inspired by the front-tracking 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 [1,20–22]. These comparisons will be based on specific test cases for 2D-GG, using the same approach and results presented in a previous work [7]. Given the application of the new model in this article, 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.

2. Numerical method

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 [23], 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.

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. 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. 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.

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. Fig. 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

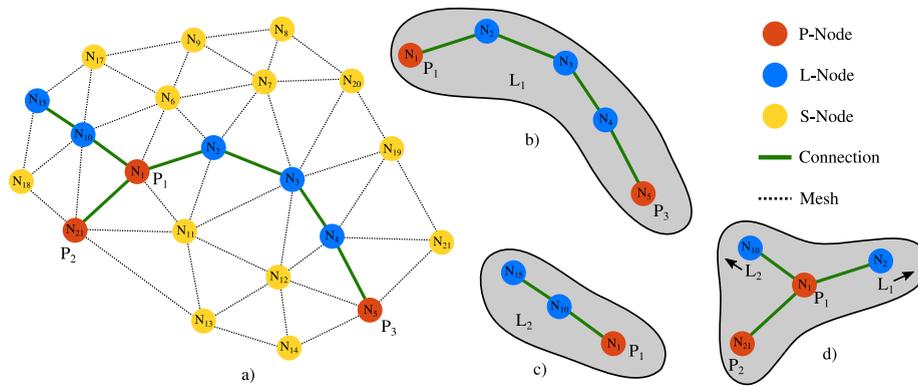


Fig. 1. Discrete geometric connectivity immersed in a triangular mesh. (a) Mesh containing 21 Nodes: 3 P-Nodes (red), 5 L-Nodes (blue) and 13 S-Nodes (yellow). (b) structure of line L_1 , (c) structure of line L_2 , (d) structure of point P_1 . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1

Example of the data structure of point P_1 from Fig. 1.

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 2

Example of the data structure of line L_1 from Fig. 1.

Initial point	P_1
Node set	
Local index	Node
0	N_2
1	N_3
2	N_4
Final point	P_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 does 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 1 and 2 summarize an example of the data structure of point P_1 and line L_1 from Fig. 1.

Finally the data set of geometric surfaces contains a non empty set of elements, a set of S-Nodes, a set of limit lines and a set of limit points. Here, order has been chosen not to be relevant, thus all sets are unordered. Fig. 2(a) shows the same discretization of Fig. 1(a) with additional element information and Fig. 2(b-d) show its decomposition in surfaces S_1 , S_2 and S_3 . Table 3 presents an example for the data structure of Surface S_2 .

2.2. Preprocessor : LS-TRM interface and geometric reconstruction

Considering the importance and the capabilities of the LS method, we have consider 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

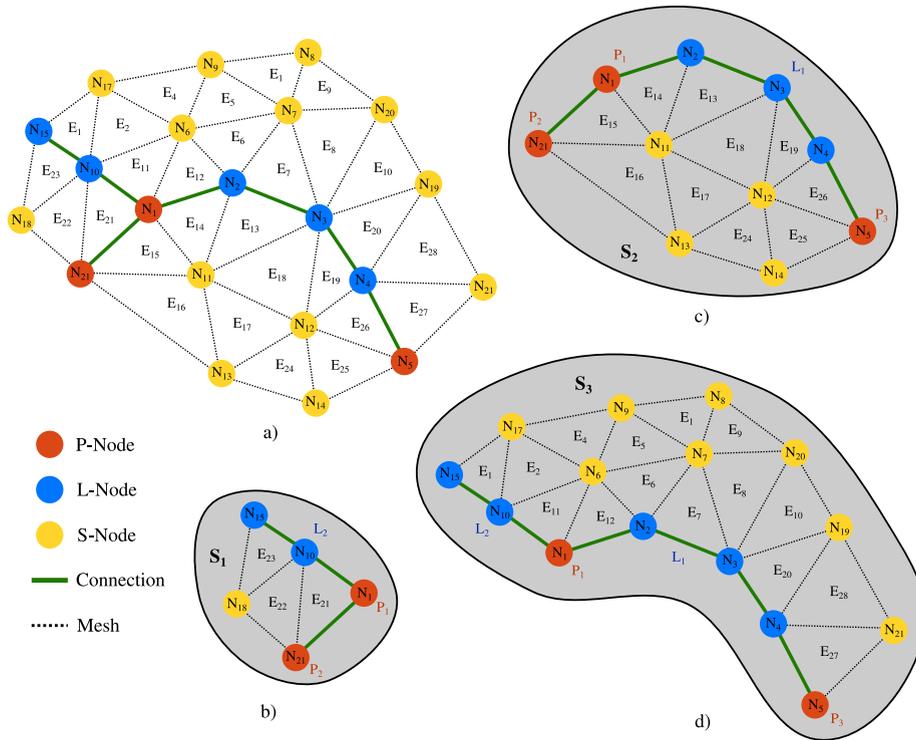


Fig. 2. Discrete geometric connectivity immersed in a triangular mesh. (a) Mesh of Fig. 1(a) with additional element information (b) structure of surface S_1 , (c) structure of surface S_2 , (d) structure of surface S_3 .

Table 3

Example of the data structure of surface S_2 from Fig. 2.

Element set	Node set	Point set	Line set
E_{15}	N_{11}	P_1	L_1
E_{13}	N_{13}	P_2	
E_{16}	N_{12}	P_3	
E_{14}	N_{14}		
E_{17}			
E_{18}			
E_{24}			
E_{19}			
E_{25}			
E_{26}			

mesh and a set of nodal data representing the degree of each node as explained at the beginning of Section 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 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 need to be performed on line entities for the presented 2D model.

2.2.1. Body fitted mesh

Normally, when using the LS method, interfaces are described implicitly as the zero iso-value of the considered LS fields [2,24], 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. Previous works have been done

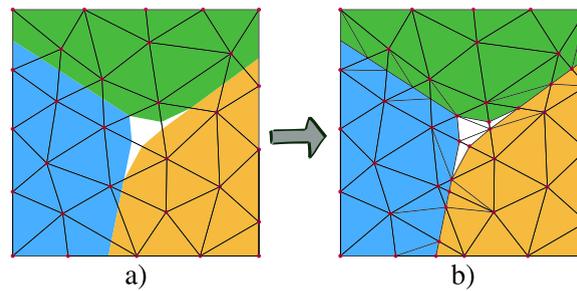


Fig. 3. Example of behavior of the *fitting* procedure presented in [25,26] when performed over a triangular mesh with 3 LS fields linearly interpolated, each color represents a phase. (a) initial state, interfaces are implicitly described, (b) state after applying the *fitting* algorithm, nodes appear at the interface between 2 phases but not at the multiple junction. Vacuum regions are preserved and some void elements are created.

addressing this particular purpose: first, a remeshing technique allowing to adapt the mesh to the implicit interface was presented in [25] and discussed further in [26] via a *fitting* algorithm, using a very similar approach as in [27]. This procedure is well adapted when the interface corresponds to the boundaries between two domains, nonetheless, when the number of domains increases, multiple points (boundary between more than two domains) may appear, the presence of these multiple points leads to the appearance of vacuum regions on the interpolated LS field, see Fig. 3(a). If the remeshing procedure presented in [25] is applied under this circumstance, void elements instead of multiple points will appear, see Fig. 3(b), note that some of the created nodes lie on the boundary between two domains but not at the multiple junction.

A variant of this remeshing technique that solves this problem was presented in [7], here, a *fitting and joining* algorithm (FJA) closes all vacuum regions within elements and then fit the interfaces at the edges, producing a body-fitted mesh without vacuum. Fig. 4 illustrates the behavior of the algorithm when performed on a typical multiple junction of three domains. Fig. 4(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 4(b)), of course, these adaptations are done without disrupting the fitting of the mesh. The reader is referred to [7] for further information. Fig. 4(c) shows the resulting body-fitted mesh after the adaptation procedure.

2.2.2. 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 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 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 equal 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 equal 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. 5 shows an example of nodal geometric tagging over the body fitted mesh obtained at the end of Section 2.2.1.

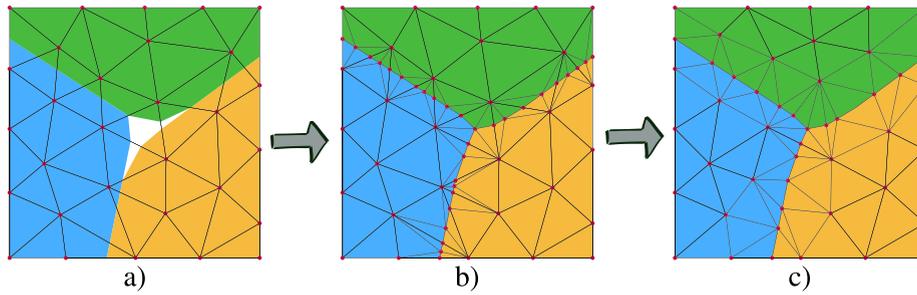


Fig. 4. Example of behavior of the *fitting and joining* procedure presented in [7] when performed over a triangular mesh with 3 LS fields interpolated linearly, each color represents a phase. (a) initial state, interfaces are implicitly described, (b) state after applying the *fitting and joining* algorithm, 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 [7].

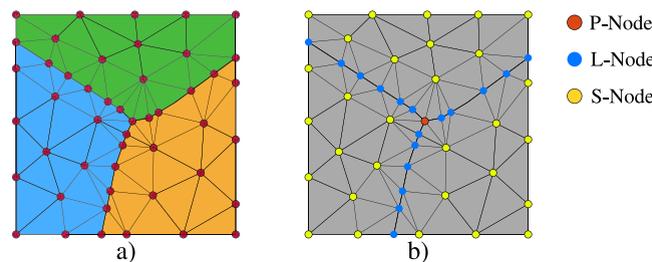


Fig. 5. Example of the nodal geometric tagging performed on a body fitted mesh. (a) considered topology and (b) resulting nodal tagging.

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.

2.2.3. 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 computation 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 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 its corresponding P-Node N_i is connected to any P-Node N_j on the triangular mesh, if it is true, the point of P-Node N_i will add N_j as a connection.

2.2.4. Lines reconstruction

Once all points are reconstructed, the line reconstruction is performed, Section 2.1 shows that line entities maintain an order denoted by where they appear on the line. 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 on 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.

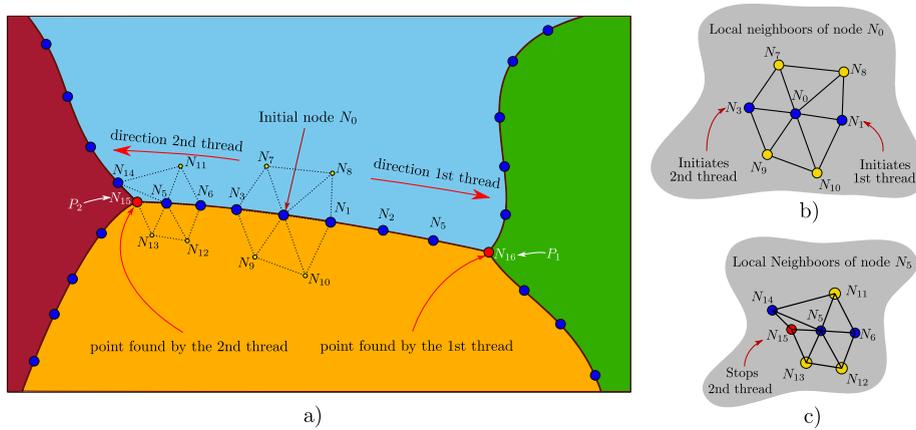


Fig. 6. Example of construction of a line in 2D. (a) L-Node N_0 is chosen as the initial node of the recursive algorithm, each treated L-Node launches the procedure on each adjacent L-Node not treated till it finds an adjacent P-Node. Only some relevant elements are plotted. (b) Local node patch of Node 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. 6(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. 6(b)) that still appear on the queue. The recursion stops once one of the neighbor nodes is a P-Node as in Fig. 6(c), where the coupled point P_1 of the P-Node N_5 is added to the list as a final point. Two lists are then created, the first is composed of the L-Nodes i_1, i_2, i_5 and point P_1 and the second by i_3, i_6, i_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 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).

2.2.5. 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. 7, 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 2.2.4. 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 of 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 ($Nodes(S_1)$):

$$ELTs = Elements(S_1) \tag{1}$$

where function $Elements(S_1)$ extracts the elements present in surface S_1 .¹

$$G_{NE} = \{Nodes(El_i) \quad \forall \quad El_i \in Elts\} \tag{2}$$

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 .

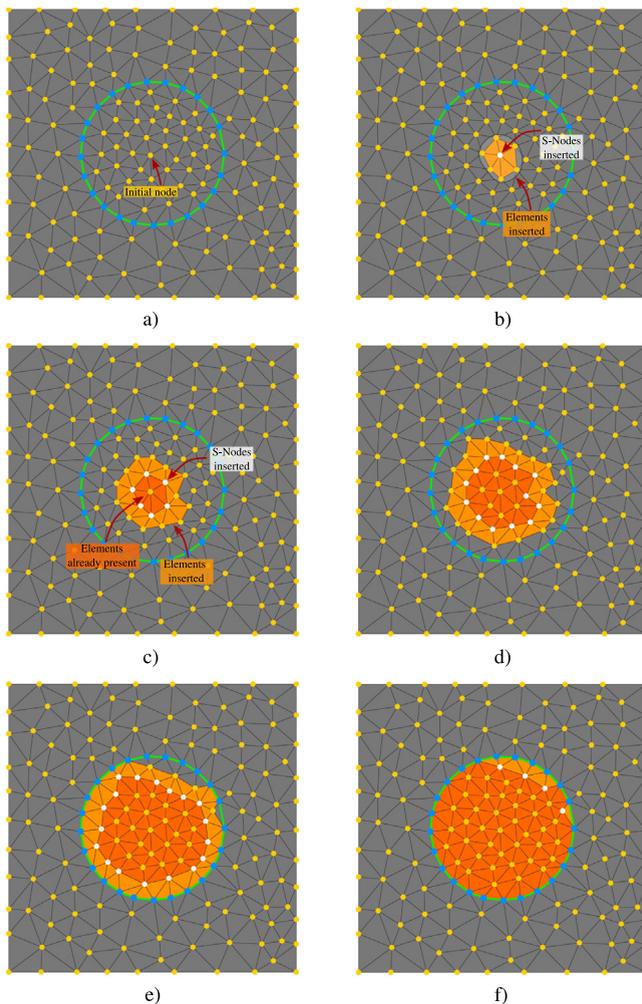


Fig. 7. Example of construction of a surface in 2D, L-Nodes — blue, S-Nodes — yellow, Line L_1 — green, elements in the queue — gray, elements attributed to Surface S_1 — orange. (a) an Initial S-Node is chosen, (b) the L-Node is inserted in Surface S_1 along with its element patch, (c) and (d) the new S-Nodes of the inserted elements are also inserted to S_1 . Each new S-Node inserts its un-attributed 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. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$G_{SNE} = \{N_i/N_i \in G_{NE} \text{ with } Type(N_i) = SNode\} \tag{3}$$

where function $Nodes(El_t_i)$ extracts the nodes belonging to El_t_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 $Type(N_i) = SNode$.

$$G_{newSN} = G_{SNE} \setminus Nodes(S_1) \tag{4}$$

the nodes in G_{newSN} are colored in white in each frame of Fig. 7. Once these nodes are computed, they are inserted in Surface S_1 and extracted from the queue.

The set of nodes filtered G_{other} from G_{NE} in Eq. (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 2.1.

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

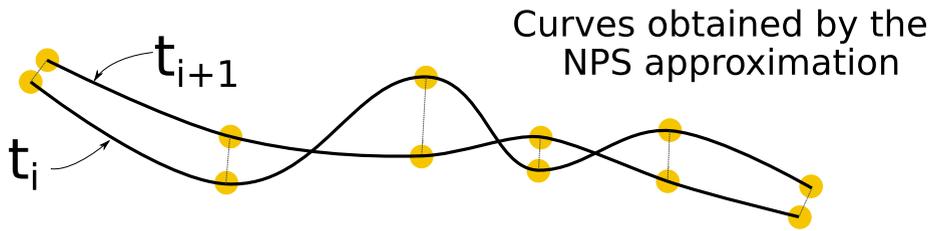


Fig. 8. Parametric piece-wise curve of a NPS approximation. Local curvatures are captured; The interface is capable to correct itself over time into a smooth interface if dt is low.

The 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 :

$$G_{Elt} = \{ Elements(N_i) \quad \forall \quad N_i \in G_{newSN} \} \setminus Elements(S_1). \quad (5)$$

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. 7(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.

2.3. Computation of geometrical properties: Curvature and normal

One of the objective 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 2.1 and the preprocessing procedure presented in Section 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) [28] which allow to capture the local variations of normal and curvature with a piece-wise polynomial of third degree.

2.3.1. 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 [29,30]) 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 Fig. 8).

The stability of the TRM model using natural parametric splines will be discussed in Section 4.2.

2.4. Selective remeshing operations : preserving topology

One of the main features of the presented TRM model is its remeshing procedure. Section 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,

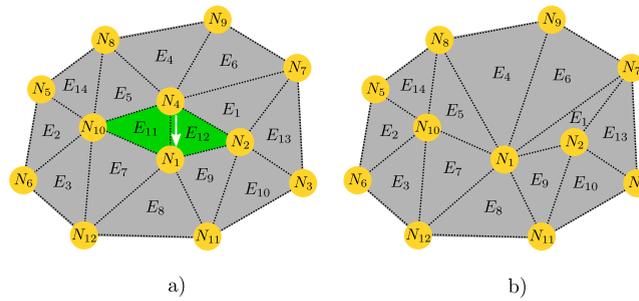


Fig. 9. Example of node collapsing. (a) initial state, elements to disappear are colored green (b) state after collapsing. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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 etc.). Generally, the element patch is replaced by another element patch with a 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 criterion is achieved [25]. The second uses the separate definition of local remeshing operations (vertex smoothing, node collapsing, edge splitting, edge swapping etc.) to be performed on the mesh, depending on its local requirements (edge size, element shape etc.) [31,32] 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 the 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 [25], 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.

2.4.1. 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 percent of the local mesh size desired. Fig. 9 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.

In our TRM model, node collapse should be avoided when the operation cannot determine 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 2.1:

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

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 Fig. 10) and if they are consecutive within the line (see

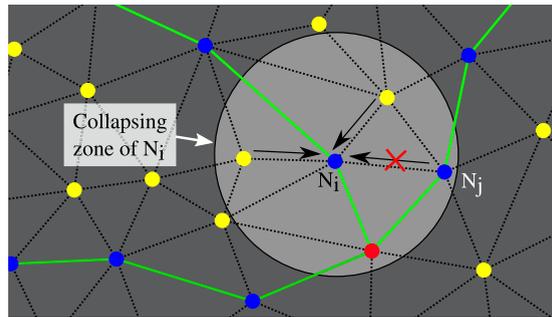


Fig. 10. Node Collapsing on surface. Some nodes are within the collapsing zone of N_i : Two S-Nodes (yellow) will collapse, one L-Node (blue) cannot collapse and one P-Node (red) cannot collapse. N_i cannot collapse P-Nodes (topological degree), N_i can collapse L-Nodes but N_j does not belong to the same line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

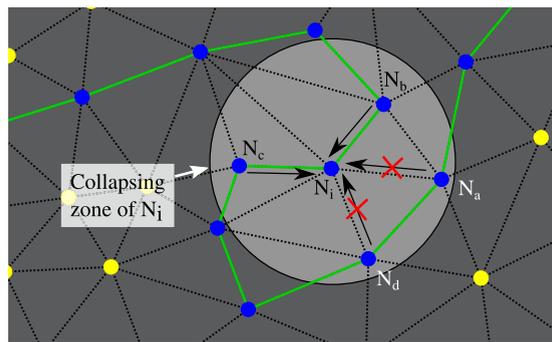


Fig. 11. Node Collapsing on surface. Some nodes are within the collapsing zone of N_i : four L-Nodes (blue) N_a , N_b , N_c and N_d . Only Nodes N_b and N_c can collapse as they are consecutive to N_i within the same line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 11). 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.

2.4.2. 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 . Fig. 12 shows one example of an edge splitting performed in the edge $N_{10} - N_2$, the element patch involved before and after the splitting is colored green.

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 appears on the list of connections of the other point as presented in Section 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 connected³ on one geometric line, the inserted node N_{new} will be of

³ Connected nodes does 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 Figs. 10 and 11 for examples of non-connected nodes.

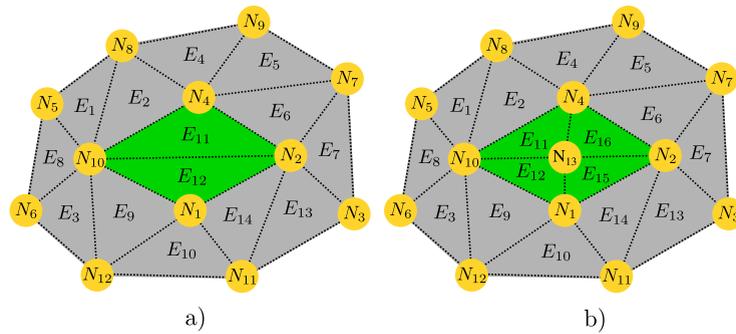


Fig. 12. Example of edge swapping. (a) initial state, the edge $N_2 - N_{10}$ will be split, elements involved in the procedure are colored green (b) state after splitting. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

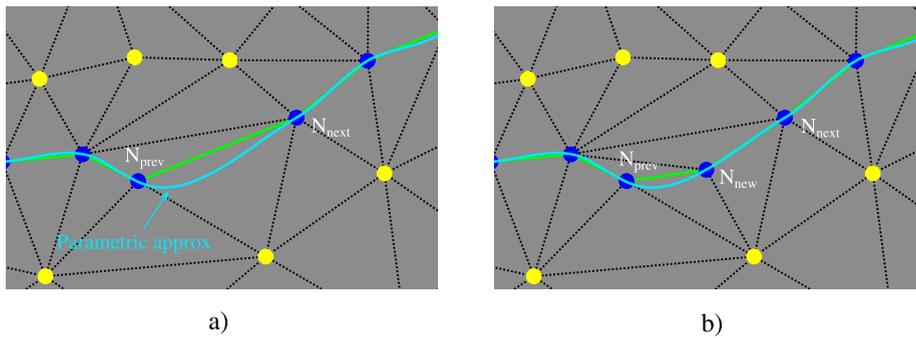


Fig. 13. Example of edge splitting of a L-L edge within a geometric line, connection within the geometric line is displayed green, the parametric approximation of the line is displayed in cyan. (a) initial state, the edge $N_{prev} - N_{next}$ will be split, (b) state after splitting, Node N_{new} is placed at the midpoint of the parametric approximation of the line between nodes N_{prev} and N_{next} . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

type $Type(N_{new}) = LNode$ and will be inserted on its corresponding position on the line data structure. Additionally, N_{new} will be inserted on the midpoint⁴ of the parametric approximation of the line between its previous and next nodes (L-Node or P-Node) on the line. An example for this case is shown in Fig. 13

- 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.

2.4.3. 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 [33,34].

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).

2.4.4. Selective vertex smoothing

Vertex smoothing is the most simple remeshing operator, it consists of moving nodes to a location where the quality Q of the local patch is improved (usually taken as the patch varycenter). Fig. 15 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.

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

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.

2.4.5. Selective vertex 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.

Fig. 16 shows an example of vertex gliding when performed on some of the L-Nodes (blue) of a given line using its parametric approximation (cyan).

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 2.3 or not, each node N_i of the mesh is allowed to be moved to a new position \vec{u}_i in a Lagrangian way using the velocity vector field \vec{v} and a given time step dt .

$$\vec{u}_i = \vec{u}_i^0 + \vec{v}_i \cdot dt. \quad (6)$$

where \vec{u}_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. Fig. 17 shows one example of element flipping on Element E_4 when node N_4 is moved to a location outside its element patch.

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 dividing by half 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 to do not damage the mesh while moving, moreover, as the corrections are made locally and for very few displacements, precision should not be highly influenced.

3. Grain growth modeling

The simulation of microstructural evolutions are given by the addition of complex and different mechanisms as GG [1,20–22,35], Recrystallization (ReX) [1–4,24,36,37] or Zener Pinning (ZP) [16–18,38]. 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 the following equation:

$$\vec{v} = -M\gamma\kappa\vec{n}, \quad (7)$$

where M is the mobility of the interface, γ the grain boundary energy, κ the local curvature in 2D and trace of the curvature tensor in 3D and \vec{n} the outside unit normal to the grain interface. Isotropic conditions will be consider hereafter: M is only dependent on the temperature and γ is constant.

In this section, two numerical models will be taken into account for the modeling of GG by capillarity using Eq. (7), the first will use the implicit description and evolution of grain boundaries by the resolution of PDEs over LS fields in a FE context [20,21,37], and the other will use the presented TRM model composed of the presented formalisms: the data structure and preprocessor of Sections 2.1 and 2.2, the approach for the computation of geometric properties of Section 2.3, the remeshing strategy of Section 2.4 and the Lagrangian description and evolution of grain boundaries of Section 2.5.

3.1. Level-set approach

The simulation of GG by capillarity using the level set approach is done by the substituting Eq. (7) for an advection equation per level-set field ϕ_i :

$$\frac{\partial \phi_i}{\partial t} - M\gamma\kappa\vec{n} \cdot \vec{\nabla}\phi_i = 0 \quad (8)$$

this equation can also be used in the context of anisotropic conditions of M and γ as in [39].

Complex microstructure topologies make Eq. (8) particularly difficult to solve by usual convective approaches. The local curvature κ involves second derivatives of the level-set function ϕ_i , whose computation may not be very precise or highly unstable. This is why an alternative solution has been developed by rewriting Eq. (8) by adding an additional assumption: the LS fields ϕ_i remain at all times signed distance functions ($|\vec{\nabla}\phi_i| = 1$). This hypothesis allows to transform Eq. (8) into a diffusive equation of the form:

$$\frac{\partial \phi_i}{\partial t} - M\gamma\Delta\phi_i = 0 \quad (9)$$

Eq. (11) is in general much more stable than Eq. (8) which avoids the direct estimation of κ .

Finally, additional optimizations to the modeling of GG have been developed over the years, allowing more precise and fast computations [2,24,25,40–43]. These optimizations along with a general algorithm for the complete modeling of GG are summarized in [7].

3.2. 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.

3.2.1. Velocity computation and interface migration

After obtaining a valid data structure for the whole polycrystal topology by means of the procedure explained in Section 2.2 the model is ready to initiate the simulation of the migration of grain boundary interfaces. 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 2.3. These geometrical values can be used on Eq. (7) to obtain a value for the local velocity vector \vec{v} for every node of the interface, i.e. for every P-Node and every L-Node in our structure.

With the local velocity \vec{v} obtained on the grain boundary, it is possible to move the grain boundary network by using the procedure explained in Section 2.5 over every node at the interface.

3.2.2. Remeshing strategy

Till now, the steps explained in Section 3.2.1 can be done iteratively to try to obtain the general behavior of the grain boundary migration, 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 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 its 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 its length and in order to maintain a certain accuracy on its approximation, it will be needed to insert some nodes in it, 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.

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 of 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 ($|\overline{N_i N_j}|$), if this distance $d_{ij} > \delta_c(N_i, N_j)$ and the collapsing is able to be done (see conditions in Section 2.4.1) the operation is performed. In our model we have chosen to make some relations for the values of δ_c :

$$\begin{cases} \delta_c(N_i, N_j) = d_{min} & \forall N_i / Type(N_i) = PNode \\ \delta_c(N_i, N_j) = d_{min} & \forall (N_i, N_j) / Type(N_i) = LNode, Type(N_j) = SNode \\ \delta_c(N_i, N_j) = 3d_{min} & \forall (N_i, N_j) / Type(N_i) = LNode, Type(N_j) = LNode \\ \delta_c(N_i, N_j) = 6d_{min} & \forall (N_i, N_j) / Type(N_i) = SNode, Type(N_j) = SNode \end{cases} \quad (10)$$

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 of only $S - Nodes$ will be 2 times than for edges composed of 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:

$$\begin{cases} \delta_s(N_i, N_j) = d_{max} & \forall (N_i, N_j) / Type(N_i) = PNode, Type(N_j) = PNode \\ \delta_s(N_i, N_j) = d_{max} & \forall (N_i, N_j) / Type(N_i) = PNode, Type(N_j) = LNode \\ \delta_s(N_i, N_j) = 3d_{max}/2 & \forall (N_i, N_j) / Type(N_i) = PNode, Type(N_j) = SNode \\ \delta_s(N_i, N_j) = d_{max} & \forall (N_i, N_j) / Type(N_i) = LNode, Type(N_j) = LNode \\ \delta_s(N_i, N_j) = 3d_{max}/2 & \forall (N_i, N_j) / Type(N_i) = LNode, Type(N_j) = SNode \\ \delta_s(N_i, N_j) = 3d_{max} & \forall (N_i, N_j) / Type(N_i) = SNode, Type(N_j) = SNode \end{cases} \quad (11)$$

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 on their normal direction, the higher the risk of stagnation (see Fig. 17). 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 Fig. 14) 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:

Remeshing algorithm

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 : $N_j \rightarrow N_i$

end if

end for

end for

for all S-Nodes : SN_i **do**

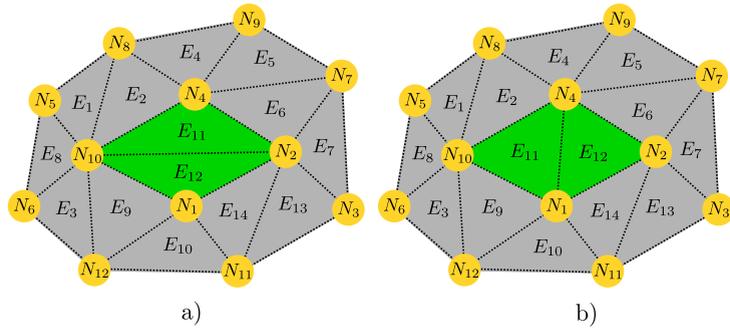


Fig. 14. Example of edge swapping. (a) initial state, the edge $N_2 - N_{10}$ will swap, elements involved in the procedure are colored green (b) state after swapping. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

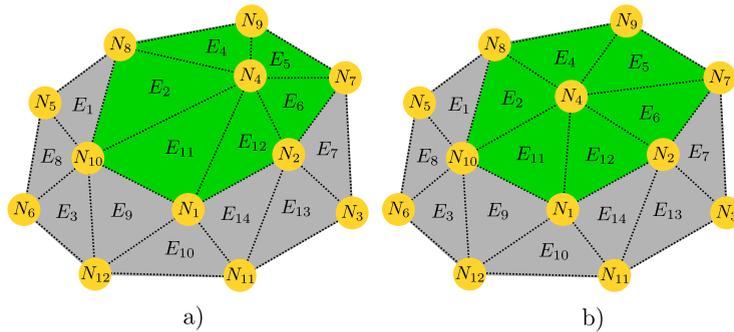


Fig. 15. Example of vertex smoothing. (a) initial state, the node N_4 will be smoothed, elements involved in the procedure are colored green (b) state after smoothing. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

```

selective vertex smoothing :  $SN_i$ 
end for
for all L-Nodes :  $LN_i$  do
    selective vertex gliding :  $LN_i$ 
end for
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 :  $N_i, N_j$ 
    end if
end for
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 :  $\{N_j, N_k\}$ 
        end if
    end for
end for
end for
    
```

3.2.3. 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

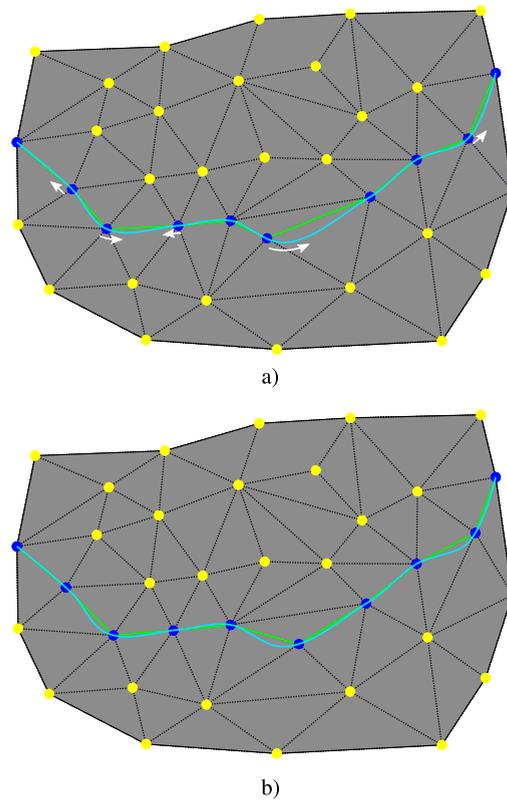


Fig. 16. Example of vertex gliding when performed on some of the L-Nodes (blue) of a given line using its parametric approximation (cyan), (a) initial state and upcoming gliding (white arrows), (b) final state after gliding. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

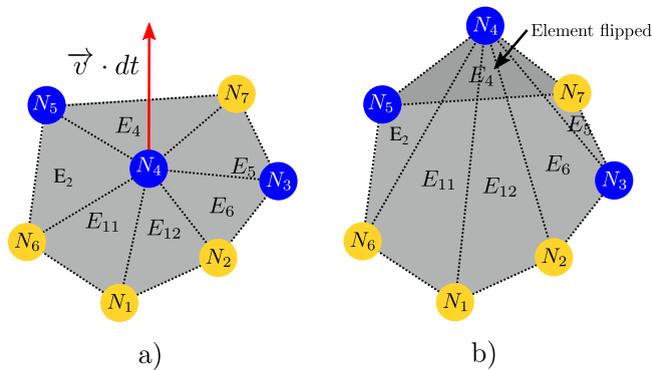


Fig. 17. Example of element flipping, (a) initial state, the displacement vector of N_4 $\vec{v} \cdot dt$ lies outside the element patch. (b) state after updating the position of N_4 , element E_4 has been flipped.

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 2.4.1, meaning that always when a phase disappears a point of its interface will remain at the moment of the event.

3.2.4. 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

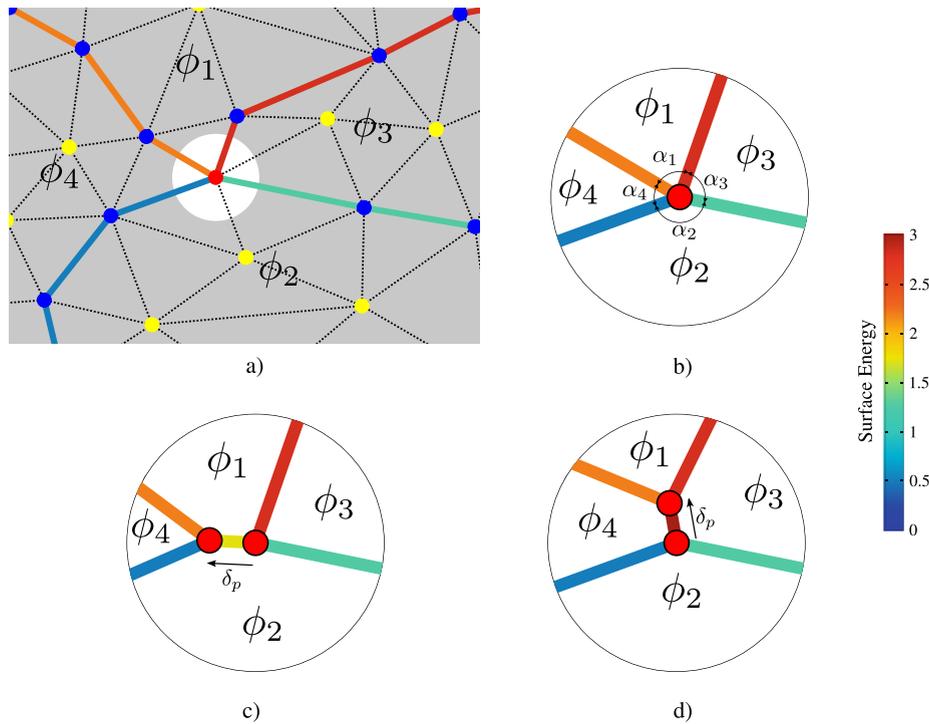


Fig. 18. Example of an multiple junction decomposition with anisotropic energies, (a) initial state. (b) detailed view of the multiple junction. (c) decomposition example, interfaces of ϕ_4 are separated from the initial multiple junction and a new interface is created ($\phi_1 - \phi_2$) (d) decomposition example, interfaces of ϕ_1 are separated from the initial multiple junction and a new interface is created ($\phi_4 - \phi_3$).

state can be obtained by creating new interfaces and redistributing the multiple junction in several triple junctions (multiple junction with 3 connected interfaces) [7]. In the case of anisotropic boundary conditions, the decomposition procedure is as follows: first, the decomposition procedure cannot 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. 18(a) and (b) with a multiple junction of 4 phases ($\phi_1, \phi_2, \phi_3, \phi_4$) with anisotropic surface properties, here the local energy state depends on the energy values, the length and the different angles between interfaces ($\alpha_1, \alpha_2, \alpha_3, \alpha_4$). Here the structure decomposition can occur in 2 different ways: Fig. 18(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. 18(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 [39]. This will be discussed further in a future publication focused on the simulation of microstructural evolutions with anisotropic properties using the TRM model.

When using isotropic boundary 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. 19, 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 2.4. In fact, this operation is not accessible by a reconnection procedure of the *star-connection* algorithm of [25]. It consists in the *splitting* of a node of the mesh and a special mesh reconnection which does not erase the pre-existent connections, note that all the edges from the previous mesh (Fig. 19(a)), still exist in the new mesh (Fig. 19(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

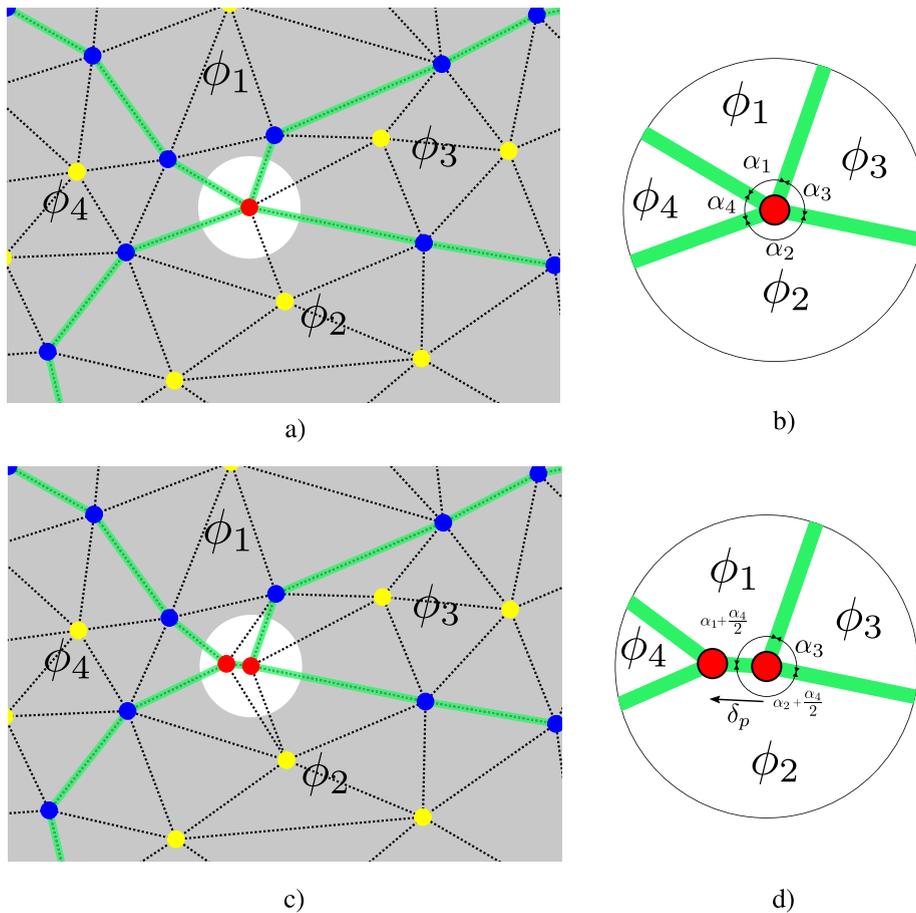


Fig. 19. Example of an multiple junction decomposition with isotropic boundary energies, (a) initial state. (b) detailed view of the multiple junction. Here $\alpha_4 < \alpha_1 < \alpha_3 < \alpha_2$, the choice of the phase to detach from the multiple point is determined by the lowest angle. (c) final 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 ($\phi_1 - \phi_2$), the new point will be placed along the line determined by the angle $\alpha_4/2$ measured from one of the two detached interface to the inner side of ϕ_4 at a distance δ_p from the initial multiple point.

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 grain boundary network.

4. Numerical results

A series of test cases have been realized to validate the TRM method using the same examples presented in a previous publication for the benchmark of the LS method under different remeshing conditions [7], results obtained on that publication 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 pure Grain Growth under isotropic conditions.

4.1. Considered geometries

Correspondingly to [7] the considered geometries will respond to different typical situations encountered during the Grain Growth process, the first three cases being adimensional 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.

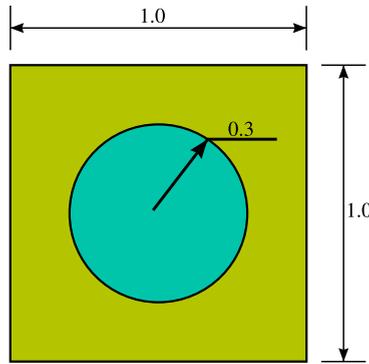


Fig. 20. Circle shrinkage test case: initial state [7].

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 on the microstructural web 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 [7] three remeshing approaches in the context of the LS GG simulation have been tested for each one of this geometrical configurations: Static Mesh (SM), Isotropic Mesh Adaptation (IMA) and a New Fitting and Joining Algorithm (NFJA) presented in this publication, 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 in [7], was the one using a static mesh.

4.2. Circle shrinkage

The circle shrinkage test case permits, 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 (\bar{n} and κ). Additionally, this case enables to compare the precision of the model through time by comparing it to the evolution of the equivalent analytical circle: as stated in [7], the evolution of a circle of surface S subjected to capillarity is simple given by the following linear equation:

$$\frac{ds}{dt} + 2\pi M\gamma = 0 \implies s(t) = S_0 - 2\pi M\gamma t. \tag{12}$$

The error to this analytical evolution will be computed in two ways, the first using a classical L2-error computation over the surface of the *circular*⁵ domain obtained by the TRM model, and the second using Eq. (11) of [7] which express the relative error in terms of surface change per increment:

$$E_{(h,t,\Delta t)} = \frac{\Delta s_{\phi(h,t,\Delta t)} - \Delta s_{(t,\Delta t)}}{\Delta s_{(t,\Delta t)}}. \tag{13}$$

where $\Delta s_{\phi(h,t,\Delta t)}$ is the total change of surface of the circle defined by ϕ , h is the mesh size tangent to the interfaces, t is the elapsed simulation time and Δt is the time step. 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).

Similarly to [7] we will compute a mean value for $\overline{\Delta s_{\phi(h,t,\Delta t)}}$ 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. Finally, the initial state and dimensions used in the circle shrinkage test are described in Fig. 20.

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. 21.

⁵ In fact, the numeric *circular* domain cannot define a perfect circle but the errors given by its real profile are ignored.

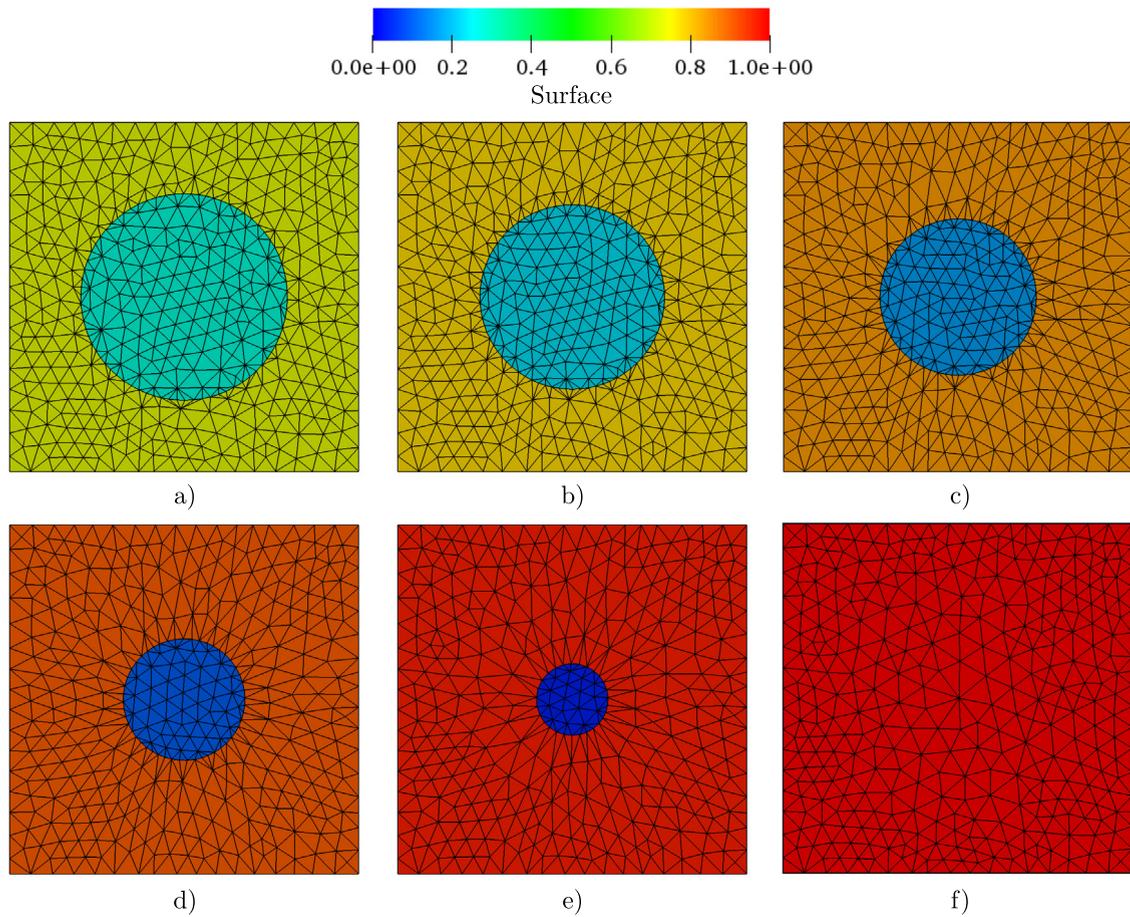


Fig. 21. Example of the evolution of the circle shrinkage test using the TRM method and a parameter $h_{trm} = 0.006$, the mesh and the surface field per domain are displayed. (a) Initial state, and (b) to (f) state at $t = 0.01$, $t = 0.02$, $t = 0.03$, $t = 0.04$ and $t = 0.05$.

Fig. 22 describes 4 subsets of evolutions of the circular phase, each subset corresponds to a different value of the time step dt maintaining a given value for mesh size parameter h_{trm} constant. **Fig. 23** illustrates the corresponding L2-Error evolution for the 4 subsets of **Fig. 22**. **Figs 22(c-d)** and **23(c-d)** exhibit the normal behavior of the model when subjected to normal conditions, the maximum error being not higher than 1%. **Fig. 22(a)** details the evolution of multiple simulations that eventually encountered continues instabilities in the velocity computation and produced a divergence from the analytical solution, **Fig. 22(b)** shows the transition between the normal response (for time steps $dt = 1e - 5$ and $dt = 2e - 5$) and an unstable response (for time steps $dt = 3e - 5$, $dt = 4e - 5$ and $dt = 5e - 5$) of the TRM model. These instabilities manifest in the L2-Error plots (**Fig. 23(a-b)**) as sudden increases in the error value that go up to 16% in the worst case (not shown in the figures). **Fig. 24** 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 velocity vector \vec{v} direction points to the outer region of the circular phase). As mentioned in Section 2.3.1 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 mesh size parameter $h_{trm} = 0.004$ and a time step $dt = 3 \times 10^{-5}$ lies exactly on the transition between the stable and the unstable region, **Figs. 22(b)** and **23(b)** show that with these values, the evolution of the surface is still very near the reference curve, obtaining a L2-Error

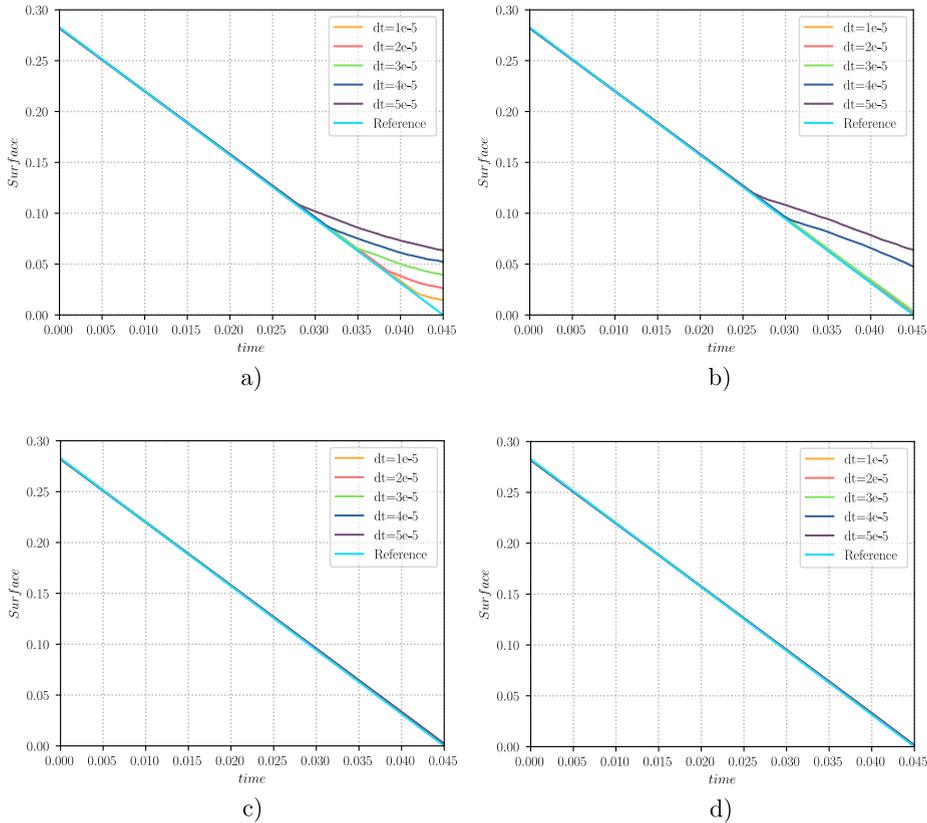


Fig. 22. Evolution of the surface of the circle shrinkage test for the circular phase using the TRM method for different values of dt . The corresponding analytic evolution (Reference) is also shown. (a) for a mesh size parameter $h_{trm} = 0.002$, (b) $h_{trm} = 0.004$, (c) $h_{trm} = 0.006$, (d) $h_{trm} = 0.008$.

of only 1% at the end of the simulation, while Fig. 24(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 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. 25 and hereafter we will restrain the content of the present article to this stable region.

Finally, Fig. 26 shows the different curves obtained when using Eq. (13) to compare the errors of the TRM method to the different approaches presented in [7], 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 TRM model in the stable region was the best.

4.3. 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. In the context of isotropic Grain Growth, the evolution over time of these triple points converges to a 120° - 120° - 120° quasi steady-state equilibrium (young equilibrium) given by the Herring’s equation [44]. 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.

The T-Junction test case (see Fig. 27) 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 [10] to calculate the velocity of the multiple point. The results will be compared here again to other models presented in [7].

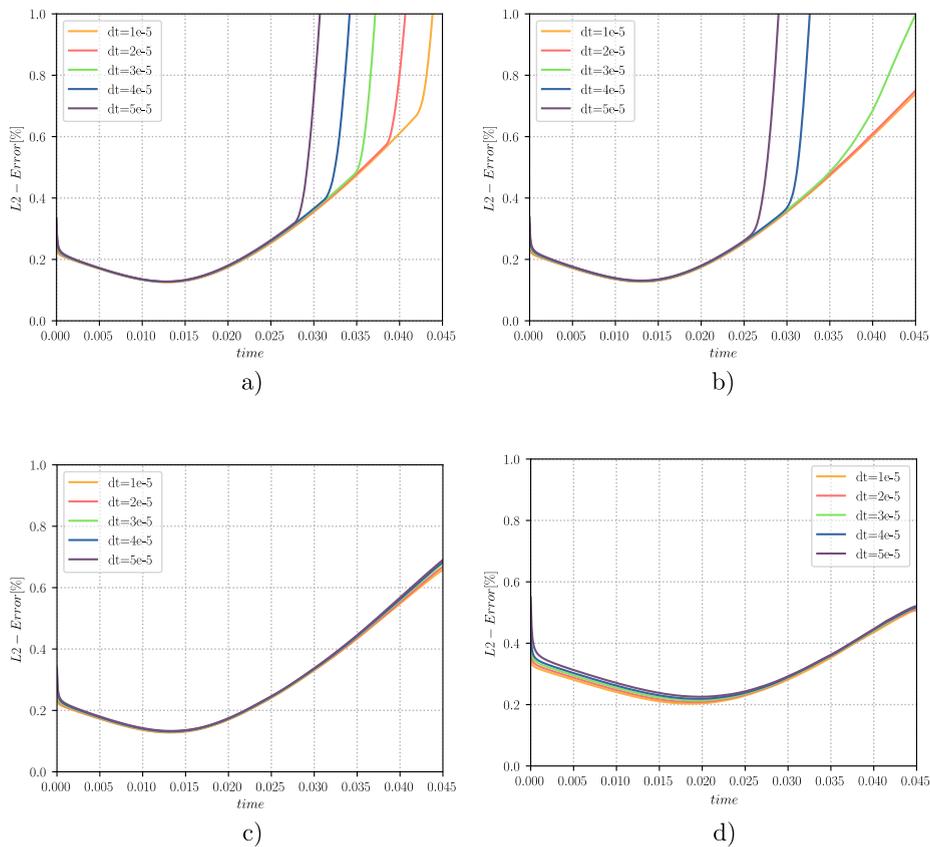


Fig. 23. L2-Error computation for the evolution of the surface of the circle shrinkage test for the circular phase using the TRM method for different values of dt . (a) for a mesh size parameter $h_{trm} = 0.002$, (b) $h_{trm} = 0.004$, (c) $h_{trm} = 0.006$, (d) $h_{trm} = 0.008$.

As stated in [7] 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 500 000 elements (where convergence has been obtained, see Fig 19 of [7]) has been used as reference.

A test using the same parameters as in [7] ($dt = 5 \times 10^{-5}$ and $h_{trm} = 0.006$) was made, the evolution of the mesh and surfaces for this test are described in Fig. 28. The triple point was found to be very near of the position of the reference case while the interfaces showed a similar curved shape (see Fig. 29) after 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 the 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 of the triple point of [13] as the results were almost the same as for the reference configuration.

4.4. Square-shrinkage case

Another very common case encountered on the evolution of microstructures subjected to grain growth is the disparition of a grain, the square shrinkage reproduces this case by putting a square shaped phase in the middle of other 4 phases (see Fig. 30). The angles values on each corner of the square-shaped phase make it to shrink and to eventually disappear. The symmetry of the model 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 points, reducing the total surface energy of the system [7].

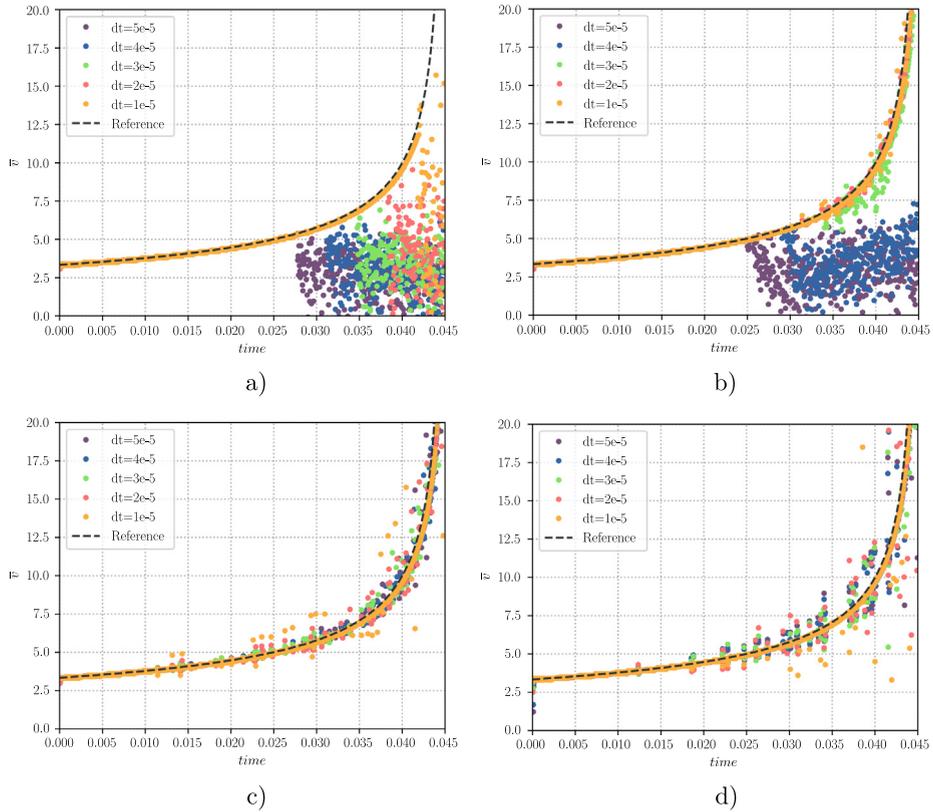


Fig. 24. Mean velocity of the interface the circle shrinkage test using the TRM method for different values of dt . The corresponding analytic velocity (Reference) is also shown. (a) for a mesh size parameter $h_{trm} = 0.002$, (b) $h_{trm} = 0.004$, (c) $h_{trm} = 0.006$, (d) $h_{trm} = 0.008$.

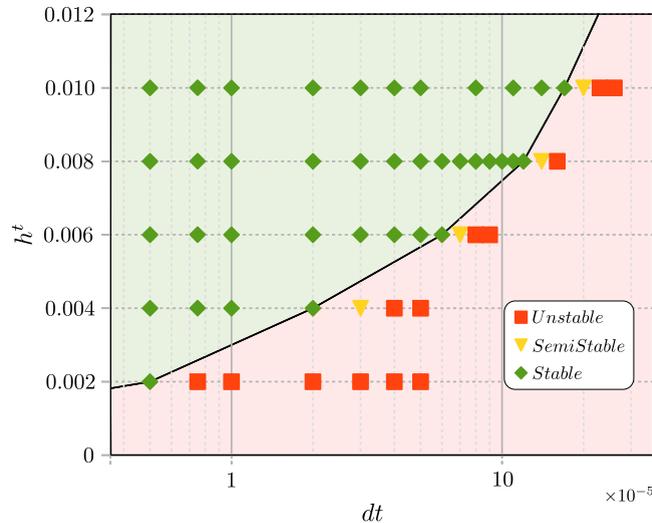


Fig. 25. Stability of the model in function of the mesh size h_{trm} parameter and the time step dt .

As explained in Section 3.2.4, the final decomposition state should be taken accordingly 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

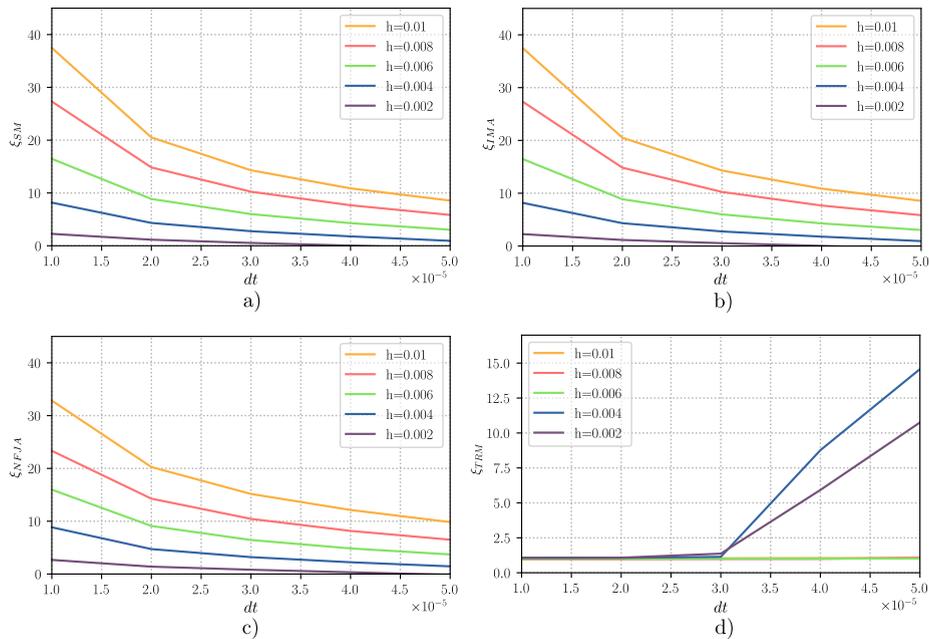


Fig. 26. Comparisons of the error for different models for the circle shrinkage test using Eq. (13). (a) Level-Set FE Static Mesh (SM) (b) Level-Set FE Isotropic Mesh Adaptation (IMA) (c) Level-Set FE New Fitting and Joining Algorithm (NFJA), (d) TRM (TRM).

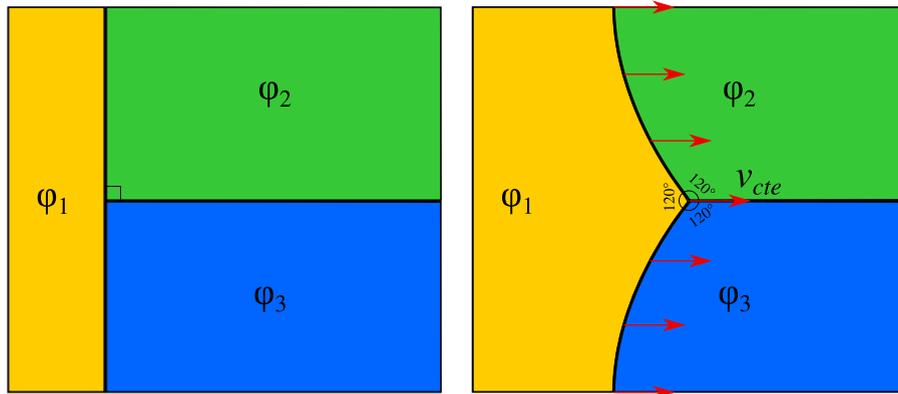


Fig. 27. T-junction case. left: initial state, right: steady state [7].

was imposed at the initial state, these angles should be around $90^\circ + \xi_{an}$ 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 difference 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. 30 are valid.).

One test for the square shrinkage case using the same parameters as in [7] ($dt = 5 \times 10^{-5}$ and $h_{trm} = 0.006$) was made, the evolutions of the mesh and surfaces for this test are described in Fig. 31

Fig. 32 illustrates the comparison of the TRM model and the methods used in [7] to the reference case (Using a LS FE method with a static mesh composed of 2 millions elements) after $t = 0.05$. The error on the area of the square shaped phase was $\xi_{TRM} = 6.6\%$ while for the other cases was $\xi_{NFJA} = 38.4\%$ and $\xi_{IMA} = 16.3\%$.

4.5. A 2D-10000 grains case

Finally, a massive multidomain test composed of 10000 initial grains is considered. Here, the TRM model will be compared to the Isotropic Mesh Adaptation (IMA) case and to the Static Mesh (SM) case from [7] using a mesh

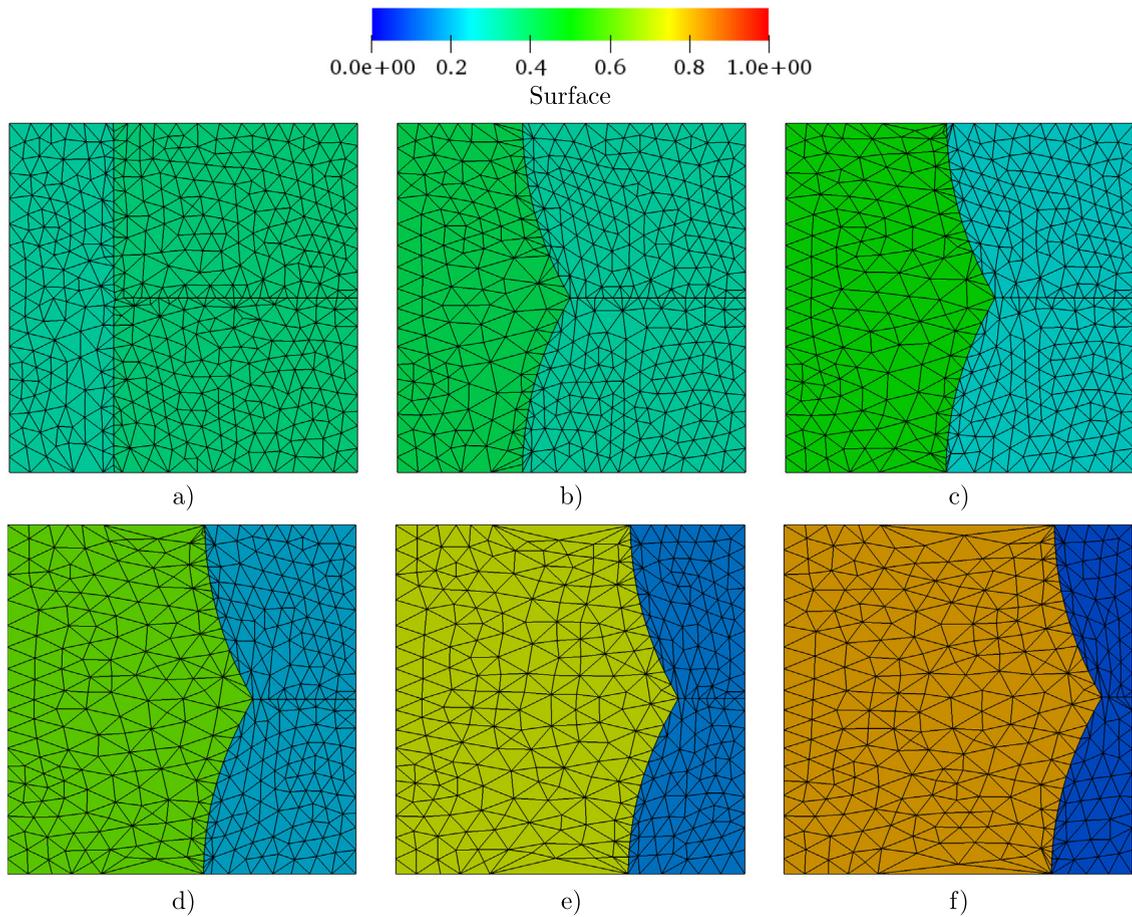


Fig. 28. Example of the evolution of the T-junction test using the TRM method and a parameter of $h_{trm} = 0.006$, the mesh and the surface field per phase are displayed. (a) Initial state, (b) state at $t = 0.1$, (c) state at $t = 0.2$ (d) state at $t = 0.3$, (e) state at $t = 0.4$, (f) state at $t = 0.5$.

size of $h = 0.004$ mm at the interfaces, additionally, statistical comparisons with the response obtained by a FE-LS approach as presented in [1,3,20,45] 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 a reference case will be used to compare the different models, this reference is obtained after a convergence analysis over the SM case having the mesh size as a parameter in the same way as in [7] (the resulting mesh of the reference case employing a mesh size $h = 0.001$ mm)

The initial microstructure along with its thermomechanical properties for this test have been detailed in [7], the generation of the initial tessellation have been realized thanks to a Laguerre–Voronoi cells generation procedure [46–48] over a squared domain of surface $A = 10$ mm² 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 $M_0 = 1.56 \cdot 10^{11}$ mm⁴/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/mm²) [2,3].

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. 33, here a comparison with the SM model has been given. A detailed view of the same comparison can be found in Fig. 34 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

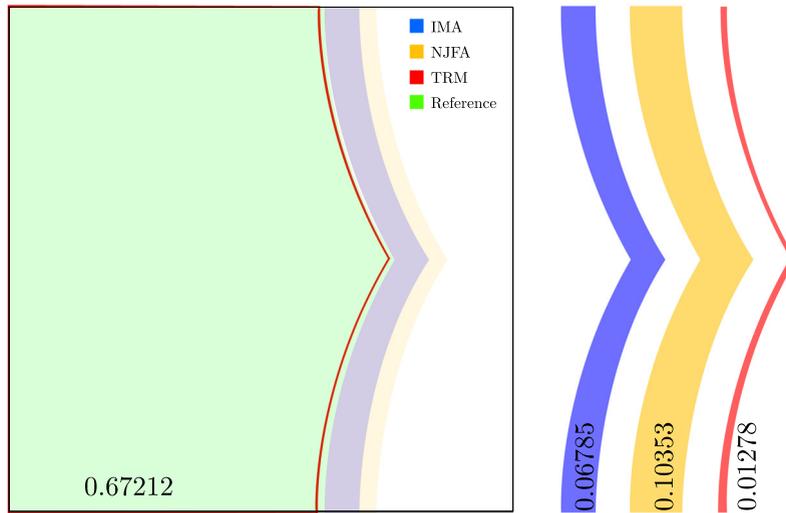


Fig. 29. Difference between the growing phase of the T-junction test case for the reference model and the geometric difference of the same phase obtained with the other models (the IMA and NFJA) at time $t = 0.35$. Values for the area of each section are given.

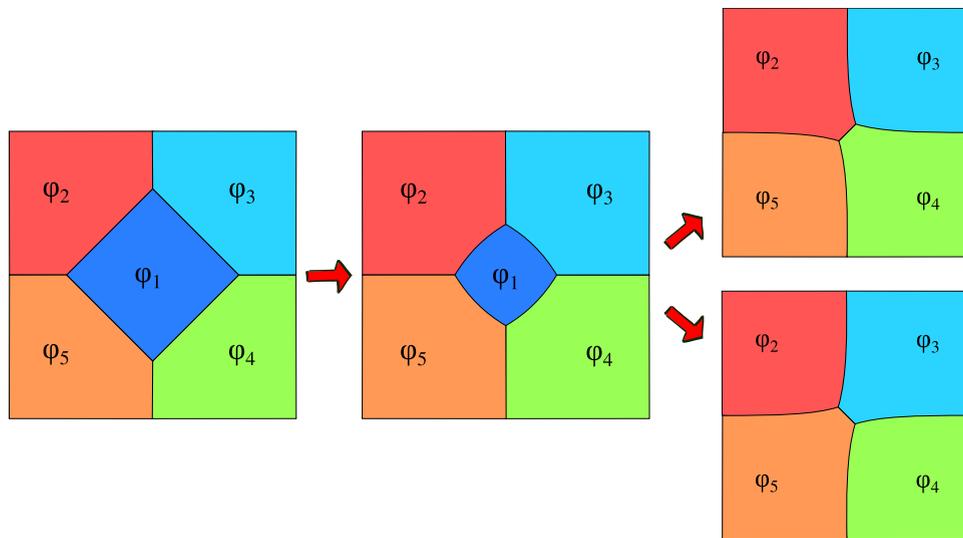


Fig. 30. Square-Shrinkage. left: initial state, center: square shrinking, right: the square disappears and a new interface is made ($\phi_2 - \phi_4$ or $\phi_3 - \phi_5$) [7].

different moments during the simulation, while a very small difference on the grain sizes values (see the color distribution in Fig. 33(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 Figs. 35 and 36 respectively. Here the comparison with the AMA and the IMA and the reference case 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 [7] to the reference case.

The CPU-time evolution over time for the different models can be found in 37, where the TRM model has been the one with the lowest computational cost, performing 14,8 times faster that the SM model and 156 times faster than the AMA case. These results are very promising as the ratio accuracy — CPU-time of the TRM model shows a significant improvement against the other compared models.

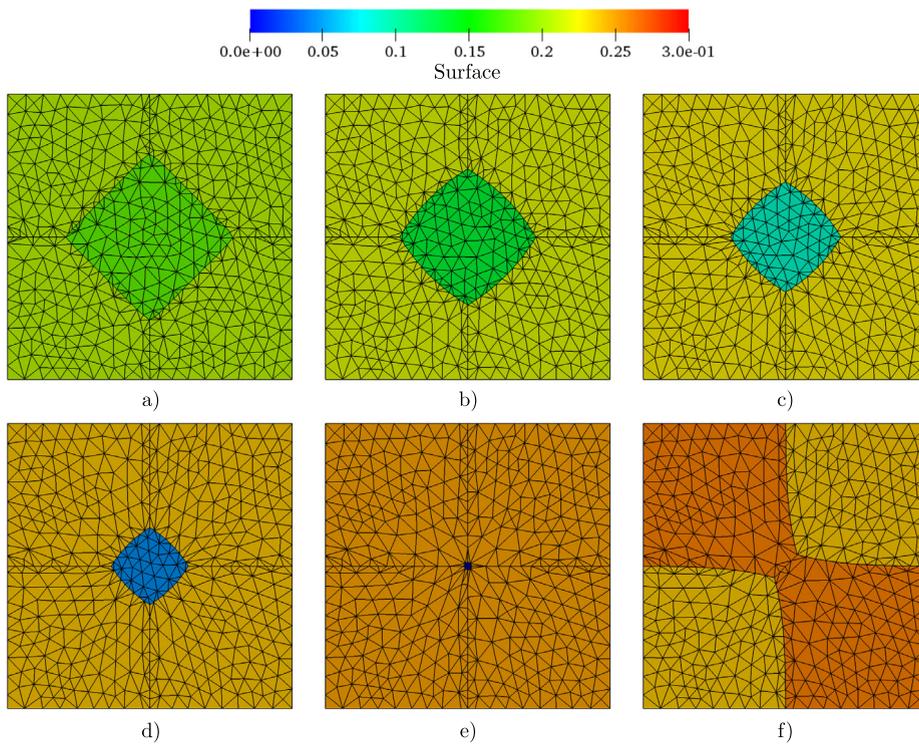


Fig. 31. Example of the evolution of the square shrinkage test using the TRM method and a parameter of $h_{trm} = 0.006$, the mesh and the surface field per phase are displayed. (a) Initial state, (b) state at $t = 0.0225$, (c) state at $t = 0.045$ (d) state at $t = 0.0675$, (e) state at $t = 0.09$, (f) state at $t = 0.1125$.

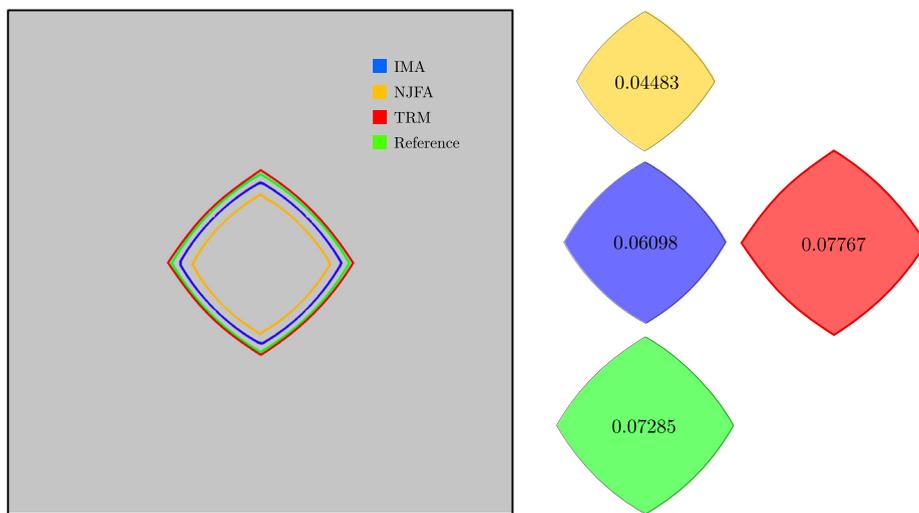


Fig. 32. Square-Shrinkage. Comparison of the ϕ_1 phase at $t = 0.05$ (left side). Values for the area of each domain are given (right side).

Finally, the mobility term M can be fitted in order to minimize the difference between the curves of 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. Figs. 38 and 39 show 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:

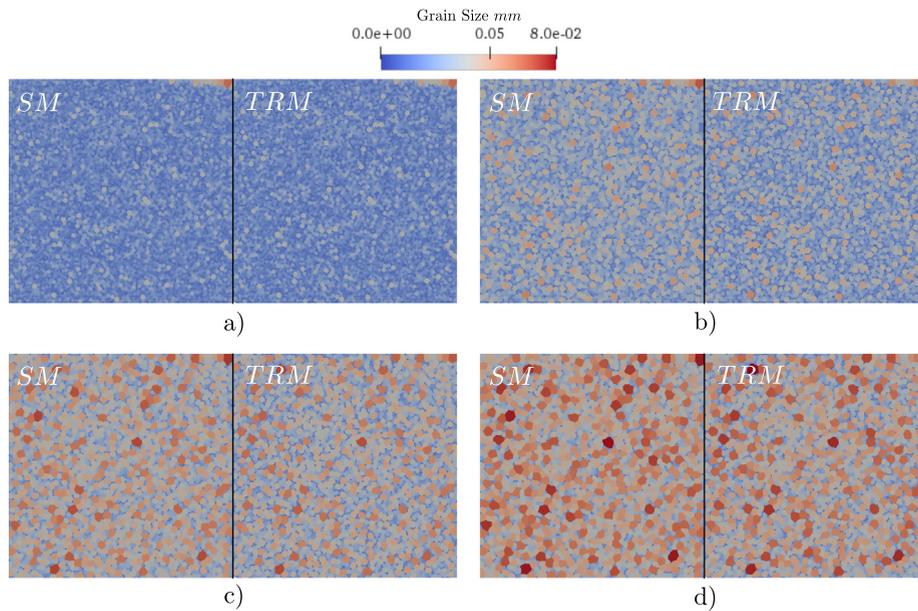


Fig. 33. 2D 10000 grains case. Comparison of the evolution of the TRM model (right) and the SM model (left). (a) Initial state (b) state at $t = 840$ s (c) state at $t = 1940$ s (d) state at $t = 2780$ s.

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.

5. Discussion, conclusion and perspectives

The development of the TRM method has implemented multiple concepts: the initial LS-TRM preprocessor and geometrical reconstruction has 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 a piece-wise third degree polynomial (natural splines) and to remesh using a 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 aid 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 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 [7]).

A significance improvement on 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, Figs. 35 and 36 illustrated the difference on 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, model with a 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.

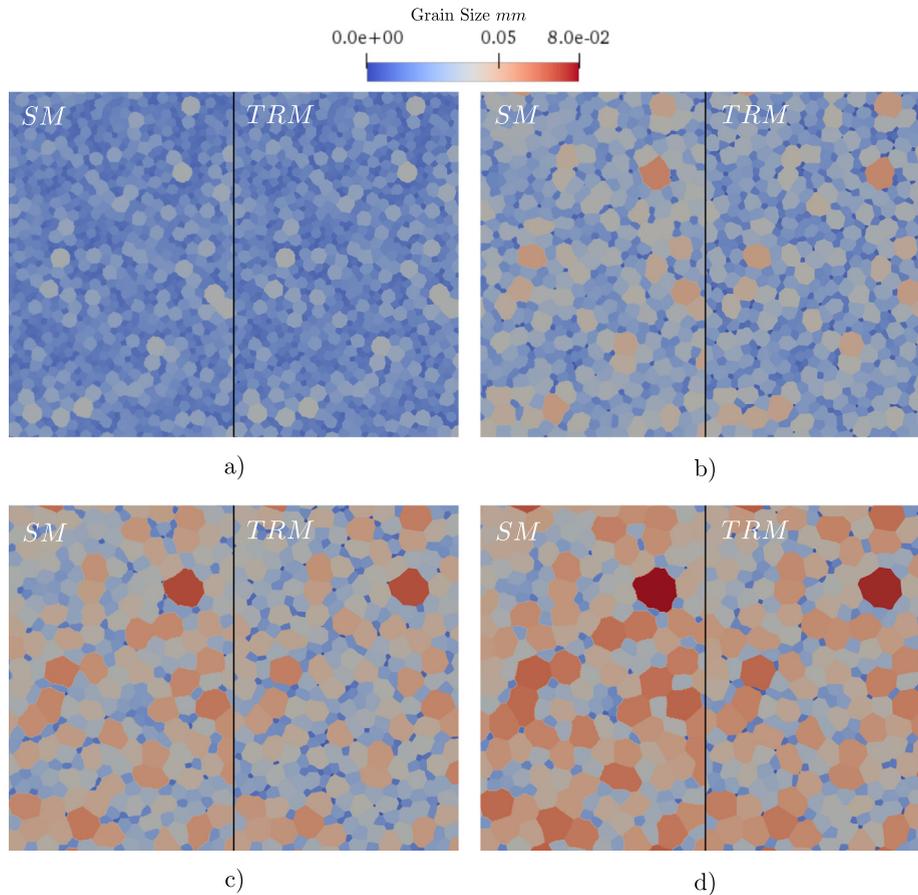


Fig. 34. 2D 10000 grains case. detailed view of the evolution of the TRM model (right) and the SM model (left). (a) Initial state (b) state at $t = 840$ s (c) state at $t = 1940$ s (d) state at $t = 2780$ s.

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 ($M = 8.27549 \cdot 10^{-07}$ mm⁴/Js) was corrected in -12.8% for the SM model ($M_{sm} = M * (1 - 0.128) = 7,21623 \cdot 10^{-07}$ mm⁴/Js) and of $+6.69\%$ for the TRM model ($M_{trm} = M * (1 + 0.0669) = 8,829120 \cdot 10^{-07}$ mm⁴/Js). Figs. 39 and 38 illustrate the evolution of the mean grain size and the grain size distribution respectively after the correction, a 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 grain growth, the TRM approach can be applied to simulate mechanism such as 2D Recrystallization (ReX) or 2D microstructural evolutions taking into account anisotropic grain boundary properties. Further efforts will be made in order to integrate these mechanisms to the TRM model as well as the development of the TRM model in 3D.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

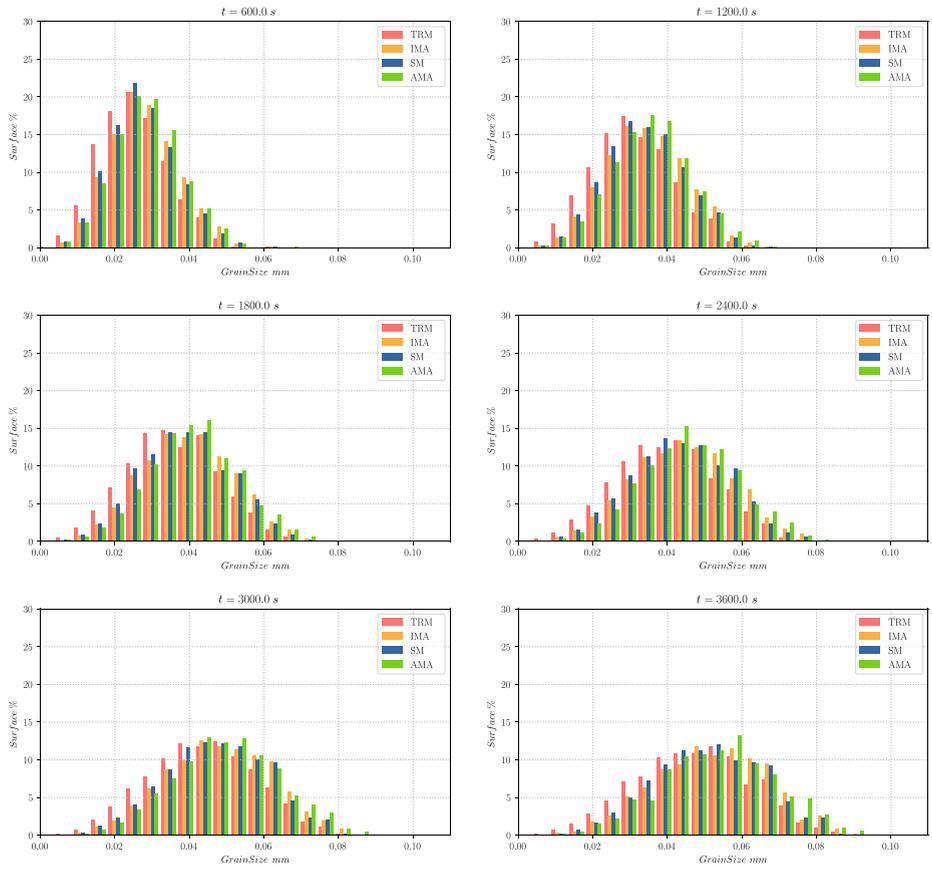


Fig. 35. Evolution of the grain size distribution pondered by surface for the different models. The TRM model appears to be slower than the others.

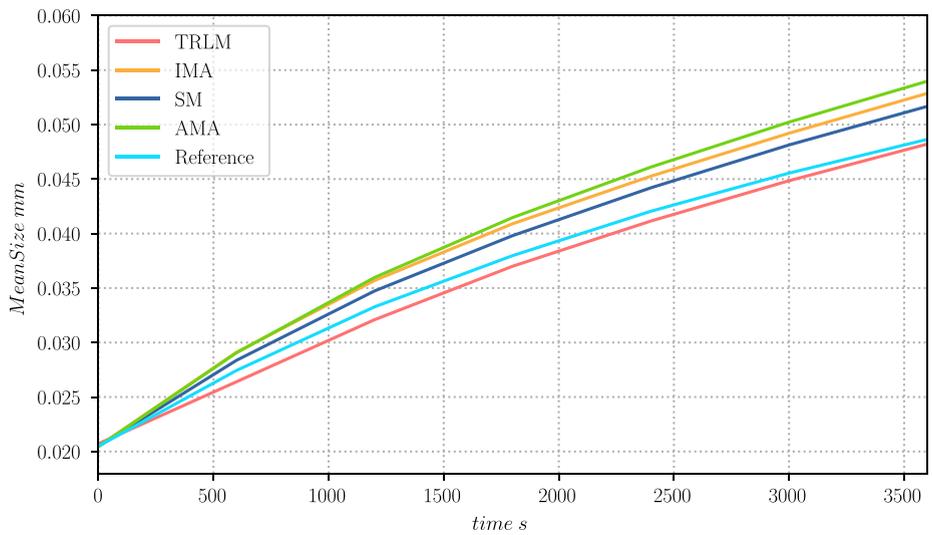


Fig. 36. Grain size evolution for the 10000 grains test case. The TRM model appears to be slower than the others.

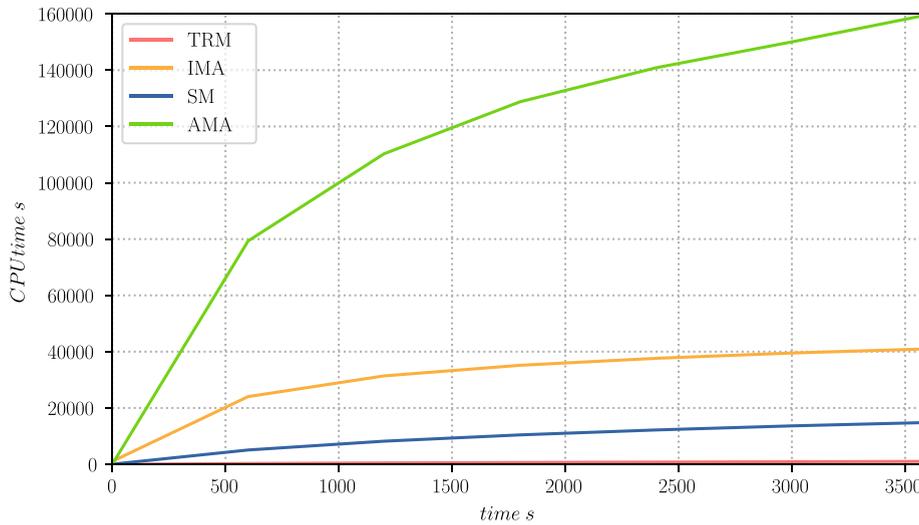


Fig. 37. CPU-time comparison for the 10000 grains test case. the TRM model was 14,8 times faster that the SM model and 156 times faster than the AMA case.

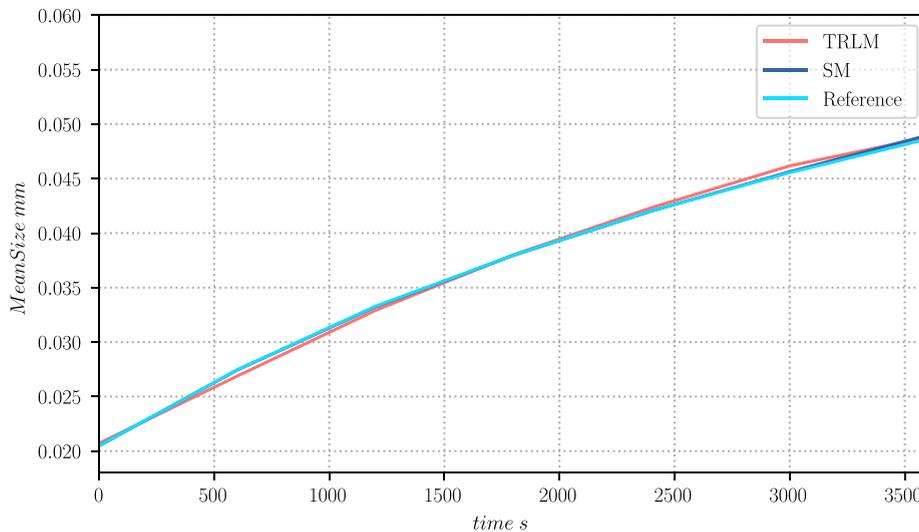


Fig. 38. Grain size evolution for the 10000 grains test case after a correction on the Mobility M of -12.8% and $+6.69\%$ for the SM model and the TRM model respectively. All curves are superposed. This result illustrates a well known behavior of full field simulations of GG: the reduced mobility is classically impacted by the choice of the numerical method and is not only a universal physical parameter.

Data availability

The raw data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study. The processed data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study.

Acknowledgments

The authors thank the ArcelorMittal, ASCOMETAL, Aubert & Duval, CEA, FRAMATOME, SAFRAN, TIMET, CONSTELLIUM and TRANSVALOR companies and the ANR for their financial support through the DIGIMU consortium, France and ANR, France industrial Chair (Grant No. ANR-16-CHIN-0001).

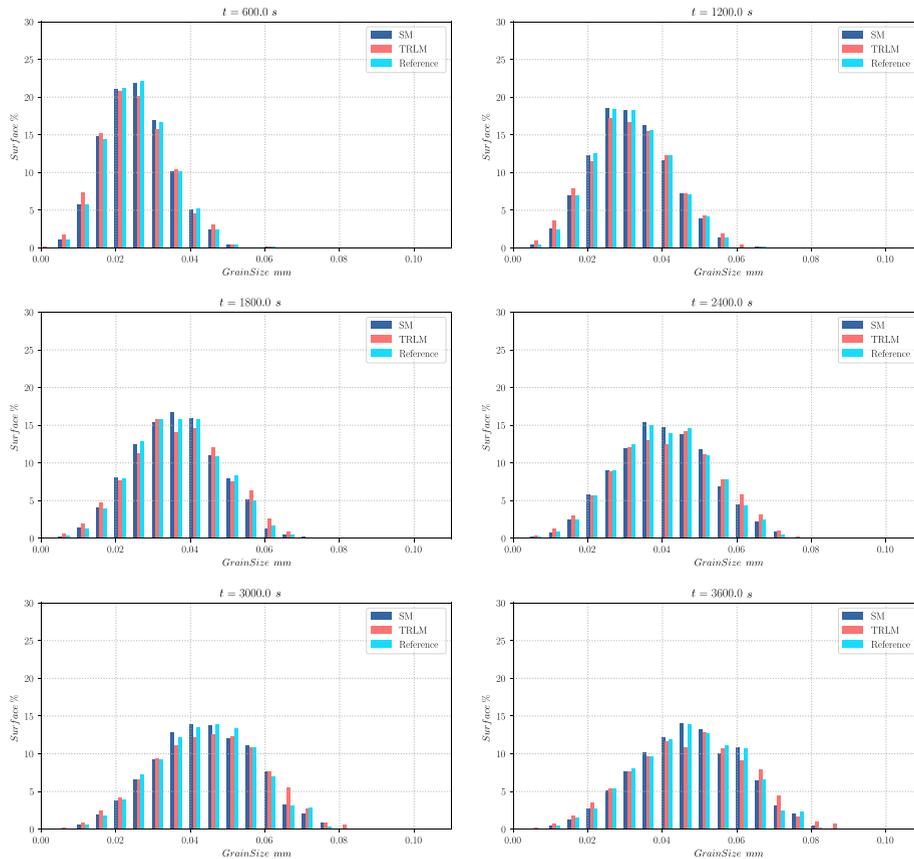


Fig. 39. Evolution of the grain size distribution pondered by surface after a correction on the Mobility M of -12.8% and $+6.69\%$ for the SM model and the TRM model respectively.

References

- [1] M. Bernacki, Y. Chastel, T. Coupez, R.E. Logé, Level set framework for the numerical modelling of primary recrystallization in polycrystalline materials, *Scr. Mater.* 58 (12) (2008) 1129–1132, <http://dx.doi.org/10.1016/j.scriptamat.2008.02.016>.
- [2] B. Scholtes, R. Boulais-Sinou, A. Settefrati, D. Pino Muñoz, I. Poitault, A. Montouchet, N. Bozzolo, M. Bernacki, 3D level set modeling of static recrystallization considering stored energy fields, *Comput. Mater. Sci.* 122 (2016) 57–71, <http://dx.doi.org/10.1016/j.commatsci.2016.04.045>.
- [3] L. Maire, B. Scholtes, C. Moussa, N. Bozzolo, D.P. Muñoz, A. Settefrati, M. Bernacki, Modeling of dynamic and post-dynamic recrystallization by coupling a full field approach to phenomenological laws, *Mater. Des.* 133 (2017) 498–519, <http://dx.doi.org/10.1016/j.matdes.2017.08.015>.
- [4] H. Hallberg, A modified level set approach to 2D modeling of dynamic recrystallization, *Modelling Simulation Mater. Sci. Eng.* 21 (8) (2013) 1–33, <http://dx.doi.org/10.1088/0965-0393/21/8/085012>.
- [5] H. Hallberg, V.V. Bulatov, Modeling of grain growth under fully anisotropic grain boundary energy, *Modelling Simulation Mater. Sci. Eng.* 27 (045002) (2019) 1–27, <http://dx.doi.org/10.1088/1361-651X/ab0c6c>.
- [6] B. Scholtes, Development of an efficient level set framework for the full field modeling of recrystallization in 3D (Ph.D. thesis), MINES ParisTech, 2016, <http://dx.doi.org/10.1017/jfm.2015.566>, arXiv:1410.7235.
- [7] S. Florez, M. Shakoor, T. Toulorge, M. Bernacki, A new finite element strategy to simulate microstructural evolutions, *Comput. Mater. Sci.* 172 (2020) 109335, <http://dx.doi.org/10.1016/J.COMMATSCI.2019.109335>, URL <https://www.sciencedirect.com/science/article/pii/S0927025619306342?dgcid=author>.
- [8] R.I. Saye, J.A. Sethian, The voronoi implicit interface method for computing multiphase physics, *Proc. Natl. Acad. Sci.* 108 (49) (2011) 19498–19503, <http://dx.doi.org/10.1073/pnas.1111557108>.
- [9] R.I. Saye, J.A. Sethian, Analysis and applications of the voronoi implicit interface method, *J. Comput. Phys.* 231 (18) (2012) 6051–6085, <http://dx.doi.org/10.1016/j.jcp.2012.04.004>.
- [10] K. Kawasaki, T. Nagai, K. Nakashima, Vertex models for two-dimensional grain growth, *Phil. Mag. B* 60 (3) (1989) 399–421, <http://dx.doi.org/10.1080/13642818908205916>, URL <https://www.tandfonline.com/doi/full/10.1080/13642818908205916>.

- [11] D. Weygand, Y. Bréchet, J. Lépinoux, A vertex dynamics simulation of grain growth in two dimensions, *Phil. Mag. B* 78 (4) (1998) 329–352, <http://dx.doi.org/10.1080/13642819808206731>, URL <https://www.tandfonline.com/doi/full/10.1080/13642819808206731>.
- [12] J. Lépinoux, D. Weygand, M. Verdier, Modeling grain growth and related phenomena with vertex dynamics, *C. R. Phys.* 11 (3–4) (2010) 265–273, <http://dx.doi.org/10.1016/j.crhy.2010.07.015>, URL <https://linkinghub.elsevier.com/retrieve/pii/S1631070510000800>.
- [13] L.A. Barrales Mora, 2D vertex modeling for the simulation of grain growth and related phenomena, *Math. Comput. Simulation* 80 (7) (2010) 1411–1427, <http://dx.doi.org/10.1016/j.matcom.2009.08.005>.
- [14] Y. Mellbin, H. Hallberg, M. Ristinmaa, A combined crystal plasticity and graph-based vertex model of dynamic recrystallization at large deformations, *Modelling Simulation Mater. Sci. Eng.* 23 (4) (2015) 045011, <http://dx.doi.org/10.1088/0965-0393/23/4/045011>.
- [15] K.A. Brakke, The surface evolver, *Experiment. Math.* 1 (2) (1992) 141–165, <http://dx.doi.org/10.1080/10586458.1992.10504253>.
- [16] G. Couturier, C. Maurice, R. Fortunier, Three-dimensional finite-element simulation of Zener pinning dynamics, *Phil. Mag.* 83 (30) (2003) 3387–3405, <http://dx.doi.org/10.1080/1478643031000152771>.
- [17] G. Couturier, C. Maurice, R. Fortunier, R. Doherty, J.H. Driver, Finite element simulations of 3D Zener pinning, in: *Materials Science Forum*, Vol. 467–470, 2004, pp. 1009–1018, <http://dx.doi.org/10.4028/www.scientific.net/msf.467-470.1009>.
- [18] G. Couturier, R. Doherty, C. Maurice, R. Fortunier, 3D finite element simulation of the inhibition of normal grain growth by particles, *Acta Mater.* 53 (4) (2005) 977–989, <http://dx.doi.org/10.1016/j.actamat.2004.10.044>.
- [19] J.K. Becker, P.D. Bons, M.W. Jessell, A new front-tracking method to model anisotropic grain and phase boundary motion in rocks, *Comput. Geosci.* 34 (2008) 201–212, <http://dx.doi.org/10.1016/j.cageo.2007.03.013>.
- [20] A.L. Cruz-Fabiano, R. Logé, M. Bernacki, Assessment of simplified 2D grain growth models from numerical experiments based on a level set framework, *Comput. Mater. Sci.* 92 (2014) 305–312, <http://dx.doi.org/10.1016/j.commatsci.2014.05.060>.
- [21] L. Maire, B. Scholtes, C. Moussa, D. Pino Muñoz, N. Bozzolo, M. Bernacki, Improvement of 3-D mean field models for pure grain growth based on full field simulations, *J. Mater. Sci.* 51 (24) (2016) 10970–10981.
- [22] J. Furstoss, M. Bernacki, C. Ganino, C. Petit, D. Pino-Muñoz, 2D and 3D simulation of grain growth in olivine aggregates using a full field model based on the level set method, *Phys. Earth Planet. Inter.* 283 (2018) 98–109.
- [23] J.C.J.C. Cantrell, *Geometric Topology*, Academic Press, 1979, p. 698.
- [24] B. Scholtes, M. Shakoov, A. Settefrati, P.-O. Bouchard, N. Bozzolo, M. Bernacki, New finite element developments for the full field modeling of microstructural evolutions using the level-set method, *Comput. Mater. Sci.* 109 (2015) 388–398, <http://dx.doi.org/10.1016/j.commatsci.2015.07.042>.
- [25] M. Shakoov, M. Bernacki, P.-O.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, *Eng. Fract. Mech.* 147 (2015) 398–417, <http://dx.doi.org/10.1002/nme.2788>.
- [26] M. Shakoov, P.-O. Bouchard, M. Bernacki, An adaptive level-set method with enhanced volume conservation for simulations in multiphase domains, *Internat. J. Numer. Methods Engrg.* 109 (4) (2017) 555–576, <http://dx.doi.org/10.1002/nme.5297>.
- [27] C. Dapogny, C. Dobrzynski, P. Frey, Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems, *J. Comput. Phys.* 262 (2014) 358–378, <http://dx.doi.org/10.1016/j.jcp.2014.01.005>.
- [28] C. de Boor, A practical guide to splines, in: *Applied Mathematical Sciences*, vol. 27, (2) Springer New York, New York, NY, 1978, pp. 999–1002, <http://dx.doi.org/10.1007/978-1-4612-6333-3>, URL <http://link.springer.com/10.1007/978-1-4612-6333-3>.
- [29] M. Amirfakhrian, H. Mafikandi, Approximation of parametric curves by moving least squares method, *Appl. Math. Comput.* 283 (2016) 290–298, <http://dx.doi.org/10.1016/j.amc.2016.02.039>.
- [30] Q. Wang, W. Zhou, Y. Cheng, G. Ma, X. Chang, Y. Miao, E. Chen, Regularized moving least-square method and regularized improved interpolating moving least-square method with nonsingular moment matrices, *Appl. Math. Comput.* 325 (2018) 120–145, <http://dx.doi.org/10.1016/j.amc.2017.12.017>.
- [31] G. Compère, J.F. Remacle, E. Marchandise, Transient mesh adaptivity with large rigid-body displacements, in: *Proceedings of the 17th International Meshing Roundtable, IMR 2008, (iMMC) 2008*, pp. 213–230, <http://dx.doi.org/10.1007/978-3-540-87921-3-13>.
- [32] R. Rangarajan, A.J. Lew, G. Compère, J.-F.J.-F. Remacle, J. Jansson, J. Hoffman, G. Compère, J.-F.J.-F. Remacle, J. Jansson, J. Hoffman, A mesh adaptation framework for dealing with large deforming meshes, *Internat. J. Numer. Methods Engrg.* 82 (7) (2009) 843–867, <http://dx.doi.org/10.1002/nme.2788>, arXiv:1010.1724. URL <http://doi.wiley.com/10.1002/nme.2788> <http://arxiv.org/abs/1201.4903>.
- [33] R. Loubère, P.H. Maire, M. Shashkov, J. Breil, S. Galera, Reale: A reconnection-based arbitrary-Lagrangian-eulerian method, *J. Comput. Phys.* 229 (12) (2010) 4724–4761, <http://dx.doi.org/10.1016/j.jcp.2010.03.011>.
- [34] F. Alauzet, A changing-topology moving mesh technique for large displacements, *Eng. Comput.* 30 (2) (2014) 175–200, <http://dx.doi.org/10.1007/s00366-013-0340-z>.
- [35] 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 Simulation Mater. Sci. Eng.* 22 (8) (2014) 085005, <http://dx.doi.org/10.1088/0965-0393/22/8/085005>.
- [36] M. Bernacki, H. Resk, T. Coupez, R.E. Logé, Finite element model of primary recrystallization in polycrystalline aggregates using a level set framework, *Modelling Simulation Mater. Sci. Eng.* 17 (6) (2009) 064006, <http://dx.doi.org/10.1088/0965-0393/17/6/064006>.
- [37] M. Bernacki, R.E. Logé, T. Coupez, Level set framework for the finite-element modelling of recrystallization and grain growth in polycrystalline materials, *Scr. Mater.* 64 (6) (2011) 525–528, <http://dx.doi.org/10.1016/j.scriptamat.2010.11.032>.
- [38] D. Weygand, Y. Bréchet, J. Lépinoux, Zener pinning and grain growth: a two-dimensional vertex computer simulation, *Acta Mater.* 47 (3) (1999) 961–970, [http://dx.doi.org/10.1016/S1359-6454\(98\)00383-8](http://dx.doi.org/10.1016/S1359-6454(98)00383-8), URL <https://linkinghub.elsevier.com/retrieve/pii/S1359645498003838>.

- [39] J. Fausty, N. Bozzolo, D. Pino Muñoz, M. Bernacki, A novel level-set finite element formulation for grain growth with heterogeneous grain boundary energies, *Mater. Des.* 160 (2018) 578–590, <http://dx.doi.org/10.1016/J.MATDES.2018.09.050>, URL <https://www.sciencedirect.com/science/article/pii/S0264127518307524>.
- [40] J.A. Sethian, A fast marching level set method for monotonically advancing fronts., *Proc. Natl. Acad. Sci.* 93 (4) (1996) 1591–1595, <http://dx.doi.org/10.1073/pnas.93.4.1591>, URL <http://www.pnas.org/cgi/doi/10.1073/pnas.93.4.1591>.
- [41] R. Kimmel, J.A. Sethian, Computing geodesic paths on manifolds, *Proc. Natl. Acad. Sci.* 95 (15) (1998) 8431–8435, <http://dx.doi.org/10.1073/pnas.95.15.8431>.
- [42] J.A. Sethian, A. Vladimirsky, Fast methods for the eikonal and related Hamilton- Jacobi equations on unstructured meshes, *Proc. Natl. Acad. Sci.* 97 (11) (2000) 5699–5703, <http://dx.doi.org/10.1073/pnas.090060097>.
- [43] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* 114 (1) (1994) 146–159, <http://dx.doi.org/10.1006/JCPH.1994.1155>, URL <https://www.sciencedirect.com/science/article/pii/S0021999184711557?via%3Dihub>.
- [44] C. Herring, Surface tension as a motivation for sintering, in: *Fundamental Contributions to the Continuum Theory of Evolving Phase Interfaces in Solids*, Springer, 1999, pp. 33–69.
- [45] R. Logé, M. Bernacki, H. Resk, L. Delannay, H. Dignonnet, Y. Chastel, T. Coupez, Linking plastic deformation to recrystallization in metals using digital microstructures, *Phil. Mag.* 88 (30–32) (2008) 3691–3712, <http://dx.doi.org/10.1080/14786430802502575>.
- [46] H. Imai, M. Iri, K. Murota, Voronoi diagram in the Laguerre geometry and its applications, *SIAM J. Comput.* 14 (1) (1985) 93–105.
- [47] K. Hitti, P. Laure, T. Coupez, L. Silva, M. Bernacki, Precise generation of complex statistical representative volume elements (RVEs) in a finite element context, *Comput. Mater. Sci.* 61 (2012) 224–238, <http://dx.doi.org/10.1016/j.commatsci.2012.04.011>.
- [48] D.N. Ilin, M. Bernacki, Advancing layer algorithm of dense ellipse packing for generating statistically equivalent polygonal structures, *Granul. Matter* 18 (3) (2016) 43, <http://dx.doi.org/10.1007/s10035-016-0646-9>.