A novel 3D anisotropic Voronoi microstructure generator with an advanced spatial discretization scheme

Citation for published version (APA):

van Nuland, T. F. W., Palmeira Belotti, L., van Dommelen, J. A. W., & Geers, M. G. D. (2021). A novel 3D anisotropic Voronoi microstructure generator with an advanced spatial discretization scheme. *Modelling and Simulation in Materials Science and Engineering*, 29(8), [085014]. <https://doi.org/10.1088/1361-651X/ac32b3>

DOI:

[10.1088/1361-651X/ac32b3](https://doi.org/10.1088/1361-651X/ac32b3)

Document status and date:

Published: 10/11/2021

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

PAPER • OPEN ACCESS

A novel 3D anisotropic Voronoi microstructure generator with an advanced spatial discretization scheme

To cite this article: T F W van Nuland *et al* 2021 *Modelling Simul. Mater. Sci. Eng.* **29** 085014

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

You may also like

- [Characterization of iron ferromagnetism by the local atomic volume: from three-dimensional structures to isolated atoms](#)
Lei Zhang, M Šob, Zhe Wu et al.
- [How well proportioned are lens and prism spaces?](#)
R Aurich and S Lustig
- [Statistical topology of perturbed two-dimensional lattices](#)
Hannes Leipold, Emanuel A Lazar, Kenneth A Brakke et al.

IOP | ebooks™

Bringing together innovative digital publishing with leading authors from the global scientific community.

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

A novel 3D anisotropic Voronoi microstructure generator with an advanced spatial discretization scheme

T F W van Nuland^{ID}, L Palmeira Belotti,
J A W van Dommelen^{*ID} and M G D Geers

Department of Mechanical Engineering, Eindhoven University of Technology, PO
Box 513, 5600 MB, Eindhoven, The Netherlands

E-mail: J.A.W.v.Dommelen@tue.nl

Received 30 June 2021, revised 13 October 2021

Accepted for publication 22 October 2021

Published 10 November 2021

CrossMark

Abstract

At the microstructural scale, Voronoi tessellations are commonly used to represent a polycrystalline morphology. However, due to spherical growth of nuclei, an anisotropic tessellation with spatially varying elongated grain directions, which is present in many applications, cannot be obtained. In this work, a novel 3D anisotropic Voronoi algorithm is presented, together with its implementation and two application cases. The proposed algorithm takes into account preferred grain growth directions, aspect ratios and sizes in the definition of an ellipsoidal growth velocity field defined per grain. For applications in which a predetermined mesh is used, e.g. voxel-mesh based simulations, the grains are extracted in a straight-forward manner. In cases where a fully grain conforming discretization is desired, e.g. finite element simulations, a hexahedral mesh generator is incorporated to arrive at a discretization which can be directly used in microstructural modeling simulations. Two application cases are studied (a wire + arc additively manufactured and a magnesium alloy microstructure) in which the algorithm's capability for curved, non-convex, periodic domains is shown. Furthermore, the resulting grain morphology is compared to experimental data in terms of grain size, grain aspect ratio and grain columnar direction distribution. In both cases, the algorithm adequately produces a representative volume element with convincing representativeness of the experimental data. The 3D anisotropic Voronoi algorithm is highly versatile in a wide range of application

* Author to whom any correspondence should be addressed.

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

cases, specifically suitable for the generation of polycrystalline microstructures that include grains with spatially varying elongated directions.

Keywords: microstructure generation, Voronoi algorithm, anisotropic tessellation, grain morphology, representative volume element, wire + arc additive manufacturing, 3D hexahedral finite element mesh

(Some figures may appear in colour only in the online journal)

1. Introduction

In many engineering applications, the macroscopic properties of materials are predicted based on detailed modeling of the underlying microstructure. Being able to generate specific geometrical features of the microstructure is thereby an important ingredient. Voronoi tessellations are frequently used to represent grain shapes of (metal) polycrystalline materials at the microstructural scale [1–3]. Such tessellations are generated starting from a set of grain seed points, which represent the grain nuclei. In the Voronoi tessellation, a spherical growth from these seed points outward, is assumed [4]. The interface where two growth fronts meet, represents a grain boundary. As a result, a specific grain is defined by the volume that is closer to the corresponding grain seed point than to all other grain seed points.

For homogeneously distributed grain seed points, an isotropic grain morphology is obtained, due to the assumption of spherical growth. However, for many applications, this does not properly represent the actual microstructure. Therefore, many extensions to the standard Voronoi algorithm have been used to improve the microstructural description. This includes e.g. hard-sphere Voronoi [5, 6], Laguerre–Voronoi [7, 8] or the Johnson–Mell tessellation [9, 10]. Using such algorithms, grain size distributions, curved and concave-shaped grains can be incorporated. However, all these extensions still rely on an isotropic (spherical) growth speed. This is not representative for applications in which columnar grains with (spatially varying) columnar directions are present. This typically applies to microstructures obtained by additive manufacturing [11, 12], casting [13], welding [14], etc, i.e. methods which have a ‘directional’ driving force. For the modeling of such microstructures, the standard Voronoi approach would not reproduce the correct grain morphology since a spherical growth velocity field is not appropriate.

In previous work [15], an anisotropic Voronoi algorithm was developed which generates two-dimensional computational microstructures based on an elliptical grain growth velocity field to incorporate spatially varying grain growth directions. The output of this algorithm consists of (convex) grain vertices and connectivity. Finite element discretization of the geometry is applied afterwards. In this work, the algorithm is extended to three dimensions by considering an ellipsoidal grain growth velocity field and by including the mesh generation step. This enables the generation of concave shaped grains, increasing the representativeness of the generated microstructure.

Various standard element types are generally available in mesh generation algorithms and in commercial finite element packages. Although in 3D, tetrahedral meshes, based on Delaunay triangulation [16], are generally easier to generate compared to hexahedral meshes, tetrahedral meshes also bring serious disadvantages [17]. Especially, tetrahedral meshes tend to show volumetric locking (artificial stiffening) when used in finite element simulations of (nearly) incompressible materials, such as based on crystal plasticity, requiring the use of e.g. non-standard tetrahedral elements [18]. Therefore, in this work, specific attention is given to the

generation of an all-hexahedral based mesh, which can be directly used for microstructural simulations.

In general, a distinction can be made between geometry-first and mesh-first finite element discretization procedures [19]. In the former, the surface of the domain is used as a starting point for the mesh to develop inwards. The latter uses a simple (or even regular) mesh that covers the entire domain of interest. Then, this initial mesh is modified locally to capture specific features [20]. The 3D anisotropic Voronoi algorithm generates grains and a final grain-conforming mesh is essential to make sure no spurious stress concentrations arise at staggered, non-grain-conforming boundaries [21]. Hence, a mesh-first generation procedure is adopted in the current work.

This paper is organized as follows. In section 2, the mathematical definition of the ellipsoidal grain growth velocity field is specified. Then, in section 3, the numerical implementation of the algorithm is detailed. Furthermore, the 3D anisotropic Voronoi algorithm is applied to two cases in section 4, along with an analysis of the generated microstructure correspondence to input data. Finally, conclusions are drawn in section 5.

2. Mathematical background

The definition of a grain C_i is based on the time t_i required to grow from the grain seed point $\vec{x}_i$ to an arbitrary point in the domain of interest $\vec{y}$, with respect to a Cartesian coordinate system $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$, as

$$C_i = \left\{ \vec{y} \in \mathbb{R}^n, t_i(\vec{y}) \leq t_j(\vec{y}), \forall j \neq i \right\}, \quad (1)$$

in which $i, j = 1, \dots, N$, with N the number of grain seed points. The time t_i is obtained from

$$t_i(\vec{y}) = \frac{d(\vec{y}, \vec{x}_i)}{v_i(\vec{y})}, \quad (2)$$

in which $d(\vec{a}, \vec{b}) \equiv \|\vec{a} - \vec{b}\|$ is the Euclidean distance function and $v_i(\vec{y})$ is the (scalar) growth speed in the direction $\vec{y} - \vec{x}_i$.

In order to take into account the preferred growth directions of the grains, an ellipsoidal velocity field $v_i(\vec{y})$ is now introduced to describe the anisotropic growth speed of each grain. Given grain seed point i , an arbitrary ellipsoidal velocity field can be defined by the volume of the ellipsoid V_i and two preferred growth vectors, $\vec{p}_i$ and $\vec{q}_i$, which are perpendicular to each other (i.e. $\vec{p}_i \cdot \vec{q}_i = 0$). The directions and lengths of these vectors define the principal axes and corresponding aspect ratios of the ellipsoid, respectively, as defined in appendix A and as depicted in figure 1. The resulting description of the directional growth speed v_i can be expressed in terms of the directional unit vector $\vec{e}_r = \vec{e}_r(\vec{y}) = \frac{\vec{y} - \vec{x}_i}{\|\vec{y} - \vec{x}_i\|}$ as

$$v_i = s_i \hat{r}_3(\vec{e}_r), \quad (3)$$

in which the scaling factor s_i is defined by

$$s_i = \sqrt[3]{\frac{3}{4} \frac{V_i}{\pi \|\vec{p}_i\| \|\vec{q}_i\|}} \quad (4)$$

and $\hat{r}_3$ is the dimensionless radius of the 3D ellipsoid defined as

Figure 1. Schematic representation of the 3D ellipsoidal growth velocity field. The highlighted blue cross-section represents the plane in which $\vec{e}_r^*$ is defined.

$$\hat{r}_3(\vec{e}_r) = \left[\frac{1}{[\hat{r}_2(\vec{e}_r^*)]^2} + \left(\frac{1}{\|\vec{q}_i\|^2} - \frac{1}{[\hat{r}_2(\vec{e}_r^*)]^2} \right) (\vec{e}_q \cdot \vec{e}_r)^2 \right]^{-\frac{1}{2}}, \quad (5)$$

with

$$\hat{r}_2(\vec{e}_r^*) = \left[1 + \left(\frac{1}{\|\vec{p}_i\|^2} - 1 \right) (\vec{e}_p \cdot \vec{e}_r^*)^2 \right]^{-\frac{1}{2}}. \quad (6)$$

The derivation of these equations is given in appendix A. In these equations, the principal directions $\vec{e}_p = \frac{\vec{p}_i}{\|\vec{p}_i\|}$ and $\vec{e}_q = \frac{\vec{q}_i}{\|\vec{q}_i\|}$ are used. Furthermore, $\hat{r}_2$ represents the dimensionless radius of the 2D elliptical cross-section of the 3D ellipsoid in the plane through the center and orthogonal to $\vec{q}_i$, which is highlighted in figure 1. Consequently, $\vec{e}_r^*$ is the normalized projection of $\vec{e}_r$ onto this plane, i.e. $\vec{e}_r^* = \frac{\vec{e}_r - \vec{e}_r \cdot \vec{e}_q \vec{e}_q}{\|\vec{e}_r - \vec{e}_r \cdot \vec{e}_q \vec{e}_q\|}$. Note that in case the anisotropic growth field is spheroidal (i.e. an ellipsoid with two axes of the same length), the ellipsoid is rotationally symmetric around its single preferred growth direction $\vec{p}_i$ and the description of the velocity field v_i reduces to

$$v_i = s_i \hat{r}_2(\vec{e}_r), \quad \text{with} \quad s_i = \sqrt[3]{\frac{3}{4} \frac{V_i}{\pi \|\vec{p}_i\|}}. \quad (7)$$

3. Algorithm for generating anisotropic computational microstructures

3.1. Input

The input of the algorithm consists of (i) an initial hexahedral mesh, covering the domain of interest Ω , (ii) a set of N_b domain boundary functions $f_1(\vec{y}), \dots, f_{N_b}(\vec{y})$ and their gradients $\vec{\nabla} f_1(\vec{y}), \dots, \vec{\nabla} f_{N_b}(\vec{y})$ whose definition will be provided later, (iii) the seed point coordinates $\vec{x}_1, \dots, \vec{x}_N$, (iv) the corresponding preferred growth direction vectors $\vec{p}_1, \dots, \vec{p}_N$ (including optional second preferred growth vectors $\vec{q}_1, \dots, \vec{q}_N$) and (v) the corresponding volumes of

the ellipsoids $V_1, \dots, V_N$. The initial hexahedral mesh can be a regular voxel mesh, but this is not required; it can also be an irregular hexahedral mesh which has been generated with standard hexahedral finite element discretization algorithms in case the domain of interest is more complex. Together with the initial hexahedral mesh, domain boundary functions $f_k(\vec{y})$ should be provided. Each of these functions $f_k(\vec{y})$ are defined to be zero at the k th face of the domain, Γ_k , reading

$$\begin{cases} f_k(\vec{y}) = 0 & \forall \vec{y} \in \Gamma_k, \\ f_k(\vec{y}) \neq 0 & \forall \vec{y} \notin \Gamma_k. \end{cases} \quad (8)$$

For example, signed, unsigned or squared distance functions fulfill these conditions. The use of these boundary functions is twofold. First, for each node p in the mesh with position $\vec{y}_p$, let K_p contain the set of domain boundary face numbers k on which this node is located, i.e.

$$K_p = \{k, |f_k(\vec{y}_p)| < \epsilon\}, \quad (9)$$

with ϵ a given tolerance, and define $B_p = \#K_p$ as the number of boundaries in the set K_p . The boundary functions can then be used to classify nodes in the mesh based on their location, i.e. either in the interior ($B_p = 0$), on a domain face ($B_p = 1$), on a domain edge ($B_p = 2$) or at a domain vertex ($B_p = 3$). Second, the boundary functions are used to displace nodes on particular faces while ensuring that these nodes remain on the corresponding faces. The remaining input consists of data $(\vec{x}_i, \vec{p}_i, \vec{q}_i$ and $V_i)$ representing the characteristics of each grain. This can be based on experimental observations or statistics of the (morphology of the) microstructure. In order to represent a specific part of a microstructure, it is also possible to extract the required input from (3D) electron backscatter diffraction (EBSD) data by fitting ellipsoids to grains. One can also obtain the preferred growth direction vector $\vec{e}_p$ as the largest temperature gradient from e.g. a thermal process simulation, since crystals tend to grow in that direction.

For the elaboration of the detailed numerical implementation, the example shown in figure 2 is considered. The initial hexahedral mesh consists of a regular voxel mesh of $10 \times 10 \times 5$ elements having a total size of $1 \times 1 \times \frac{1}{2}$. The corresponding six signed distance boundary functions (one for each face of the domain of interest depicted in figure 2) are given as

$$\begin{aligned} f_1(\vec{y}) &= \vec{y} \cdot \vec{e}_1, \\ f_2(\vec{y}) &= \vec{y} \cdot \vec{e}_2, \\ f_3(\vec{y}) &= \vec{y} \cdot \vec{e}_3, \\ f_4(\vec{y}) &= 1 - \vec{y} \cdot \vec{e}_1, \\ f_5(\vec{y}) &= 1 - \vec{y} \cdot \vec{e}_2, \\ f_6(\vec{y}) &= \frac{1}{2} - \vec{y} \cdot \vec{e}_3. \end{aligned} \quad (10)$$

In section 3.3, this choice is motivated in more detail. Furthermore, the five considered grain seed points along with their ellipsoidal velocity growth fields have been depicted in figure 2 as well.

3.2. Assignment of elements to grains

The first part of the algorithm is to assign each element to a specific grain. This is done in two steps. First, for each element e the time required to grow from each grain seed point i to that

Figure 2. Illustration of the example case input.

element is determined using equation (2), in which $\vec{y} = \vec{y}_e$ is the center of element e . Based on these growth times $t_1(\vec{y}_e), \dots, t_N(\vec{y}_e)$, the grain seed points are sorted from minimum to maximum growth time. Preferably, each element e is eventually assigned to the first grain seed point in this list. For the example case, the preferred assignment of elements to grains is shown in figure 3(a), in which the color of the element corresponds to the seed point velocity field in figure 2. Furthermore, the light-shaded color of the elements corresponds to the preferred grain seed point assignment (which is the result of the current, first step) and a dark-shaded element corresponds to the actually assigned grain seed point (which will be detailed in the second step). By considering the red and purple grain in figure 3(a), it is clear how the anisotropic, ellipsoidal growth speed of the grains is taken into account. In this first step, interactions between grains are not taken into account. Consequently, a grain may be split into multiple, unconnected parts by another grain. In the example case (figure 3(a)), the yellow grain consists of two unconnected parts (it is also not connected in the interior). However, the grain could not have grown from the seed point on the right to the left part, since growth would be blocked by the purple grain in between. Because the grain seed point is on the right, the left part will be referred to as an ‘island’. The left part is thus physically not representative and unacceptable. Therefore, the preferred assignment of grains cannot be accepted directly.

In the second step of the assignment of elements to grains, it is ensured that two important aspects are not present in the final grain connectivity: grain islands and non-manifold meshes. The former has been discussed in the previous paragraph and the importance of the latter will be discussed in section 3.3. An element e is identified as part of an island when there is no path of neighboring elements assigned to the same grain that connects e to the element containing the corresponding seed point. Here, and in the rest of the detailing of the algorithm, elements are only defined as neighbors when they share a common face. This is achieved by iteratively accepting the preferred assignment of grains. During the first iteration, the elements that contain a grain seed point are accepted if their preferred assignment corresponds to the same grain seed point. This means that in case two or more seed points are located within the same element, only the grain seed point that grows to the center of that element first, is accepted. In figure 3(b), the state of the example case is shown after the first iteration. Again, the light-shaded color of the elements corresponds to the color of the preferred grain seed point assignment and a dark-shaded element corresponds to the actually assigned grain seed point. During all subsequent iterations, the preferred assignment of element e to grain i is accepted when (i) at least one neighboring element has already been accepted and assigned to the same grain i as element e and (ii) it does not introduce a non-manifold condition in the

Figure 3. Stages of the acceptance algorithm for the assignment of elements to grains: (a) before the start; (b)–(e) after the 1st, 2nd, 3rd and 4th iteration, respectively; (f) after the 18th iteration; (g) before the 19th iteration; (h) and (i) after the 22nd iteration; (j) after finishing. In (i) only the elements corresponding to the blue grain are shown. Note, the light-shaded color of the elements corresponds to the color of the preferred grain seed point assignment in figure 2. A dark-shaded element corresponds to the actually assigned grain seed point.

connectivity of grain i . A non-manifold condition is obtained when the connectivity of a given mesh contains a feature that cannot exist in the ‘real’ world, e.g. a position where elements are only connected via a single vertex or edge. In order to identify non-manifold grain connectivity, the Euler number E , defined as

$$E = N_{\text{nodes}} - N_{\text{edges}} + N_{\text{faces}}, \quad (11)$$

is utilized. Given a 3D hexahedral mesh with N_{nodes} boundary nodes, N_{edges} boundary edges and N_{faces} boundary faces, the Euler number E must be equal to 2 in order to be free of non-manifold conditions [22]. For each node p of element e , the Euler number $E_{p,i}^e$ is calculated for the set of elements that share node p and are assigned to grain i , including element e . Furthermore, the Euler number $E_{p,-i}^e$ (considering the set of elements that share node p and are not assigned to grain i , thus excluding element e) is also computed. If for any p it holds that $E_{p,i}^e \neq 2$ or $E_{p,-i}^e \neq 2$, a non-manifold condition would be introduced upon accepting the preferred assignment of grain i to element e , which is therefore rejected. In figures 3(c)–(e), the state of the example case is visualized for iterations 2–4, respectively.

After a number of iterations, it may occur that no new elements are accepted in the last iteration. However, it is possible that there are still elements that have not been accepted yet. This is the case in figure 3(f), where the state of the example case is depicted after 18 iterations. In the next iterations, elements are also allowed to be assigned to their second preferred grain seed point, which, for the example case, have been indicated in figure 3(g). A number of iterations later, only a single element, as depicted in figure 3(h), has not yet been accepted by the algorithm. This is because accepting this grain seed point assignment would introduce a non-manifold condition in the connectivity of the blue grain (it is not an island since it can be reached from below the element, see figure 3(i) in which only the elements corresponding to the blue grain are depicted). Therefore, in the final grain connectivity shown in figure 3(j), this element has been assigned to the closest grain seed point that does not introduce an island or non-manifold condition, which is the purple grain. Using this algorithm, the final grain connectivity does not contain of any grain islands or non-manifold conditions.

3.3. Converting to a grain-conforming mesh

For specific applications that rely on element regularity, e.g. using an fast Fourier transform (FFT) solver, the resulting voxel-based mesh can be used directly. However, in many other applications, e.g. when using crystal plasticity and a finite element solver, a non-grain-conforming mesh is undesirable as it can introduce spurious stress concentrations [23]. Therefore, the next step is to convert the mesh to a grain-conforming discretization. This is accomplished using four steps, which are detailed in the following.

First, so-called pillow elements are inserted at the grain boundary interface as proposed by Owen *et al* [22]. Therefore, faces shared by two elements that have been assigned to two different grain seed points are extruded to both sides of the interface. Each grain boundary face thus introduces two new hexahedral elements, effectively surrounding each grain by a single layer of new elements, hence the name pillow elements. In figure 4(a), the resulting mesh is shown for the example case. A new node p^* corresponding to grain i is obtained by (duplicating and) displacing its parent node on the grain boundary p by a distance d in the local grain boundary inward normal direction $\vec{n}_p$. This normal vector can be obtained by averaging the inward normals of the grain boundary element faces that share node p . Notice that for this step it is crucial to have a mesh free of non-manifolds. By using the local grain boundary normal direction (and not for example the direction vector pointing to the geometric center of the grain), concave shaped grains can also be handled by the algorithm, as e.g. illustrated by the green or blue grain in figure 4(a). The pillow size d can be chosen arbitrarily (small), however, it must be smaller than half the local mesh size. Otherwise, the nodes of an element with pillowing on opposite faces would overlap, which directly introduces an invalid element. For visualization purposes, in figure 4(a), $d = 0.02$, which is $\frac{1}{5}$ of the mesh size. The main purpose of this step is to ensure that each element has maximum one face on a grain boundary interface. Consequently, the grain conformity is automatically addressed in the element connectivity.

During the second step, the element connectivity is not changed, solely the nodes are displaced by smoothing the mesh. Therefore, the well-known Laplacian mesh smoothing operation is applied [24]. In the algorithm, the domain vertex nodes are fixed. These nodes are not displaced during the entire algorithm, as it changes the domain geometry. The Laplacian smoothing step is applied first to nodes on a domain edge. The new positions of these nodes on the boundary are only influenced by the domain vertices and domain edge nodes. After this smoothing step, the domain edge nodes are temporarily fixed and the domain face nodes

Figure 4. Steps involved in the conversion to a grain conforming mesh. The state of the example case is visualized after (a) step 1: volume element insertion, (b) step 2: Laplacian mesh smoothing, (c) step 3: surface element insertion and (d) step 4: Laplacian mesh smoothing.

are displaced by the smoothing step. Finally, the interior nodes are displaced. In the Laplacian mesh smoothing operation, the new nodal position $\vec{y}_p^*$ depends on old nodal positions of a set of neighboring nodes $\vec{y}_l$ according to

$$\vec{y}_p^* = \frac{1}{N_l} \sum_{l \in L_p} \vec{y}_l, \quad (12)$$

in which N_l is the number of local nodes in the set L_p . Furthermore, a distinction is thus made between the handling of a given node p which is classified as domain vertex ($B_p = 3$), domain edge node ($B_p = 2$), domain face node ($B_p = 1$) or interior node ($B_p = 0$), as discussed before. Let A_p be all adjacent nodes of node p , then L_p is defined as

$$L_p = \{n \in A_p, B_n \geq B_p\}. \quad (13)$$

By performing the smoothing operation in this manner, there are no special constraints needed for the displacement of nodes at the domain boundary if the domain shape is cuboid, which is the case for the example domain. However, in case of non-straight boundaries, a gradient descend algorithm is used to keep the nodes on the corresponding external boundaries. Therefore, the residual R_p of node p is defined as

$$R_p = \sum_{k \in K_p} f_k(\vec{y}_p^*), \quad (14)$$

which vanishes when the node is located on the corresponding external boundary. The iterative update of nodal positions is then given by

$$\delta \vec{y}_p^* = - \frac{R_p}{(\vec{\nabla} R_p \cdot \vec{e}_R)} \vec{e}_R, \quad (15)$$

with

$$\vec{e}_R = \frac{\vec{\nabla} R_p}{\|\vec{\nabla} R_p\|} \quad (16)$$

the unit vector in the direction of steepest ascend. This algorithm is iterated until the residual R_p drops below a given tolerance ϵ . Notice that in theory the residual R_p in equation (14) can also vanish if two boundary functions cancel each other. If this is the case for a specific application, the boundary functions $f_k(\vec{y})$ can easily be adapted to represent the squared

Figure 5. Section of the example case after (a) step 2 and (b) step 4 of the conversion to a grain conforming mesh. All elements belonging to the red and green grain are depicted. The blue line indicates the internal junction of the red, green and blue grain (of which the latter is not shown). Furthermore, the yellow circle highlights the influence of step 3 and 4 on the grain conformity of the mesh.

or unsigned distance to a domain boundary face, leading to $f_k(\vec{y}) \geq 0, \forall \vec{y}$. However, it should be noted that for these boundary functions, the unit vector in the direction of steepest ascend $\vec{e}_R$, see equation (16), becomes inaccurate at the domain boundary. This is the reason signed distance boundary functions are currently adopted.

Due to these local operations, the element quality of the pillow elements increases. The entire smoothing operation is iterated five times, which yields a good balance between obtaining a smooth grain conforming mesh and preserving the main grain shape features. In figure 4(b), the example case mesh is shown after the smoothing operations have been applied.

In figure 5(a), only the red and green grain of the example case are shown. Here, in the highlighted yellow region, an element with two edges on an internal junction between three grains (the red, green and blue grain, indicated by the blue line) is identified. Such elements are undesirable since smooth grain boundaries at these locations would introduce highly distorted or even invalid elements. Therefore, in the third step of the conversion to a grain conforming mesh, it is ensured that each element has maximum one edge on an internal junction between three or more grains. This is accomplished by another pillowing step (which is referred to as the surface pillowing step), in which a pillow layer is introduced around the set of elements that have one of their faces on a given internal grain boundary face. The rest of the pillowing operation is identical to the first step, which has been explained before. For the example case, this leads to the mesh as visualized in figure 4(c).

The fourth and final step of the conversion to a grain conforming mesh is the same as the second step. The mesh is smoothened locally to increase the mesh quality of the new surface pillow elements and their neighbors. Figure 5(b) reveals that after the third and fourth step, each element has maximum one edge on a junction between three or more grains, allowing to obtain a fully grain conforming mesh, which, for the example case, is shown in figure 4(d).

3.4. Increasing mesh quality

For the example case, the mesh is now complete. It does not contain any invalid or highly distorted elements and the discretization can directly be used in microstructural simulations. However, for more complex cases (including the application cases described in section 4), it is necessary to increase the mesh quality or even eliminate inverted hexahedral elements. Therefore, the algorithm described in [25], which is specifically applicable to hexahedral meshes, is adopted. This algorithm deploys a nodal quality metric based on a normalized Jacobian. Given a hexahedral element e with node p and neighboring element nodes A , B and C (which are also a vertex of the same element e), as visualized in figure 6, let $\vec{a}$, $\vec{b}$ and $\vec{c}$ denote the

Figure 6. Hexahedral element with node p and corresponding neighbors A , B and C . Note that the ordering of nodes A , B and C with respect to node p matters. Adapted from [25].

edge vectors from node p to nodes A , B and C , respectively. Note that the ordering of these edge vectors is defined such that the volume $J = \vec{a} \cdot (\vec{b} \times \vec{c})$ spanned by these vectors is positive for a well-defined hexahedral element e . The quality Q_p^e of node p with respect to element e is then given by

$$Q_p^e = \hat{\vec{a}} \cdot (\hat{\vec{b}} \times \hat{\vec{c}}), \quad (17)$$

in which

$$\hat{\vec{a}} = \frac{\vec{a}}{L_{\max}}, \quad \hat{\vec{b}} = \frac{\vec{b}}{L_{\max}}, \quad \hat{\vec{c}} = \frac{\vec{c}}{L_{\max}}, \quad (18)$$

with $L_{\max}$ the maximum local element edge length, i.e.

$$L_{\max} = \max \{ \|\vec{a}\|, \|\vec{b}\|, \|\vec{c}\| \}. \quad (19)$$

By defining the quality Q_p^e in this manner, not only the orthogonality of element e matters, but also its aspect ratio. Furthermore, the quality Q_p of node p can now be defined as

$$Q_p = \min_e Q_p^e, \quad (20)$$

representing the minimum quality Q_p^e of all elements e that share node p . Finally, the mesh quality Q_{mesh} is defined as

$$Q_{\text{mesh}} = \min_p Q_p. \quad (21)$$

An invalid mesh can then be identified as $Q_{\text{mesh}} \leq 0$, indicating that the discretization contains at least one element with a negative volume. Furthermore, the mesh quality is considered poor if Q_{mesh} is smaller than a given minimum mesh quality $Q_{\min}$.

In order to improve the mesh quality Q_{mesh} , nodes are displaced. In general, multiple iterations are required to reach the minimum mesh quality $Q_{\min}$. The condition to accept a new nodal position $\vec{y}_p^*$ is therefore based on a positive iterative difference in quality ΔQ_p (and not

on the absolute value of the nodal quality Q_p). Furthermore, it is required that the new nodal position $\vec{y}_p^*$ does not decrease the quality of any of its adjacent nodes A_p , i.e.

$$\Delta Q_p > 0 \quad \wedge \quad \Delta Q_a \geq 0 \quad \forall a \in A_p. \quad (22)$$

The nodes are sorted based on their quality Q_p and the node with the lowest quality is improved first. The algorithm uses two techniques to increase the nodal quality of node p , which are the same as described by Calvo and Idelsohn [25]. For the sake of completeness, they are briefly described below:

(a) Gradient driven.

The update of the nodal position $\Delta \vec{y}_p$ is in the direction of the gradient of Q_p with a step size equal to a fraction ξ of the average local mesh size $L_{\text{avg}} = \frac{1}{3}(\|\vec{a}\| + \|\vec{b}\| + \|\vec{c}\|)$, i.e.

$$\Delta \vec{y}_p = \xi L_{\text{avg}} \frac{\vec{\nabla} Q_p}{\|\vec{\nabla} Q_p\|}. \quad (23)$$

The fraction ξ is iteratively decreased from a maximum of $\xi = 1$ to a minimum of $\xi = \frac{1}{128}$, each iteration halving the fraction. The first nodal displacement $\Delta \vec{y}_p$ to be accepted (i.e. condition (22) is fulfilled), is applied.

(b) Randomized.

In case the first technique did not result in an accepted nodal displacement, the second technique is applied. Here a random vector on the unit sphere $\vec{e}_s$ is adopted as the direction vector, and a random fraction $0 < \xi < 1$ defines the step size, leading to

$$\Delta \vec{y}_p = \xi L_{\text{avg}} \vec{e}_s. \quad (24)$$

The number of repetitions of taking a random direction vector $\vec{e}_s$ and a random fraction ξ is linearly scaled with the quality Q_p , between 50 for $Q_p \leq 0$ and 0 for $Q_p = 1$. Here, also, the first nodal displacement $\Delta \vec{y}_p$ to fulfill condition (22), is applied.

In this manner, nodes in the interior of the domain ($B_p = 0$) are updated. Nodes on a domain boundary face ($B_p = 1$) require extra consideration. These are kept on the corresponding boundary by applying the same gradient descend algorithm as described in section 3.3. Naturally, the evaluation of the new nodal position $\vec{y}_p^*$ is performed after convergence of the gradient descend iterations. Furthermore, it has been observed that it is not required for the nodes on a domain boundary edge ($B_p = 2$) to be moved, to obtain a valid mesh.

After the nodes are displaced individually, the new mesh quality Q_{mesh} is calculated. This entire procedure is performed iteratively until a given minimum mesh quality Q_{min} is reached. In order to speed up this mesh optimization algorithm, a node p for which $Q_p > 2Q_{\text{min}}$ is skipped in the displacement update, since its update has a negligible effect on the new mesh quality Q_{mesh} . Nodes p with a quality $Q_{\text{min}} < Q_p < 2Q_{\text{min}}$ are still updated using this procedure, since this update typically allows for more freedom of neighboring nodes with a lower quality.

The mesh optimization strategy as discussed is applied twice in the 3D anisotropic Voronoi algorithm, after both step 2 and step 4 of the conversion to a grain conforming geometry. It is thus used as an extra optimization step after the Laplacian mesh smoothing operation to ensure a good final mesh quality. As explained before, at the end of step 2 the element connectivity is not yet suitable for smooth internal junctions between three or more grains. Therefore, in the first mesh optimization step, nodes which are located on such edges are not taken into account for the determination of the mesh quality Q_{mesh} .

3.5. Generation of a periodic mesh

In micromechanical modeling applications, it is often desired to have a periodic discretization. This can also be achieved with the 3D anisotropic Voronoi algorithm using the general approach of generating periodic computational microstructures. However, a few important considerations should be highlighted.

First, the initial mesh is required to be periodic. In the algorithm, nodes from the initial mesh on opposite domain boundary faces, edges and vertices are identified and tied together. This also holds for the newly generated nodes during the pillowing operations. The ties are applied specifically after the Laplacian mesh smoothing operations and during the mesh quality improvement procedure. Here, it is also taken into account in the acceptance of a new nodal position.

Secondly, all grain seed points are copied $3 \times 3 \times 3 = 27$ times in order to generate a periodic geometry in the center, but the initial mesh is not replicated to save computational costs. In the iterative acceptance of the assignment of elements to grains (section 3.2), the periodicity is taken into account in the definition of neighboring elements. Furthermore, in order for the pillowing operations to be periodic (and to introduce the new nodes periodically), the intersection line of a given grain boundary with a domain boundary face should be at the same location on opposite periodic boundary faces. Therefore, it is important that two periodic neighboring elements sharing a common element face on an opposite periodic domain boundary are assigned to the same grain. Equivalently, a set of elements sharing an edge (or vertex) on corresponding periodic domain edges (or corners), are assigned to the same grain. These considerations ensure that a periodic hexahedral mesh is created.

4. Application

The algorithm to generate 3D anisotropic Voronoi microstructures along with their finite element discretization can be applied to generate computational microstructures in a variety of fields, and is specifically suitable to polycrystalline microstructures in which grains with spatially varying elongated shape directions are a typical feature. This includes additive manufacturing [12, 26, 27], twinning [28, 29], casting [13], welding [30, 31], (dynamic) recrystallization [32], etc. In this section, the algorithm is applied to two application cases. In the first case, the algorithm's capability of generating a statistically representative volume element (RVE) is demonstrated for a unit cell with non-straight periodic boundaries. In the second application case, 3D EBSD data is directly used as input for the algorithm to generate a mesh that can be used to model a specific part of the microstructure. In both cases, grains with spatially varying elongated directions are characteristic features of the selected microstructures.

4.1. Wire + arc additive manufacturing

The first application case consists of an AISI 316LSi stainless steel microstructure which has been produced using wire + arc additive manufacturing (WAAM). This process is similar to multi-pass fusion welding. By stacking weld lines on top of each other, a large-scale metallic component (~ 1 to >10 m) can be built up. Locally, an electric arc is used to melt the material during deposition. When this melt pool solidifies, a new bead of solid metal results. During solidification, crystals tend to grow in the direction of the local temperature gradient, which is strongly present during the manufacturing process. This promotes the growth of strongly columnar grains and, due to the melt pool shape, the columnar direction is spatially varying.

Electron backscatter diffraction (EBSD) data of a typical grain structure for austenitic stainless steels produced by the WAAM process obtained from [33] is depicted for two

Figure 7. EBSD data of the WAAM microstructure in two cross-sectional planes: (a) deposition direction—building direction (6.5×6.5 mm) and (b) transverse direction—building direction [33]. In (b) four different scans of the same cross-section are shown, which all have a size of 4.0×2.4 mm, which is approximately one periodic unit. Colors indicate crystallographic directions in the building direction.

Figure 8. Four stages of the 3D anisotropic Voronoi algorithm during the computational WAAM microstructure generation: (a) input and domain of interest, (b) initial mesh and assignment of elements to grains, (c) final geometry and (d) final hexahedral mesh. The domain of interest has a periodic size of $3.85 \times 2.34 \times 0.50$ mm. Colors correspond to grain orientations with respect to the building direction.

cross-sectional planes: figure 7(a) shows the deposition direction—building direction plane and figure 7(b) displays the transverse direction—building direction plane, which have been generated using MTEX [34]. Here, the deposition direction is the direction in which the electric arc moves, the building direction is the vertical stacking direction of layers and the transverse direction is the direction of neighboring beads in a single layer. In order to generate a 3D statistically representative and periodic computational microstructure, the 3D anisotropic Voronoi algorithm is applied. The domain of interest and corresponding boundary functions are acquired from the repeated stacking of weld lines. An experimentally obtained average 2D periodic fusion zone shape [35], which is non-convex and contains curved boundaries, is extruded in the third dimension to account for about four grains in the deposition direction, which is based on the average grain thickness obtained from figure 7(a). The resulting domain edges are visualized in figure 8(a) by black lines.

The grain seed points used as input for the algorithm are obtained from the EBSD data displayed in figure 7(b). For this purpose, first, grains are identified from the EBSD data by a misorientation threshold of 5° . Ellipses are fitted to the grains by equalizing their second moments [36]. The ellipse centroids obtained from the four regions in figure 7(b) are aligned according to their corresponding experimental fusion zone boundaries estimated from the EBSD data and stacked upon each other in the third (deposition) direction. The ellipses are symmetric around their long axes to form a spheroid. This assumes that the short axis of the grain in the transverse direction—building direction cross-section is the same as the thickness of the grain in the deposition direction, which is supported by the EBSD data shown in figure 7(a). Finally, a random normal distribution with a standard deviation of 10% of the layer height is applied to the grain seed point coordinates in the third (deposition) direction. This then results in the final grain seed point input for the algorithm as displayed in figure 8(a). Here, each spheroid represents the velocity field of an individual grain seed point. Their color represents the grain orientation with respect to the building direction, which is obtained from the corresponding grain in the EBSD data.

As mentioned before, the domain of interest and corresponding boundary functions are acquired from the repeated stacking of weld lines. The experimentally obtained average 2D periodic fusion zone shape [35] is discretized into quad elements and subsequently extruded to form 10 layers of hexahedral elements in the out-of-plane (deposition) direction. Although this is a relatively coarse initial discretization in the deposition direction, in the final mesh, each grain consists of at least 5 elements in their smallest dimension, due to the pillow element insertion step. The initial mesh consists of 54 100 hexahedral elements and 61 171 nodes as shown in figure 8(b). Here, the state of the algorithm after the first step (assignment of elements to grains) is illustrated by the element coloring.

The final computational geometry and mesh, generated by the 3D anisotropic Voronoi algorithm, are depicted in figures 8(c) and (d), respectively. The former shows a 3D computational microstructure consisting of smooth grain boundaries and strongly columnar grains with their long axes corresponding to the experimental data shown in figure 7(b). The latter consists of a periodic hexahedral and grain conforming mesh (381 616 elements and 415 839 nodes) that can be used directly in e.g. finite element crystal plasticity simulations.

In order to analyze the representativeness of the computational microstructure with respect to the experimental data, the same cross-sections on which the EBSD data (figure 7) is gathered, are obtained from the final computational microstructure (figure 8(c)). These cross-sections are shown in figure 9, in which the color represents the grain orientation in the building (vertical) direction. During the following analysis, the periodicity of the microstructure is taken into account. Therefore, grains that continue over the domain boundary are treated as a single grain. This leads to the grain shapes illustrated in figure 9 by black grain boundaries. In these figures, the light-shaded grains in the background serve to emphasize the periodicity of the microstructure and these are not taken into account in the analysis.

A comparison is made in terms of the grain size distribution and columnar grain direction and aspect ratio, for both cross-sectional planes, to evaluate the correspondence between the EBSD data and the resulting computational microstructure. First, focus is put on the deposition direction—building direction plane. Figure 10(a) shows the total percentage of area (cumulative grain size) that is covered by grains having a certain grain size or smaller in this cross-section. Depending on the choice of misorientation threshold used in the definition of a sub-grain ($\sim 0.5^\circ$) or a grain ($\sim 5^\circ$) in the EBSD data, a range of grain size distributions can be obtained. This is illustrated by the gray shaded area representing misorientations between 0.5° – 5° as indicated. Obviously, the resulting computational microstructure adequately represents the experimental grain size distribution. The grain columnar direction and aspect ratio

Figure 9. Periodic cross-sections of the computational WAAM microstructure (figure 8(c)) corresponding to the same planes as used for the EBSD data (figure 7): (a) deposition direction—building direction and (b) transverse direction—building direction. The cross-section in (a) is taken at the location indicated by the vertical dash-dot line in (b). The solid black lines indicate grain boundaries of the grains used in the analysis and the light-shaded background grains illustrate the microstructural periodicity. This is also emphasized by the dashed line, indicating the periodic domain boundary. Colors indicate crystallographic directions in the building direction.

are displayed in figure 11(a), where each dot represents a single grain. Both quantities are defined with respect to a fitted ellipse: the former defines the orientation of the major axis and the latter specifies the ratio between the major and minor axis length. Also here, the resulting computational microstructure represents the EBSD data appropriately. Notice that the large grain size variations in the experimental data, see figure 10(a), are not obtained with the computational microstructure due to the limited thickness of the domain and the small number of grains in the deposition direction. Also, due to the relatively large EBSD scan with many grains in this cross-section, a wider spread of experimental data is observed in both figures 10(a) and 11(a).

In figures 10(b) and 11(b), the same comparison is shown for the transverse direction—building direction plane. Note that here, the data of the four cross-sections shown in figures 7(b) and 9(b) are combined. Again, the computational microstructure generated by the 3D anisotropic Voronoi algorithm adequately represents the experimental EBSD data. Some minor deviations can be observed in figure 10(b) in the representation of small grains. This can be attributed to the initial mesh size. Some grain seed points with a very small volume have not been assigned to an element in the initial mesh. Upon refinement, more small grains can be taken into account. Such refinement would also improve the resulting grain aspect ratios. The insertion of a total of four pillow elements over the minor axis of an elongated grain has a much larger effect on the minor axis length than the effect of insertion of the same four pillow elements over the major axis of the same grain on the major axis length. Therefore, due to the pillow element insertion step of the algorithm, the aspect ratio of a largely elongated grain decreases slightly. A finer initial discretization reduces this effect and improves the representativeness of the computational microstructure even further.

Figure 10. Comparison of the grain size distribution between the experimental EBSD data and the resulting computational microstructure for the WAAM example in two cross-sectional planes: (a) deposition direction—building direction and (b) transverse direction—building direction. The gray shaded area illustrates the range of experimental grain size distributions that can be obtained for a grain misorientation threshold taken between 0.5°–5° as indicated.

Figure 11. Comparison between the experimental EBSD data for two grain misorientation threshold definitions (as indicated) and resulting computational geometry for the WAAM microstructure in two cross-sectional planes: (a) deposition direction—building direction and (b) transverse direction—building direction. Each dot represents a single grain. The angular coordinate corresponds to its columnar direction and the radial coordinate corresponds to its aspect ratio.

4.2. Magnesium alloy

To show the wide range of applicability of the algorithm, the second application case consists of a tension loaded magnesium AZ31B alloy with a twinned microstructure. Although twinning is not the mechanism that is mimicked in the 3D anisotropic Voronoi algorithm, the resulting

Figure 12. (a) Experimental 3D EBSD data of the magnesium alloy [28] and four stages of the 3D anisotropic Voronoi algorithm during the computational microstructure generation: (b) input and domain of interest, (c) initial mesh and assignment of elements to grains, (d) final geometry and (e) final hexahedral mesh. The top and bottom rows show the front and back of the domain, respectively.

experimental microstructure consists of elongated grains with columnar directions that vary spatially, which can still be represented by the algorithm. 3D EBSD data of a magnesium alloy microstructure is obtained from [28] and a subsection of $18.4 \times 20.8 \times 20 \mu\text{m}$ is shown in figure 12(a).

The input of the 3D anisotropic Voronoi algorithm is obtained by fitting ellipsoids to grains by again equalizing their second moments. In this case, some grains have been fitted by multiple ellipsoids through first splitting each grain into various bulky parts. However, note that the same orientation is assigned to each grain part, so that it becomes a single grain in the final geometry. Each of these regions is represented by a single ellipsoid. Its center is subsequently used as seed point coordinate and the ellipsoid shape represents the corresponding velocity growth field. The resulting input of the algorithm is illustrated in figure 12(b).

A voxel mesh of $16 \times 18 \times 17 = 4896$ hexahedral elements is adopted as initial mesh, which is depicted in figure 12(c), together with the algorithm's assignment of elements to grains. Periodic boundary conditions are not applied, since the purpose of this application case is to represent a specific part of the magnesium alloy microstructure. The final computational geometry and corresponding hexahedral mesh are shown in figures 12(d) and (e), respectively. Upon comparison of the experimental 3D EBSD data (figure 12(a)) with the final computational grain geometry (figure 12(d)), a convincing resemblance is observed. The final hexahedral mesh is fully grain conforming and consists of 27 964 elements and 31 626 nodes. Note that using this discretization (94 878 degrees of freedom) in microstructural simulations is much more computationally efficient compared to using the experimental 3D EBSD data as direct voxel mesh (381 123 degrees of freedom). On top of that, using the experimental data directly, results in the presence of non-grain-conforming boundaries, possibly introducing spurious stress concentrations at these staggered interfaces [21], which are not present in the final mesh of the 3D anisotropic Voronoi algorithm.

Figure 13. Comparison of the (a) grain size distribution and (b) aspect ratio (maximum over minimum axis length) distribution between the input and resulting computational geometry generated by the 3D anisotropic Voronoi algorithm.

Figure 14. Equal area pole figures in which each dot represents a single normalized principal axis direction vector of (a) the input velocity field and (b) the grain fitted ellipsoids of the computational geometry. The color of each dot resembles the ratio of the corresponding principal axis length to the minimum principal axis length for each ellipsoid. This ratio trivially equals 1 for the principal axis vector with minimum length, which is not relevant. These vectors are therefore depicted by an open circle.

In order to analyze the correspondence in more detail, a comparison is made between the grain seed point input data of the algorithm and the characteristics of the output grain morphology in terms of grain size, aspect ratio and orientation. In figure 13(a), the input and resulting grain size distribution are compared. A convincing correspondence is observed. In figure 13(b), the aspect ratio (maximum over minimum principal axis length) of the grains in the input and result of the algorithm are compared. Also here, as satisfactory similarity is established. Similar to the WAAM computational microstructure, the anisotropic 3D Voronoi algorithm makes largely elongated grains slightly more spherical, due to the pillow element insertion step.

In the pole figures presented in figures 14(a) and (b), each dot represents a single normalized principal axis direction vector of the input velocity field and the resulting grain fitted ellipsoid of the computational microstructure, respectively. The color of each dot resembles the ratio of the corresponding principal axis length to the minimum principal axis length for each ellipsoid. Focusing on the positions of the dots, it is obvious that the general characteristics of the principal axis directions are reproduced properly by the 3D anisotropic Voronoi algorithm. The colors of the dots also reveal that the procedure slightly decreases the aspect ratio of largely elongated grains. If needed, this effect can be reduced by employing a finer initial mesh to improve the representativeness of the computational microstructure even further.

5. Conclusions

In this work, a previously developed two-dimensional anisotropic Voronoi algorithm is extended to 3D, along with the generation of a hexahedral mesh and mesh optimization. The algorithm takes into account the preferred grain growth directions, aspect ratios and sizes in the definition of an ellipsoidal growth velocity field, which is used to assign elements from a simple initial hexahedral mesh with corresponding boundary definitions to specific grain seed points. For specific applications in which a predetermined mesh is used, e.g. using an FFT solver, the assignment of elements to grain seed points can be considered the output of the algorithm. For many other applications, e.g. finite element crystal plasticity simulations, element pillowing, smoothing and mesh optimization operations are applied afterwards to arrive at a fully grain conforming and hexahedral mesh, which can be directly used in microstructural modeling simulations. The required input for the algorithm can be obtained from e.g. thermal (history) simulations, microstructural statistics or (3D) experimental observations.

The algorithm is useful in a wide variety of applications, and is specifically interesting for the generation of computational microstructures that include grains with a spatially varying elongated geometry. Two application cases have been discussed, a wire + arc additively manufactured microstructure and a magnesium alloy with a twinned microstructure. It is demonstrated that the algorithm can be used in curved, non-convex domains and that periodicity of the grains and the mesh can be enforced. The computational microstructure is compared to experimental data in terms of grain size, grain aspect ratio and grain columnar direction distribution. In both cases, the algorithm adequately reproduces an RVE with a convincing resemblance to the experimental data.

Acknowledgments

This research was carried out under project number P16-46/S17024e, which is part of the AiM2XL program, in the framework of the Partnership Program of the Materials innovation institute M2i (www.m2i.nl) and the Netherlands Organization for Scientific Research (www.nwo.nl). The research was conducted in collaboration with industrial partners and supported by the Rotterdam Fieldlab Additive Manufacturing BV (RAMLAB), www.ramlab.com. Furthermore, the authors would like to thank Gregory Rohrer for sharing the experimental 3D EBSD data of the magnesium alloy and Mathieu Oude Vrielink for the inspiration that led to this work.

Data availability

The data that support the findings of this study are available upon reasonable request from the authors.

Appendix A. Ellipsoidal velocity field

Given a local Cartesian coordinate system $\{\vec{e}_x, \vec{e}_y, \vec{e}_z\}$, with coordinates x, y, z , a 3D ellipsoid is defined by

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1, \quad (25)$$

with a, b, c the major axis lengths of the ellipsoid in the direction of $\vec{e}_x, \vec{e}_y, \vec{e}_z$, respectively. The volume V of this ellipsoid is

$$V = \frac{4}{3}\pi abc. \quad (26)$$

The orientation of this local Cartesian coordinate system (and consequently the ellipsoid) is defined by two preferred growth vectors $\vec{p}$ and $\vec{q}$, which should be perpendicular to each other (i.e. $\vec{p} \cdot \vec{q} = 0$). Here, $\vec{p}$ is defined to be aligned with the local x -axis (i.e. $\vec{e}_p = \vec{e}_x$) and $\vec{q}$ is aligned with the local z -axis (i.e. $\vec{e}_q = \vec{e}_z$). Furthermore, the lengths of these vectors $\|\vec{p}\|, \|\vec{q}\|$ define the aspect ratios of the ellipsoid, i.e.

$$\frac{a}{b} = \|\vec{p}\|, \quad \frac{c}{b} = \|\vec{q}\|. \quad (27)$$

Based on these equations, a, b and c are defined as

$$a = \|\vec{p}\|s, \quad b = s, \quad c = \|\vec{q}\|s, \quad (28)$$

with s a scaling factor that can be found by substitution of relations (28) in equation (26) as

$$s = \sqrt[3]{\frac{3}{4} \frac{V}{\pi \|\vec{p}\| \|\vec{q}\|}}. \quad (29)$$

The final goal is to derive the radius of the ellipsoid r in any given direction $\vec{e}_r$, representing the directional growth velocity $v(\vec{e}_r)$. First, a spherical coordinate system (radial distance r , azimuthal angle θ and polar angle φ) is introduced as

$$x = r \sin(\varphi) \cos(\theta), \quad y = r \sin(\varphi) \sin(\theta), \quad z = r \cos(\varphi). \quad (30)$$

Substitution of these relations in the definition of the ellipsoid (25) results in

$$r^2 \left[\frac{\sin^2(\varphi) \cos^2(\theta)}{a^2} + \frac{\sin^2(\varphi) \sin^2(\theta)}{b^2} + \frac{\cos^2(\varphi)}{c^2} \right] = 1. \quad (31)$$

Now, substituting relations (28) yields

$$\frac{1}{s^2} \left[\left(\frac{1}{\|\vec{p}\|^2} \cos^2(\theta) + \sin^2(\theta) \right) \sin^2(\varphi) + \frac{1}{\|\vec{q}\|^2} \cos^2(\varphi) \right] = \frac{1}{r^2}. \quad (32)$$

Let $\hat{r}_2(\theta)$ define the dimensionless radius of the 2D elliptical cross-section of the 3D ellipsoid in the plane through the center and orthogonal to $\vec{q}$, which has been highlighted in figure 1. In a similar manner as shown for the 3D ellipsoid, it can be shown that

$$\frac{1}{\hat{r}_2^2} = \frac{1}{\|\vec{p}\|^2} \cos^2(\theta) + \sin^2(\theta), \quad (33)$$

which can be substituted in equation (32):

$$\frac{1}{s^2} \left[\frac{1}{[\hat{r}_2(\theta)]^2} \sin^2(\varphi) + \frac{1}{\|\vec{q}\|^2} \cos^2(\varphi) \right] = \frac{1}{r^2}. \quad (34)$$

Then, substitution of $\sin^2(\varphi) = 1 - \cos^2(\varphi)$ and rewriting leads to

$$\frac{1}{s^2} \left[\frac{1}{[\hat{r}_2(\theta)]^2} + \left(\frac{1}{\|\vec{q}\|^2} - \frac{1}{[\hat{r}_2(\theta)]^2} \right) \cos^2(\varphi) \right] = \frac{1}{r^2}. \quad (35)$$

Now, using $\cos(\varphi) = \vec{e}_q \cdot \vec{e}_r$ gives

$$\frac{1}{s^2} \left[\frac{1}{[\hat{r}_2(\theta)]^2} + \left(\frac{1}{\|\vec{q}\|^2} - \frac{1}{[\hat{r}_2(\theta)]^2} \right) (\vec{e}_q \cdot \vec{e}_r)^2 \right] = \frac{1}{r^2}. \quad (36)$$

From this equation, the directional growth velocity $v(\vec{e}_r)$ can be found as

$$v(\vec{e}_r) = r = s\hat{r}_3, \quad (37)$$

with normalized radius $\hat{r}_3$ defined as

$$\hat{r}_3 = \left[\frac{1}{[\hat{r}_2(\vec{e}_r^*)]^2} + \left(\frac{1}{\|\vec{q}\|^2} - \frac{1}{[\hat{r}_2(\vec{e}_r^*)]^2} \right) (\vec{e}_q \cdot \vec{e}_r)^2 \right]^{-\frac{1}{2}}. \quad (38)$$

In a similar manner, it can be shown that $\hat{r}_2$ can be written as

$$\hat{r}_2(\vec{e}_r^*) = \left[1 + \left(\frac{1}{\|\vec{p}\|^2} - 1 \right) (\vec{e}_p \cdot \vec{e}_r^*)^2 \right]^{-\frac{1}{2}}, \quad (39)$$

which is more useful than equation (33) in the numerical implementation. In this equation,

$$\vec{e}_r^* = \frac{\vec{e}_r - \vec{e}_r \cdot \vec{e}_q \vec{e}_q}{\|\vec{e}_r - \vec{e}_r \cdot \vec{e}_q \vec{e}_q\|} \quad (40)$$

is the normalized projection of $\vec{e}_r$ onto the 2D elliptical cross-section of the 3D ellipsoid in the plane through the center and orthogonal to $\vec{q}$ (see figure 1).

ORCID iDs

T F W van Nuland <https://orcid.org/0000-0003-2734-8691>

J A W van Dommelen <https://orcid.org/0000-0003-4029-0308>

References

- [1] Falco S, Jiang J, De Cola F and Petrinic N 2017 Generation of 3D polycrystalline microstructures with a conditioned Laguerre–Voronoi tessellation technique *Comput. Mater. Sci.* **136** 20–8
- [2] Ghazvinian E, Diederichs M S and Quey R 2014 3D random Voronoi grain-based models for simulation of brittle rock damage and fabric-guided micro-fracturing *J. Rock Mech. Geotechnical Eng.* **6** 506–21
- [3] Benmessaoud F, Velay V, Cheikh M, Vidal V, Boher C and Rézaï-Aria F 2019 Multi-scale numerical analysis of the effect of microstructural features on the mechanical behavior of polycrystalline Ti-6Al-4V alloy *Eng. Trans.* **67** 227–42
- [4] Boots B N 1982 The arrangement of cells in random networks *Metallography* **15** 53–62
- [5] Gervois A, Troadec J P and Lemaitre J 1992 Universal properties of Voronoi tessellations of hard discs *J. Phys. A: Math. Gen.* **25** 6169–77
- [6] Morley D O and Wilson M 2020 Voronoi diagrams in quasi-2D hard sphere systems *J. Stat. Mech.* **093201**
- [7] Aurenhammer F 1987 Power diagrams: properties, algorithms and applications *SIAM J. Comput.* **16** 78–96
- [8] Oude Vrielink M A, van Dommelen J A W and Geers M G D 2020 Computational analysis of the evolution of the brittle-to-ductile transition of tungsten under fusion conditions *Modelling Simul. Mater. Sci. Eng.* **29** 015005
- [9] Möller J 1992 Random Johnson-Mehl tessellations *Adv. Appl. Probab.* **24** 814–44
- [10] Coster M, Arnould X, Chermant J-L, El Moataz A and Chartier T 2005 A microstructural model by space tessellation for a sintered ceramic: cerine *Image Anal. Stereology* **24** 105–16
- [11] van Nuland T F W, van Dommelen J A W and Geers M G D 2021 Microstructural modeling of anisotropic plasticity in large scale additively manufactured 316L stainless steel *Mech. Mater.* **153** 103664
- [12] Barriobero-Vila P, Gussone J, Stark A, Schell N, Haubrich J and Requena G 2018 Peritectic titanium alloys for 3D printing *Nat. Commun.* **9** 3426
- [13] Chen Q, Li H and Shen H 2019 Transient modeling of grain structure and macrosegregation during direct chill casting of Al–Cu alloy *Processes* **7** 333
- [14] Cho J 2015 Voronoi diagram and microstructure of weldment *J. Mech. Sci. Technol.* **29** 371–4
- [15] van Nuland T F W, van Dommelen J A W and Geers M G D 2021 An anisotropic Voronoi algorithm for generating polycrystalline microstructures with preferred growth directions *Comput. Mater. Sci.* **186** 109947
- [16] Okabe A, Boots B, Sugihara K and Chiu S N 2009 *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams (Wiley Series in Probability and Statistics)* (New York: Wiley)
- [17] Shepherd J F and Johnson C R 2008 Hexahedral mesh generation constraints *Eng. Comput.* **24** 195–213
- [18] Cheng J, Shahba A and Ghosh S 2016 Stabilized tetrahedral elements for crystal plasticity finite element analysis overcoming volumetric locking *Comput. Mech.* **57** 733–53
- [19] Owen S J and Shelton T R 2015 Evaluation of grid-based hex meshes for solid mechanics *Eng. Comput.* **31** 529–43
- [20] Fabritius B and Tabor G 2016 Improving the quality of finite volume meshes through genetic optimisation *Eng. Comput.* **32** 425–40
- [21] Lim H, Abdeljawad F, Owen S J, Hanks B W, Foulk J W and Battaile C C 2016 Incorporating physically-based microstructures in materials modeling: bridging phase field and crystal plasticity frameworks *Modelling Simul. Mater. Sci. Eng.* **24** 045016
- [22] Owen S J, Brown J A, Ernst C D, Lim H and Long K N 2017 Hexahedral mesh generation for computational materials modeling *Proc. Eng.* **203** 167–79
- [23] Francois M M *et al* 2017 Modeling of additive manufacturing processes for metals: challenges and opportunities *Curr. Opin. Solid State Mater. Sci.* **21** 198–206
- [24] Field D A 1988 Laplacian smoothing and Delaunay triangulations *Commun. Appl. Numer. Methods* **4** 709–12
- [25] Calvo N A and Idelsohn S R 2001 All-hexahedral mesh smoothing with a node-based measure of quality *Int. J. Numer. Methods Eng.* **50** 1957–67
- [26] Rodgers T M, Bishop J E and Madison J D 2018 Direct numerical simulation of mechanical response in synthetic additively manufactured microstructures *Modelling Simul. Mater. Sci. Eng.* **26** 055010

- [27] Herriott C, Li X, Kouraytem N, Tari V, Tan W, Anglin B, Rollett A D and Spear A D 2019 A multi-scale, multi-physics modeling framework to predict spatial variation of properties in additive-manufactured metals *Modelling Simul. Mater. Sci. Eng.* **27** 025009
- [28] Ram F, Lloyd J T and Rohrer G S 2020 Habit planes of twins in a deformed Mg alloy determined from three-dimensional microstructure analysis *Mater. Charact.* **159** 110014
- [29] Arul Kumar M, Capolungo L, McCabe R J and Tomé C N 2019 Characterizing the role of adjoining twins at grain boundaries in hexagonal close packed materials *Sci. Rep.* **9** 3846
- [30] Lhuillier P E, Ferre A, Schumm A and Shahjahan S 2018 Grain-scale modeling of anisotropic weld and finite element simulation of ultrasonic propagation: toward a full predictions of echoes and structural noise in weld inspection *12th European Conf. Non-destructive Testing (ECNDT 2018)*
- [31] Wei H L, Elmer J W and DebRoy T 2017 Three-dimensional modeling of grain structure evolution during welding of an aluminum alloy *Acta Mater.* **126** 413–25
- [32] Jia D, Sun W, Xu D and Liu F 2019 Dynamic recrystallization behavior of GH4169G alloy during hot compressive deformation *J. Mater. Sci. Technol.* **35** 1851–9
- [33] Palmeira Belotti L, van Dommelen J A W, Geers M G D, Goulas C, Ya W and Hoefnagels J P M 2022 Microstructural characterisation of thick-walled wire arc additively manufactured stainless steel *J. Mater. Process. Technol.* **299** 117373
- [34] Bachmann F, Hielscher R and Schaeben H 2010 Texture analysis with mtex—free and open source software toolbox *Texture and Anisotropy of Polycrystals III (Solid State Phenomena Vol 160)* (Switzerland: Trans Tech Publications Ltd) pp 63–8
- [35] Palmeira Belotti L, Hoefnagels J P M, Geers M G D and van Dommelen J A W 2021 A method for extracting average microstructure cells from micrographs of additively manufactured parts (in preparation)
- [36] Mulchrone K F and Choudhury K R 2004 Fitting an ellipse to an arbitrary shape: implications for strain analysis *J. Struct. Geol.* **26** 143–53