



**HAL**  
open science

# A simple and fast algorithm for computing discrete Voronoi, Johnson-Mehl or Laguerre diagrams of points

Hervé Moulinec

► **To cite this version:**

Hervé Moulinec. A simple and fast algorithm for computing discrete Voronoi, Johnson-Mehl or Laguerre diagrams of points. 2022. hal-03522759v1

**HAL Id: hal-03522759**

**<https://hal.science/hal-03522759v1>**

Preprint submitted on 13 Jan 2022 (v1), last revised 31 May 2022 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A simple and fast algorithm for computing discrete Voronoi, Johnson-Mehl or Laguerre diagrams of points

H. Moulinec<sup>a</sup>

<sup>a</sup> Aix-Marseille Univ, CNRS, Centrale Marseille, LMA, 4 Impasse Nikola Tesla, CS 40006, 13453 Marseille Cedex 13, France

---

## Abstract

This article presents an algorithm to compute *digital images* of Voronoi, Johnson-Mehl or Laguerre diagrams of a set of punctual sites, in a domain of a Euclidean space of any dimension. The principle of the algorithm is, in a first step, to investigate the voxels in balls centred around the sites, and, in a second step, to process the voxels remaining outside the balls. The optimal choice of ball radii can be determined analytically or numerically, which allows a performance of the algorithm in  $O(N_v \ln N_s)$ , where  $N_v$  is the total number of voxels of the domain and  $N_s$  the number of sites of the tessellation. Periodic and non-periodic boundary conditions are considered.

A major advantage of the algorithm is its simplicity which makes it very easy to implement.

This makes the algorithm suitable for creating high resolution images of microstructures containing a large number of cells, in particular when calculations using FFT-based homogenisation methods are then to be applied to the simulated materials.

*Keywords:* Voronoi diagram, Johnson-Mehl tessellation, Laguerre tessellation, micromechanics

---

## 1. Introduction

### 1.1. Generation of microstructures in computational homogenisation

Over the last three decades, considerable progress has been made in what might be called “computational homogenisation”, as the part of computational mechanics dealing with homogenisation or micromechanics.

Until the early 1990s, the available hardware and software computer resources hardly allowed numerical simulations on realistic microstructures (whether real or simulated) of heterogeneous materials whose geometries are often very complex. Therefore, most geometries of the microstructures studied were still very simplified. To overcome these limitations, the efforts of numerical engineers in the 1990s and 2000s focused on improving existing methods such as finite elements, for example with the development of multilevel finite element method (Feyel (2003)), or developing new, more efficient or more adapted methods, notably the FFT-based homogenisation method introduced by Moulinec and Suquet (1994). On the other hand, the progress of computer hardware, in particular the development of new parallel computers and the software tools (languages and application programming interfaces) that accompanied them, allowed numerical engineers to improve their codes, often at the cost of adapting numerical methods to this new paradigm (for example with domain decomposition methods (Tallec et al. (1991), Farhat and Roux (1991), Gosselet and Rey (2006))).

In parallel with research on calculation methods, studies have been carried out on the microstructures themselves. The question of the representativeness of the elementary volume studied was raised with perhaps greater acuity than before (Gusev (1997), Kanit et al. (2003)). More recently, since the 2010s, the issue of the generation of artificial microstructures has received more particular attention. One of the questions is how to create an artificial microstructure that “mimics” a real microstructure, (Fritzen et al. (2009), Quey et al. (2011), Quey and Renversade (2018), van Nuland et al. (2021)). Another important issue is the efficiency of the algorithms generating the microstructures, whose performance can differ greatly from one method to another. The reader is referred to the review article by Bargmann et al. (2018) which covers the topic of microstructure generation.

### 1.2. Context of the study

The present paper focuses on the generation of Voronoi, Laguerre and Johnson-Mehl diagrams - also called Voronoi, Johnson-Mehl and Laguerre tessellations - which are widely used for the simulation of microstructures of polycrystalline materials. More precisely, the objective being to calculate the mechanical behaviour of materials using FFT-based homogenisation methods (Moulinec and Suquet (1994), Moulinec and Suquet (1998)), the motivation of this study is to generate *digital images* of microstructures, i.e. discretisations of microstructures according to a regular grid, whose elementary data are generally called pixels in the two-dimensional case and voxels in the three-dimensional case. In the rest of this article, the term voxel is used for any dimension (2, 3 or higher).

Instead of the term “digital image”, which is widely used in the image processing community (Pratt (2007)), some authors prefer to speak of “raster images” as opposed to “vector images” or “vector graphics” for which the image is described by geometric shapes. The object of this study is thus referred to in the literature as “raster Voronoi tessellations” (Lee et al. (2011)) or “raster-based method” (Li et al. (1999)), but more often as “discrete Voronoi diagrams” (Velić et al. (2009)).

A special attention is given, in the present paper, to the case of large images with a high number of grains. Indeed, it is not uncommon nowadays to apply FFT-based homogenisation methods on microstructures discretised with hundreds of millions of voxels (Müller et al. (2015), Boittin et al. (2017), Chen et al. (2019), Vincent et al. (2020), Marano et al. (2021) ) and, in the case of polycrystalline materials, composed of hundreds of thousands of grains (Wojtacki et al. (2020)). The generation of such microstructures can lead to non negligible computation times. For example, to generate an image of  $1000 \times 1000 \times 1000$  voxels of a Voronoi tessellation containing 1 000 000 grains, the general code Neper of polycrystal generation (Quey (2021a)) requires about 11 300 seconds on a 8-core processor Xeon Silver 4208 (2.10 GHz).

Moreover, since FFT-based methods make the implicit assumption that the fields involved are periodic, we are interested, although not exclusively, in the case where the tessellations exhibit periodic conditions at the boundaries of the spatial domain studied.

An additional constraint imposed by micromechanical studies is that the “sites” or “seeds” defining the tessellation are not necessarily placed on the exact positions of voxels of the image to be created. Indeed, the grid of the image must be able to be chosen independently of the layout of the sites, for example in the case where several realisations of the same tessellation are desired with images of different spatial resolutions.

### 1.3. Algorithms for creating tessellations

Simply put, a Voronoi tessellation of a given set of discrete points in a Euclidean space, hereafter referred to as sites, is a tiling of the space into regions, each of which gathers all points in the space closer to a given site than to any other. Laguerre and Johnson-Mehl tessellations can be considered as variants of Voronoi tessellations, where the function evaluating the proximity of one point to another is different from the Euclidean distance. More precise definitions are given in section 2.

Since their introduction (Voronoi (1908)), Voronoi tessellations gave rise to an abundant literature. Many authors have been interested in the mathematical properties of these constructions (see Stoyan et al. (1995), Aurenhammer (1991) or Okabe et al. (2000) for Voronoi tessellations, Møller (1992) for those of Johnson-Mehl and Aurenhammer (1991), Okabe et al. (2000) or Lautensack and Zuyev (2008) for Laguerre tessellations) and, in particular, in how to describe and calculate the cells of the tiling of the space that constitutes them. Numerous algorithms have been proposed, among which two families can be distinguished schematically.

1. A first category of algorithms, mainly from the robotics and computer vision communities, are interested in the calculation of the Euclidean Distance Transform (EDT), which consists in evaluating for each voxel of an image its distance to a given region of interest of the image (see among others Saito and Toriwake (1994), Breu et al. (1995), Hirata (1996)). When these regions of interest are composed of isolated voxels, this problem reduces to that of creating an image of a Voronoi tessellation. See Fabbri et al. (2008) for a review and comparison of existing algorithms. Several of the proposed algorithms achieve the optimal performance in  $O(N_v)$ , where  $N_v$  is the total number of voxels in the image.
2. Another family of methods, mainly from the computational geometry community, determines the geometric parameters of the cells (for example, the vertices, edges, facets of the polygons or polyhedra which constitute the cells of the Voronoi tessellations in the two-dimensional or three-dimensional cases, respectively). An

extensive review of the existing algorithms for Voronoi diagrams, and a comparison of their efficiency can be found in Okabe et al. (2000). It appears that some algorithms attain the optimal time complexity  $O(N_s)$ , where  $N_s$  is the number of sites, but that many methods behave in  $O(N_s \ln N_s)$ , which is the “worst-case optimal time complexity”, or even worse.

However, these algorithms have a number of drawbacks for the problem addressed in this paper. Although the approximations on the calculation of the distance in early EDT algorithms, imposed largely by the technical limitations of the time, have been overcome today with algorithms that achieve what is called “Exact EDT”, all of the proposed algorithms, to the best of the author’s knowledge, place the sites of the Voronoi cells in the exact location of a voxel, which is inadequate for the present study.

For the case of interest of this study, i.e. the creation of images of tessellations, the algorithms of the second type must be completed by a step where it must be determined to which cell of the tessellation each voxels of the image belongs. So to speak, after having described the geometry of the cells of the tessellation, the algorithm must “fill” the cells with the voxels they contain. This option is rarely implemented in the existing codes but is proposed by the Neper code (Quey et al. (2011), Quey (2021a)), where it is called “raster tessellation meshing”. As the whole numerical methods in question are rather delicate to implement, materials science researchers often opt either to use Neper or to implement a simple brute force algorithm whose low performance in  $O(N_v N_s)$  makes it unsuitable for large simulations.

The aim of this article is to propose a simple algorithm, just a little more difficult to implement than the brute force algorithm, but with much better performance in  $O(N_v \ln N_s)$ .

First, in section 2 of the article, the tessellations of Voronoi, Johnson-Mehl and Laguerre are briefly introduced. In section 3, an accelerated algorithm is presented for the different types of tessellations and its performance is compared to that of Neper code, in section 4.

## 2. Voronoi, Johnson-Mehl and Laguerre tessellations

### 2.1. Voronoi tessellations

A Voronoi tessellation, also called Voronoi diagram, associated with a given set  $\mathcal{S} = \{\mathbf{x}_s\}_{s=1,\dots,N_s}$  of  $N_s$  points, called “sites”, “seeds” or “generators”, of a given domain  $\Omega$  of a Euclidean space, is a division of the domain into  $N_s$  zones, called “cells”, each cell  $s$  including all the points of the domain that are closer to the site  $s$  of position  $\mathbf{x}_s$  than to any other site in the set. Thus, the cell  $C_s$  associated to the site  $s$  of the set  $\mathcal{S}$  is a subdivision of domain  $\Omega$  defined by

$$C_s = \{\mathbf{x} \in \Omega \mid d(\mathbf{x}, \mathbf{x}_s) \leq d(\mathbf{x}, \mathbf{x}_{s'}), \forall s' \in \mathcal{S}\}, \quad (1)$$

where  $d(\mathbf{x}, \mathbf{y})$  denotes the distance between the two points  $\mathbf{x}$  and  $\mathbf{y}$ . Throughout the article, multidimensional quantities, such as vectors  $\mathbf{x}$  and  $\mathbf{x}_s$  in relation 1, are noted in bold type while scalar data are in normal type.

The distance  $d(\cdot)$  used above can be the usual Euclidean distance, but in the case of periodic conditions on the boundaries of domain  $\Omega$ , the Euclidean distance can advantageously be replaced by what is called hereafter the L-periodic distance. The reader is referred to Appendix A for the definition (A.6) and calculation (A.7) of the L-periodic distance. In the rest of the article, the distance  $d(\cdot)$  denotes either the Euclidean distance or the L-periodic distance, depending on the boundary conditions considered.

In practical applications, the Euclidean space considered is most often two- or three-dimensional, but the definition of the Voronoi tessellation given above is valid for any finite dimension.

An example of a two-dimensional Voronoi tessellation, generated from 50 sites randomly arranged in the area, is shown in Figure 1. Periodic conditions on the borders of the domain have been assumed.

### 2.2. Johnson-Mehl tessellations

The model of the Johnson-Mehl tessellation, also called Johnson-Mehl-Avrami-Kolmogorov tessellation, has been developed to account for crystallisation in a material (Johnson and Mehl (1939)). To put it simply, a crystal grows from a site that appears in the volume at a given location and at a given time, until its boundaries reach those of other crystals. The tessellation is the resulting division of the space, when all the crystals have finished growing and have therefore completely covered the domain of space  $\Omega$  considered.

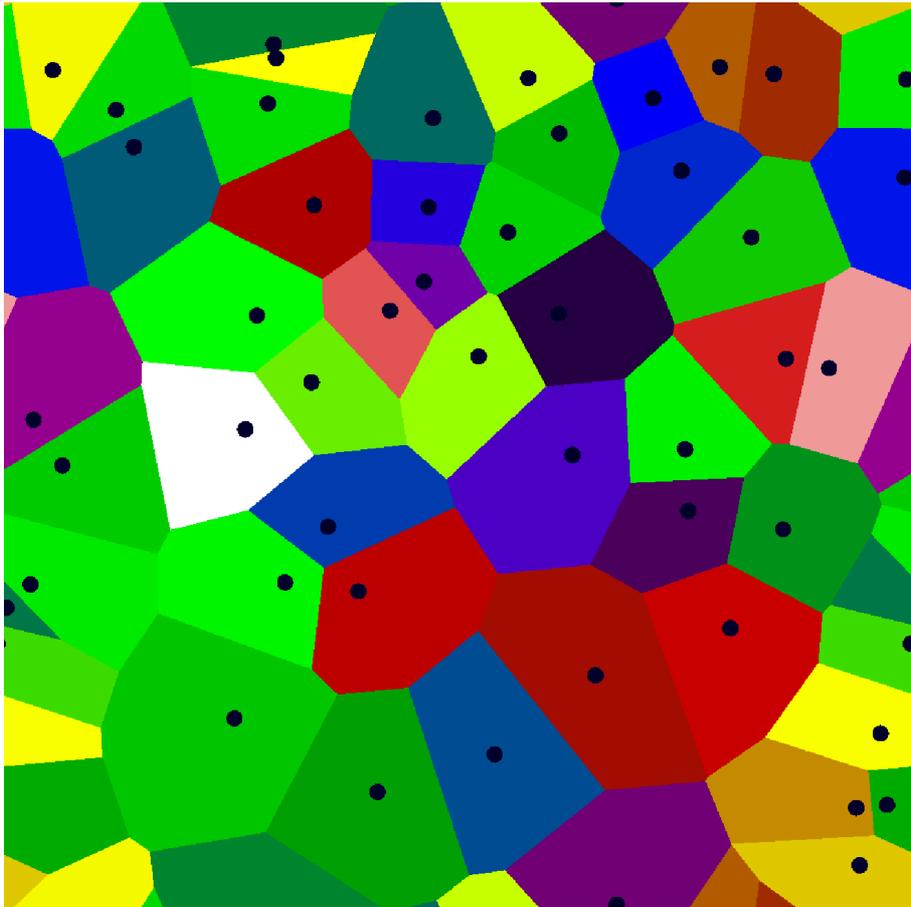


Figure 1: Voronoi tessellation of 50 punctual sites, with periodic conditions. The cells are plotted in arbitrary colours. The sites are marked by small black circles.

Thus, considering the set of sites given by

$$\mathcal{S} = \{(\mathbf{x}_s, t_s)\}_{s=1,2,\dots,N_s}, \quad (2)$$

where  $\mathbf{x}_s$  is the position of site  $s$  and  $t_s$  is its birth time (i.e. the time at which the site appears in  $\Omega$  and starts to grow), a given point  $\mathbf{x}$  of the space would be reached by the growth of the crystal generated by site  $s$  - in the absence of interactions with any other site - at time  $t$  which is

$$t = d(\mathbf{x}, \mathbf{x}_s)/G + t_s, \quad (3)$$

where  $G$  is the radial growth of the crystals, supposed to be constant and to be identical for all the crystals.

Thus, the cell associated with a given site  $s$  is defined by

$$C_s = \{\mathbf{x} \in \Omega \mid d(\mathbf{x}, \mathbf{x}_s)/G + t_s \leq d(\mathbf{x}, \mathbf{x}_{s'})/G + t_{s'}, \forall s' \in \mathcal{S}\} \quad (4)$$

or, equivalently,

$$C_s = \{\mathbf{x} \in \Omega \mid d(\mathbf{x}, \mathbf{x}_s) + Gt_s \leq d(\mathbf{x}, \mathbf{x}_{s'}) + Gt_{s'}, \forall s' \in \mathcal{S}\}, \quad (5)$$

$G$  being supposed to be positive.

### 2.3. Laguerre tessellations

A Laguerre tessellation, also called power diagram, is generally presented in the literature as a partition of the space in cells defined by a set of spheres  $s$  of centres  $\mathbf{x}_s$  and radii  $r_s$ . A given point  $\mathbf{x}$  belongs to the cell  $C_s$  associated with the site - or sphere -  $s$  which minimises the so-called ‘‘power distance’’  $p_d(\mathbf{x}, s)$  defined by

$$p_d(\mathbf{x}, s) = d^2(\mathbf{x}, \mathbf{x}_s) - r_s^2. \quad (6)$$

Hence, the cell associated to the site  $s$  is defined by

$$C_s = \{\mathbf{x} \in \Omega \mid p_d(\mathbf{x}, s) \leq p_d(\mathbf{x}, s'), \forall s' \in \mathcal{S}\} \quad (7)$$

Alternatively, from a physical point of view and as detailed in Appendix B, a Laguerre tessellation can be considered as the resulting partition of a crystallisation process starting from a set of sites  $\mathcal{S} = \{(\mathbf{x}_s, t_s)\}_{s=1,2,\dots,N_s}$ , where  $\mathbf{x}_s$  is the location of the site  $s$  and  $t_s$  is its birth time, with a constant growth  $G$  of the square of the radii of the crystals. The principle remains similar to that of Johnson-Mehl tessellation: crystal  $s$  reaches a given point  $\mathbf{x}$  at time  $t$  given by

$$t = d^2(\mathbf{x}, \mathbf{x}_s)/G + t_s, \quad (8)$$

and finally the cell  $C_s$  associated to a given site  $s$  is defined by

$$C_s = \{\mathbf{x} \in \Omega \mid d^2(\mathbf{x}, \mathbf{x}_s) + Gt_s \leq d^2(\mathbf{x}, \mathbf{x}_{s'}) + Gt_{s'}, \forall s' \in \mathcal{S}\}. \quad (9)$$

This notation has the advantage of making the definitions of Johnson-Mehl and Laguerre tessellations similar.

*Remark.* In the particular case where all sites appear at the same time, or in the case  $G$  is zero, when using definitions 5 and 9, Johnson-Mehl and Laguerre tessellations reduce to a Voronoi tessellation.

## 3. Tessellation algorithms

In this section, algorithms are presented that solve the problem studied, the terms of which are briefly recalled here.

- The algorithms must create a *digital image* of a Voronoi, Johnson-Mehl or Laguerre tessellation of a set of punctual sites.
- The positions of the sites are arbitrary and do not necessarily coincide with voxel centres.

- The distance used to estimate the proximity of a voxel to a site must be *exact*.
- The boundary conditions of the image created can be periodic or non-periodic.
- Although most applications of Voronoi tessellation relate to two- or three-dimensional spaces, the algorithms should apply to spaces of any dimension.
- The domain  $\Omega$ , in which the tessellation is created, is a  $d$ -dimensional rectangular parallelepiped, i.e. a volume of the Euclidean space of dimension  $d$ , defined by  $\Omega = [0, L_1) \times \dots \times [0, L_d)$  in a given orthonormal basis.

In addition, the arrangement of the sites in the volume are considered purely random, following a uniform probability distribution. In the case of the Voronoi diagram, this type of tessellations is sometimes called "Poisson-Voronoi tessellations".

The computational cost of the algorithms are estimated by counting the total number of calculations of distance between points performed during the execution of the algorithms (and in some cases, the number of calculations of the square of the distance). The effect of memory access and compilation optimisation are not considered, as it largely depends on the system on which the algorithms are implemented.

### 3.1. A brute force algorithm

The most simple algorithm to build an image of a Voronoi tessellation consists of calculating for each voxel its distance to all the sites and thus determining the closest site. Diagram 1 schematises this algorithm. Similar algorithms can easily be devised to Johnson-Mehl or Laguerre tessellations.

---

**Algorithm 1:** Brute force algorithm to create an image of a Voronoi tessellation from a set of positions of sites in a Euclidean space of dimension  $d$ .  $\mathbf{k}$  gathers the  $d$  indices of the voxels in the image.  $\mathbf{x}(\mathbf{k})$  is the location of voxel  $\mathbf{k}$ .

---

**input** : a set of  $N_s$  sites of positions  $\{\mathbf{x}_s\}$  in the unit cell  $\Omega = [0, L_1) \times [0, L_2) \times \dots \times [0, L_d)$

**output:**

- an image  $\mathcal{I}$  of  $\mathbf{n} = (n_1, n_2, \dots, n_d)$  voxels describing the tessellation in space domain  $\Omega$
- an image  $\mathcal{D}$  of  $\mathbf{n} = (n_1, n_2, \dots, n_d)$  voxels giving the distance of each point  $\mathbf{x}(\mathbf{k})$  to the closest Voronoi site

**forall**  $\mathbf{k} \in \{1, \dots, n_1\} \times \{1, \dots, n_2\} \times \dots \times \{1, \dots, n_d\}$  **do**

$\mathcal{D}(\mathbf{k}) = \infty$

**forall**  $s = 1$  to  $N_s$  **do**

**if**  $d(\mathbf{x}(\mathbf{k}), \mathbf{x}_s) < \mathcal{D}(\mathbf{k})$  **then**

$\mathcal{I}(\mathbf{k}) = s$

$\mathcal{D}(\mathbf{k}) = d(\mathbf{x}(\mathbf{k}), \mathbf{x}_s)$

**end**

**end**

**end**

---

In the case of Voronoi and Laguerre tessellations (but not of Johnson-Mehl tessellations), a substantial gain in efficiency can be achieved by considering the square of the distance when comparing the distances between a given voxel and all the sites, as this saves the costly calculation of a square root repeated many times. Despite this improvement, the method still performs poorly, the total cost of the algorithm being in  $O(N_v N_s)$ , where  $N_v$  is the total number of voxels in the image and  $N_s$  is the number of sites.

### 3.2. An accelerated Voronoi tessellation algorithm

#### 3.2.1. Principle of the algorithm

Starting from the observation that the cells of a Poisson-Voronoi tessellation are all fairly similar in size, the principle of the algorithm consists first in examining a given neighborhood of all the sites, and determining for each

voxel in the neighborhood whether the site considered is the closest, and then, in a second step, in processing the remaining voxels with the “brute force” algorithm described above.

The algorithm is detailed below.

### 3.2.2. Detail of the algorithm

We start by choosing balls of investigation,  $\mathcal{B}_s$ , centred on each site  $\mathbf{x}_s$  of  $\mathcal{S}$  with a given radius  $r_0$  and a corresponding volume  $v_0$ :

$$\mathcal{B}_s = \{\mathbf{x} \in \Omega \mid d(\mathbf{x}, \mathbf{x}_s) \leq r_0\}. \quad (10)$$

The choice of  $r_0$  and  $v_0$  is discussed in section 3.2.3. The voxels inside each investigation ball are considered and their distance to the associated site is evaluated and saved.

After all the balls have been examined, two cases are possible for a given voxel of domain  $\Omega$ :

- It belongs to one or more balls. In other words, the distance between the voxel and all the sites is smaller than  $r_0$  only for the sites that are the centres of these balls. Thus the voxel belongs to the cell of the nearest of these sites.
- It belongs to no balls.

To complete the tessellation, the distance between each voxel of the second case and all the sites are evaluated and the voxel is assigned to the cell of the closest site.

Diagram 2 summarises the algorithm that performs these tasks.

*Remarks.* Similarly to what has already been mentioned in section 3.1, the algorithm can be accelerated by considering the square of the distance instead of the distance.

A by-product of Algorithm 2 is the image  $\mathcal{D}$  of the EDT (see section 1.3).

### 3.2.3. Optimal choice of the size of the investigation ball

The choice of the radius  $r_0$  of the investigation balls plays a crucial role on the efficiency of Algorithm 2. When  $r_0$  is large, the number of voxels inside the balls and thus the number of calculations of their distance to the centres of the balls are high. Inversely, the number of voxels outside the balls is small and becomes zero when the balls cover the whole domain  $\Omega$ . When  $r_0$  is small, the situation is the opposite: the number of distance calculations in the first step of the algorithm is small and the one of the second step is high. Between these two extremes, there is an optimal choice for the size of the investigation balls.

The number  $n_{step\ 1}$  of distance calculations in the first step of Algorithm 2 is equal to the total number of voxels inside the investigation balls. Neglecting the slight fluctuations of the number of voxels from one ball to another, due to spatial discretisation, one has

$$n_{step\ 1} = N_v \frac{\sum_s v'_s}{V}, \quad (11)$$

where  $V$  is the volume of  $\Omega$  and where  $v'_s$  is the volume of intersection of the ball of investigation  $\mathcal{B}_s$  with the domain of interest, that is

- in the non-periodic case:  $v'_s$  is the volume of the intersection of the ball  $\mathcal{B}_s$  and the domain  $\Omega$ . The balls of investigation crossing the boundaries of  $\Omega$  must be truncated from their part outside  $\Omega$ , resulting in a reduction of the number of voxels to be investigated; in that case, one has  $v'_s < v_0$ . In the case when the ball is entirely inside the domain  $\Omega$ , one has  $v'_s = v_0$ . And in the extreme case when the ball encompasses the whole domain  $\Omega$ , one has  $v'_s = V$ .
- in the periodic case:  $v'_s$  is the volume of the intersection of the ball  $\mathcal{B}_s$  and the domain  $\Omega_s$  which is the translation of parallelepiped  $\Omega$ , centred on  $\mathbf{x}_s$ . As long as the ball is smaller than the sphere inscribed in  $\Omega_s$  (i.e.  $r_0 \leq L_i/2, \forall i = 1, \dots, d$ ), one has  $v'_s = v_0$ . When the ball is bigger than the circumscribed sphere (i.e.  $r_0 \geq \frac{1}{2} \sqrt{\sum_{i=1}^d L_i^2}$ ), one has  $v'_s = V$ . And between these two cases,  $v'_s$  takes intermediate values.

---

**Algorithm 2:** Faster algorithm to create an image of a Voronoi tessellation in a Euclidean space of dimension  $d$

---

**input :**

- a set of  $N_s$  sites of positions  $\{\mathbf{x}_s\}$  in space domain  $\Omega = [0, L_1) \times [0, L_2) \times \dots \times [0, L_d)$

**output:**

- an image  $\mathcal{I}$  of  $\mathbf{n} = (n_1, n_2, \dots, n_d)$  voxels describing the Voronoi tessellation in space domain  $\Omega$
- an image  $\mathcal{D}$  of  $\mathbf{n} = (n_1, n_2, \dots, n_d)$  voxels giving the distance of each point  $\mathbf{x}(\mathbf{k})$  to the closest Voronoi site (i.e. the map of the Euclidean Distance Transform)

Initialisation:

radius  $r_0$  of the balls of investigation is calculated using relation 17

**forall**  $\mathbf{k} \in \{1, \dots, n_1\} \times \{1, \dots, n_2\} \times \dots \times \{1, \dots, n_d\}$  **do**

$\mathcal{D}(\mathbf{k}) = \infty$

**end**

Step 1:

**forall**  $s = 1, 2, \dots, N_s$  **do**

**forall**  $\mathbf{k} \mid \mathbf{x}(\mathbf{k}) \in \mathcal{B}_s(\text{Ball of centre } \mathbf{x}_s \text{ and radius } r_0)$  **do**

**if**  $d(\mathbf{x}(\mathbf{k}), \mathbf{x}_s) < \mathcal{D}(\mathbf{k})$  **then**

$\mathcal{I}(\mathbf{k}) = s$

$\mathcal{D}(\mathbf{k}) = d(\mathbf{x}(\mathbf{k}), \mathbf{x}_s)$

**end**

**end**

**end**

Step 2:

**forall**  $\mathbf{k} \in \{1, \dots, n_1\} \times \{1, \dots, n_2\} \times \dots \times \{1, \dots, n_d\}$  **do**

**if**  $\mathcal{D}(\mathbf{k}) = \infty$  **then**

**forall**  $s = 1, 2, \dots, N_s$  **do**

**if**  $d(\mathbf{x}(\mathbf{k}), \mathbf{x}_s) < \mathcal{D}(\mathbf{k})$  **then**

$\mathcal{I}(\mathbf{k}) = s$

$\mathcal{D}(\mathbf{k}) = d(\mathbf{x}(\mathbf{k}), \mathbf{x}_s)$

**end**

**end**

**end**

**end**

---

With Voronoi tessellations, the conditions ensuring that  $v'_s = v_0$  (small balls compared to the domain size) are satisfied in most of the cases, and, in the periodic case, relation 11 becomes

$$n_{step\ 1} = N_v N_s v_0 . \quad (12)$$

If non-periodic conditions are assumed, one has only an upper bound,

$$n_{step\ 1} \leq N_v N_s v_0 , \quad (13)$$

which tends to be exact when the number of grains increases and thus when the ratio of spheres crossing the boundaries of  $\Omega$  decreases. In the following, expression 12 is used, as a fair approximation of the algorithm behaviour.

The number  $n_{step\ 2}$  of estimation of the distances in the second step of Algorithm 2 is given by

$$n_{step\ 2} = N_s n_{out} ,$$

$n_{out}$  being the number of voxels at a distance greater than  $r_0$  from all sites. The value of  $n_{out}$  depends on the set of sites, nevertheless, its statistics can be determined. When a site  $s$  is randomly placed in the volume, the probability for a given point of the domain to be outside the ball  $\mathcal{B}_s$  of radius  $r_0$  centred on the site is given by

$$p_s = 1 - \frac{v'_s}{V} . \quad (14)$$

In the same manner as above,  $v'_s$  is approximated by  $v'_s = v_0$  and thus

$$p_s = 1 - \frac{v_0}{V} . \quad (15)$$

As the positions of the  $N_s$  sites are supposed to be uncorrelated, the probability for a given point to remain outside the  $N_s$  balls of volume  $v_0$  centred on the sites, is

$$p_{N_s} = \prod_s p_s ,$$

where the notation  $\prod_s$  means the product for all  $s \in \mathcal{S}$ . Hence

$$p_{N_s} = \left(1 - \frac{v_0}{V}\right)^{N_s} .$$

Finally, the mean number of voxels outside all the balls is given by

$$E(n_{out}) = N_v \left(1 - \frac{v_0}{V}\right)^{N_s}$$

and the mean number of distance calculations in step 2 is

$$E(n_{step\ 2}) = N_s N_v \left(1 - \frac{v_0}{V}\right)^{N_s} .$$

Thus, the total number of distance calculations in Algorithm 2 is estimated by

$$E(n_{step\ 1+2}) = N_s N_v \left( \frac{v_0}{V} + \left(1 - \frac{v_0}{V}\right)^{N_s} \right) . \quad (16)$$

The optimal choice for  $v_0$ , i.e. the value of  $v_0$  which minimises  $E(n_{step\ 1+2})$ , is therefore easy to calculate:

$$\frac{v_0}{V} = 1 - \left( \frac{1}{N_s} \right)^{\frac{1}{N_s-1}} . \quad (17)$$

With this choice of  $v_0$  one has

$$E(n_{step\ 1+2}) = N_v N_s \left( 1 + \left( \frac{1}{N_s} \right)^{\frac{1}{N_s-1}} \left( \frac{1}{N_s} - 1 \right) \right) .$$

When the number  $N_s$  of sites is high, one has

$$n_{step\ 1+2} = N_v \ln N_s + N_v + o(N_v) . \quad (18)$$

Thus, the computational performance of the algorithm is in  $O(N_v \ln N_s)$ , which is significantly more efficient than the one of Algorithm 1 for which the total number of distance calculations is in  $O(N_v N_s)$ .

When different configurations are considered for which the number of sites  $N_s$  vary but the resolution, defined as the mean number of voxels per Voronoi cell, remains constant, the performance of the algorithm at constant resolution can be considered as being in  $O(N_s \ln N_s)$  while the performance of the brute force algorithm is in  $O(N_s^2)$ .

The number of distance calculations as a function of the radius  $r_0$ , deduced from relation 16, is plotted in Figure 2 and compared with the actual numbers of distance calculations made during an execution of the algorithm, applied to the creation of an image of  $200 \times 200 \times 200$  voxels of a Voronoi tessellation from 1000 given sites. The correspondence is significant and the minimum of the curve is well estimated by relation 17.

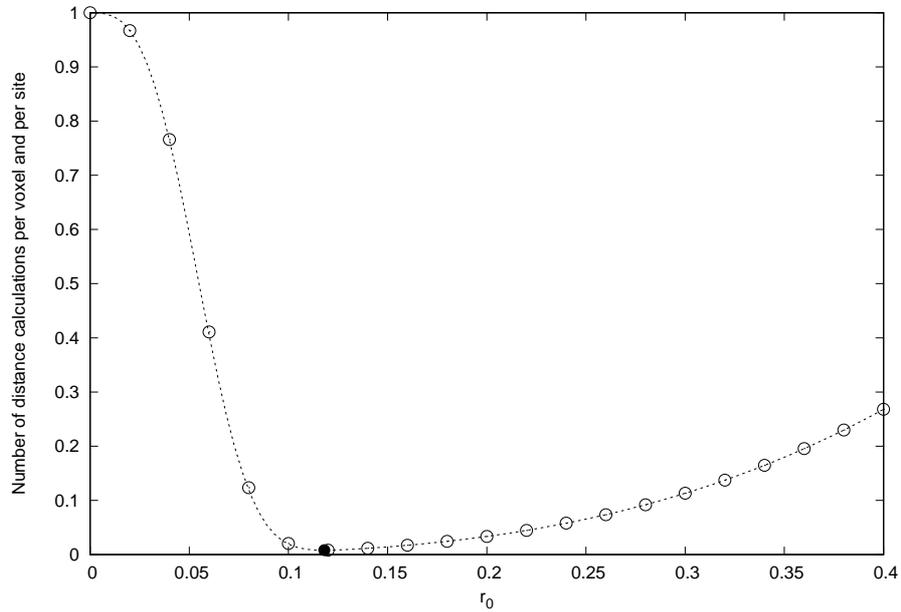


Figure 2: Generation of an image of  $200 \times 200 \times 200$  voxels of a Voronoi diagram of 1000 sites with Algorithm 2 : number of calculations of distances between voxels and sites per voxel and per site for different values of parameter  $r_0$  (radius of the balls of investigation around the sites). Empty circles: actual number of distance calculations during the execution of the algorithm. Dashed line: model given by relation 16. Black circle: optimal radius  $r_0 \approx 0.12$ , estimated using relation 17.

Expression 18, approximating the behaviour of the fast algorithm for large numbers of sites, is compared in Figure 3 with the actual number of distance calculations for the case of a  $1000 \times 1000 \times 1000$  images with varying numbers of cells, again leading to an appreciable agreement.

*Remark.* When the numbers of sites becomes very large, the expression 17 of volume  $v_0$  can be approximated by

$$\frac{v_0}{V} \approx \frac{\ln N_s}{N_s} .$$

Comparing it with the average volume of the cells  $\langle v_s \rangle = V/N_s$ , one has

$$\frac{v_0}{\langle v_s \rangle} \approx \ln N_s .$$

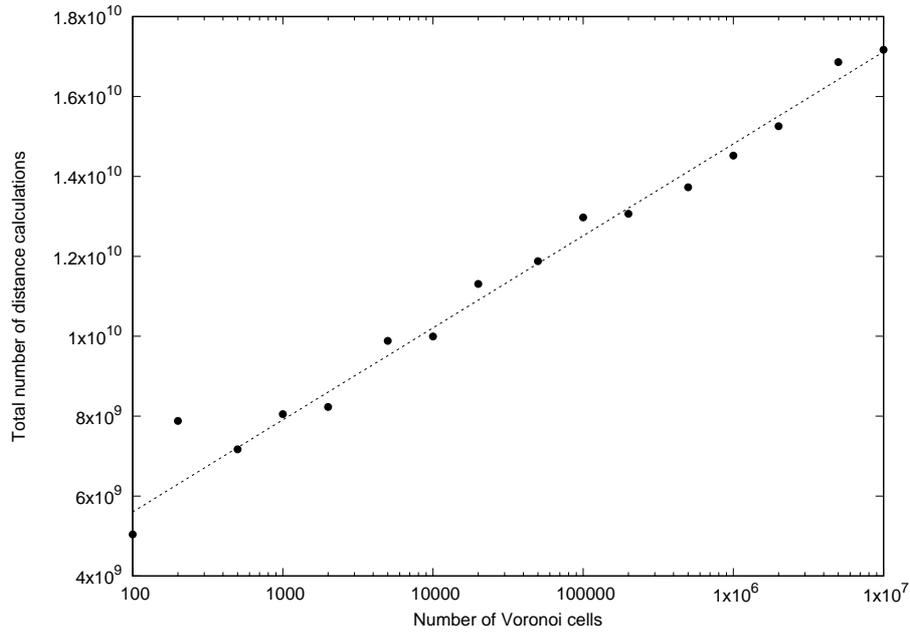


Figure 3: Number of calculations of distances between voxels and sites in the fast algorithm in the case of an image of  $1000 \times 1000 \times 1000$  voxels of a Voronoi tessellation for different numbers of sites  $N_s$ . Measure of the numbers of distance calculations actually performed during the execution of the algorithm (circles). Expression 18 approximating the behaviour of the algorithm for large numbers of cells (dashed line).

This ratio never stops increasing, which reflects the increasing computational weight of step 2 as  $N_s$  grows. This effect could be reduced by using a more efficient algorithm than the brute force algorithm in step 2. However, this point is developed in this article, for simplicity.

An example of realisation of microstructure with Algorithm 2 is presented in Figure 4.

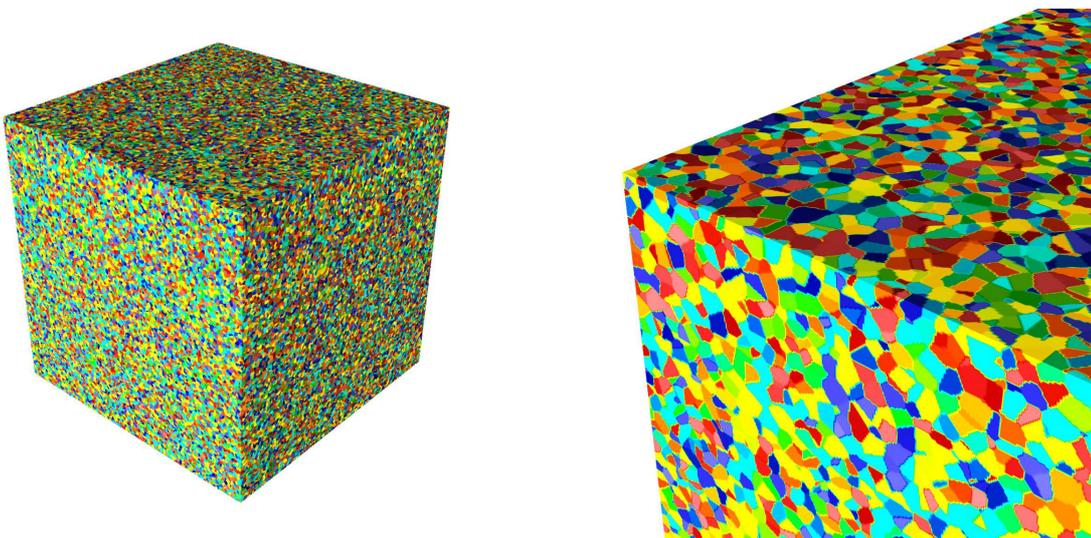


Figure 4: Image of  $1000 \times 1000 \times 1000$  voxels of a Voronoi tessellation of 1 000 000 punctual sites, generated by Algorithm 2 in 77.2 seconds on a 8-core processor Xeon Silver 4208 (2.10 GHz). The cells are plotted in arbitrary pseudo-colours. Left: general view of the volume. Right: detail.

### 3.3. An accelerated Johnson-Mehl tessellation algorithm

#### 3.3.1. Description of the algorithm

The principle of the accelerated algorithm remains the same for Johnson-Mehl tessellations, except that the quantity used to determine the site of the cell which a given voxel belongs to, is no longer the Euclidean distance, but the time  $t$  at which the growth of the crystal centred on the site reaches the point, given by relation 3, which can be recalled here:

$$t = d(\mathbf{x}, \mathbf{x}_s)/G + t_s .$$

The algorithm for the Johnson-Mehl tessellation thus resumes to the following: at a given time  $t_0$ , whose choice is discussed in section 3.3.2, the spherical region of the fictitious crystallisation process of each site is examined. The radius  $r_s$  of the ball of site  $s$  is given by

$$r_s = G(t_0 - t_s)^+ , \quad (19)$$

where  $(.)^+$  denotes the positive part. The voxels in the ball of radius  $r_s$  centred on the site  $x_s$ , i.e. the voxels satisfying

$$d(\mathbf{x}, \mathbf{x}_s) \leq r_s , \quad (20)$$

are examined and the time they are reached by the crystal growth is evaluated with relation 3.

After this step, there are two different cases for each voxel of the image:

- it belongs to one or more balls, i.e. relation 20 is satisfied for one or more sites, thus the voxel belongs to the cell of the one of these sites which minimises  $d(\mathbf{x}, \mathbf{x}_s)/G + t_s$ ,
- it belongs to no balls. In that case, the value of  $d(\mathbf{x}, \mathbf{x}_s)/G + t_s$  is calculated for all the sites and the voxel belongs to the cell of the site which minimises it.

The algorithm is summarised in Diagram 3. The differences with the algorithm for Voronoi tessellations are that the radii of the balls around the sites vary according to the sites, and that the quantity used to evaluate the proximity of a site is different.

#### 3.3.2. Optimal choice for $t_0$

The value of  $t_0$  must be chosen to minimise the computational cost of the algorithm, such as  $v_0$  in the algorithm for Voronoi tessellations.

The cost of the first step of the algorithm is proportional to the number of voxels in the balls centred on the sites, i.e.

$$n_{step\ 1} = N_v \frac{\sum_s v'_s}{V} , \quad (21)$$

where  $v'_s$  is defined as in the case of Voronoi tessellation: it is the volume of the intersection of the ball  $\mathcal{B}_s$  and of the domain  $\Omega_s$  ( respectively  $\Omega$ ) in the periodic case (respectively in the non-periodic case), except that the balls  $\mathcal{B}_s$  have radii  $r_s$  depending on the sites while they have an identical radius  $r_0$  in the Voronoi algorithm.

When a ball is smaller than the sphere inscribed in the domain, one has

$$v'_s = v_s ,$$

where  $v_s$  is the volume of the whole ball  $\mathcal{B}_s$  ( i.e. the volume of a ball of radius  $r_s$  of relation 19). When the ball encompasses the whole domain, one has

$$v'_s = V .$$

The exact expression of  $v'_s$ , taking into account the case when the ball is between these two extreme cases is uselessly complicated and can be approximated by

$$v'_s = \min(V, v_s) = V \min(1, \frac{v_s}{V}) . \quad (22)$$

It can in no way be assumed that the balls  $\mathcal{B}_s$  are all small and that  $v'_s = v_s$ . In contrast to Voronoi tessellations, a wide dispersion of size of balls can occur, and the case when some balls reach the size of the volume must be taken into account.

The cost of the first step of the algorithm thus reads

$$n_{step\ 1} = N_v \sum_s \min(1, \frac{v_s}{V}). \quad (23)$$

The cost of the second step depends on the number of voxels which have not been assigned to a cell in step 1, or, in other words, which are located outside all the investigation balls. The probability for a point  $\mathbf{x}$  to be outside a given ball  $\mathcal{B}_s$  is given by

$$p_s = 1 - v'_s/V,$$

which can be approximated, using relation 22, by

$$p_s = (1 - v_s/V)^+ . \quad (24)$$

Under the assumption that the position and size of the balls are uncorrelated, the probability for a given point  $\mathbf{x}$  to be outside all the balls is approximated by

$$\Pi_s(1 - \frac{v_s}{V})^+ . \quad (25)$$

The number of voxels that have to be taken into account in step 2 of the algorithm can therefore be estimated by

$$N_v \Pi_s(1 - \frac{v_s}{V})^+ , \quad (26)$$

and the mean of the cost of step 2 can be estimated by

$$E(n_{step\ 2}) = N_s N_v \Pi_s(1 - \frac{v_s}{V})^+ . \quad (27)$$

Finally, the estimation of the total cost of the algorithm is given by

$$E(n_{step\ 1+2}) = N_v \frac{\sum_s v'_s}{V} + N_s N_v \Pi_s(1 - \frac{v_s}{V})^+ , \quad (28)$$

with  $v'_s$  given by expression 22 .

The cost of step 1 is an increasing function of  $t_0$  which varies from 0 when  $t_0 = \min_s(t_s)$  to a maximum value when the radial growth of the first site to appear (i.e.  $\arg(\min_s(t_s))$ ) exceeds the size of the domain, i.e. at a time given by

$$t_0 = \min_s(t_s) + \frac{1}{2G} \sqrt{\sum_{i=1}^d L_i^2} ,$$

in the case of periodic conditions, or at a time  $t_0$  verifying

$$t_0 \leq \min_s(t_s) + \frac{1}{G} \sqrt{\sum_{i=1}^d L_i^2} ,$$

in the case of non-periodic conditions.

The cost of step 2 is a decreasing function of  $t_0$  that varies from  $N_s N_v$ , when  $t_0 = \min_s(t_s)$ , to 0 when  $t_0 = \min_s(t_s) + \frac{1}{2G} \sqrt{\sum_{i=1}^d L_i^2}$  in the periodic case, or at a time  $t_0$  lower than  $\min_s(t_s) + \frac{1}{G} \sqrt{\sum_{i=1}^d L_i^2}$  in the non-periodic case. Thus the optimal value for  $t_0$ , i.e. the value of  $t_0$  that minimises  $n_{step\ 1+2}$  given by relation 28, must be searched between these two extreme values, e.g. with a dichotomy method, for a negligible computational cost.

Figure 5 illustrates the computational cost of the algorithm in a three-dimensional example. An image of  $200 \times 200 \times 200$  voxels of a Johnson-Mehl tessellation with 10 000 sites is created. The slight discrepancy observed for high values of  $t_0$  between the model and the data, for  $G = 0.1$ , is most likely due to the approximation made by the model on the volume of the investigation balls when they cross the domain boundaries. This is consistent with the fact that the volume of the balls are overestimated by expression 22. Nevertheless, this does not affect the estimation of the optimal  $t_0$  which is very satisfactory.

---

**Algorithm 3:** Faster algorithm to create an image of a Johnson-Mehl tessellation

---

**input** :

- a table of  $N_s$  sites of positions and times  $\{(x_s, t_s)\}$  in space domain  $\Omega = [0, L_1) \times [0, L_2) \dots \times [0, L_d)$  and time interval  $[0, T)$ ,
- parameter  $G$ : “crystal growth rate”

**output** :

- an image  $\mathcal{I}(x)$  of  $\mathbf{n} = (n_1, n_2, \dots, n_d)$  voxels describing the Johnson-Mehl tessellation in space domain  $\Omega$

**temporary:**

- an image  $\mathcal{D}$  of same dimension as  $\mathcal{I}$

Initialisation:

$t_0$  is estimated using the method described in section 3.3.2

**forall**  $\mathbf{k} \in \{1, \dots, n_1\} \times \{1, \dots, n_2\} \times \{1, \dots, n_3\}$  **do**

  |  $\mathcal{D}(\mathbf{k}) = \infty$

**end**

Step 1:

**forall**  $s = 1, 2, \dots, N_s$  **do**

  | **forall**  $\mathbf{k} \mid \mathbf{x}(\mathbf{k}) \in (\text{Ball of center } x_s \text{ and radius } r_s = G(t_0 - t_s)^+)$  **do**

    | **if**  $d(\mathbf{x}(\mathbf{k}), \mathbf{x}_s)/G + t_s < \mathcal{D}(\mathbf{k})$  **then**

      |  $\mathcal{I}(\mathbf{k}) = s$

      |  $\mathcal{D}(\mathbf{k}) = d(\mathbf{x}(\mathbf{k}), \mathbf{x}_s)/G + t_s$

    | **end**

  | **end**

**end**

Step 2:

**forall**  $\mathbf{k} \in \{1, \dots, n_1\} \times \{1, \dots, n_2\} \times \dots \times \{1, \dots, n_d\}$  **do**

  | **if**  $\mathcal{D}(\mathbf{k}) = \infty$  **then**

    | **forall**  $s = 1, 2, \dots, N_s$  **do**

      | **if**  $d(\mathbf{x}(\mathbf{k}), \mathbf{x}_s)/G + t_s < \mathcal{D}(\mathbf{k})$  **then**

        |  $\mathcal{I}(\mathbf{k}) = s$

        |  $\mathcal{D}(\mathbf{k}) = d(\mathbf{x}(\mathbf{k}), \mathbf{x}_s)/G + t_s$

      | **end**

    | **end**

  | **end**

**end**

---

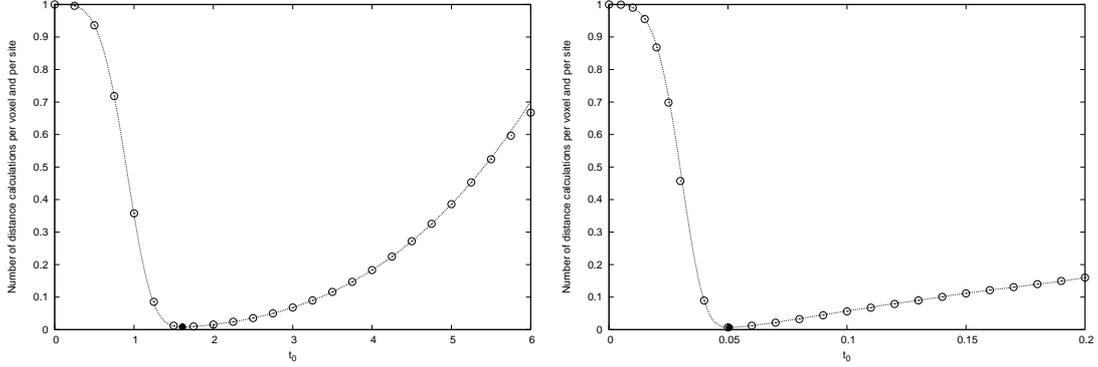


Figure 5: Generation of an image of  $200 \times 200 \times 200$  voxels of a Johnson-Mehl tessellation of 10000 sites. Number of calculations of distances between voxels and sites in Algorithm 3. Model given by relation 28 (continuous line) with varying values of  $t_0$ . Measure of the numbers of distance calculations actually carried out in a given realisation with the algorithm (empty circles). Left: growth rate  $G = 0.1$ , the optimal choice found with the algorithm is  $t_0 = 1.57$ , right: growth rate  $G = 10$ , the optimal choice found with the algorithm is  $t_0 = 0.051$ . Optimal value of  $t_0$  estimated following the method described in section 3.3.2 (black circle).

*Remark.* Before applying the algorithm, it can be useful to rearrange the list of the sites. Indeed, in some cases, some sites may result in the creation of no cell because they have been reached by the crystal growth of another site before they started to grow themselves. This happens typically when the growth rate  $G$  is large. To reduce the computation time, it is advantageous to discard these ineffective sites from the set in a preliminary step.

### 3.4. An accelerated Laguerre tessellation algorithm

The algorithm proposed for Laguerre tessellations is almost identical to the one of Johnson-Mehl tessellations, except that the investigations balls, corresponding to the crystal growth at a given time  $t_0$ , have radii given by

$$r_s = \sqrt{G(t_0 - t_s)^+} \quad (29)$$

and that the function to evaluate the proximity of a point  $\mathbf{x}$  to a site  $\mathbf{x}_s$ , is given by relation 8. Moreover, the optimal choice of  $t_0$  must be searched between  $\min_s(t_s)$  and  $\min_s(t_s) + \frac{1}{4G} \sum_{i=1}^d L_i^2$  in the periodic case,  $\min_s(t_s) + \frac{1}{G} \sum_{i=1}^d L_i^2$  in the non-periodic case.

An illustration of the behaviour of the algorithm is given in Figure 6. Similarly to the Johnson-Mehl algorithm, the optimal value of  $t_0$  is well estimated by the model, although a slight discrepancy (here for  $G = 0.1$ ) between the model and the data for high values of  $t_0$ , which is likely due to the approximation made in the model in relation 22.

An illustration of the performance of the algorithm as a function of the number of sites is presented in Figure 7.

## 4. Comparison with the Neper code

The performance of the algorithm proposed in this article is compared to that of the general code Neper (version 4.2.0.) which is widely used in the mechanics community involved in numerical simulations of crystalline materials. Neper is a versatile code developed by Romain Quey (Quey et al. (2011)) which enables to generate artificial polycrystalline microstructure of Voronoi type, and to create a mesh, typically to apply a finite element calculation, or to create an image of the tessellation, for example to visualise it or to apply a calculation using an FFT-based homogenisation method.

Neper, which offers many more general features than the algorithms proposed here, can be applied to the creation of rasterised images of tessellations. Schematically, in that case, Neper proceeds in two successive steps. In a first step, starting from the position of the sites of the Voronoi tessellation, the geometry of the tessellation is determined, i.e. the vertices, edges and facets of the polyhedral cells are determined and stored, using a so-called “cell-based algorithm”. In a second step, the image is created by a rasterisation process in which the cell to which each voxel of the image belongs is determined (Quey (2021b)).

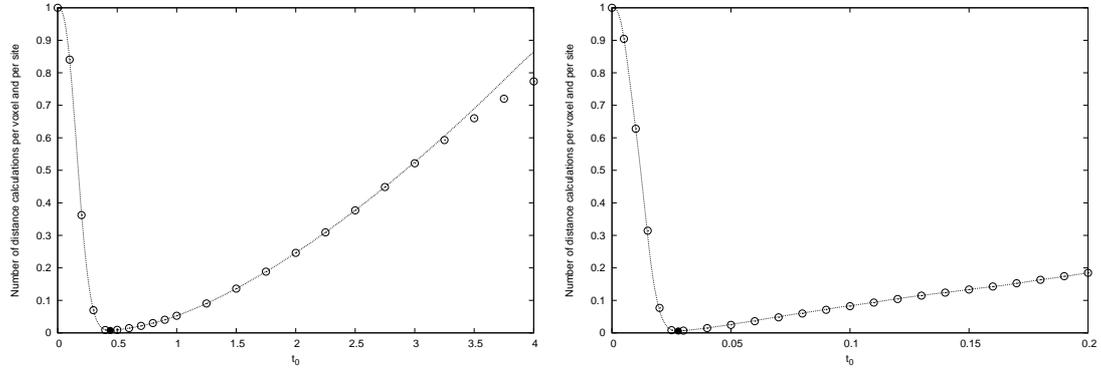


Figure 6: Generation of an image of  $200 \times 200 \times 200$  voxels of a Laguerre tessellation of 10000 sites. Number of calculations of distances between voxels and sites in the algorithm. Model giving by relation 28 and 29 (dashed line) with varying values of  $t_0$ . Measure of the numbers of distance calculations actually performed during the execution of the algorithm (empty circles). Left: growth rate  $G = 0.1$ , the optimal choice of  $t_0$  is  $t_0 = 0.426$  (black circle), right: growth rate  $G = 10$ , the optimal choice of  $t_0$  is  $t_0 = 0.028$  (black circle).

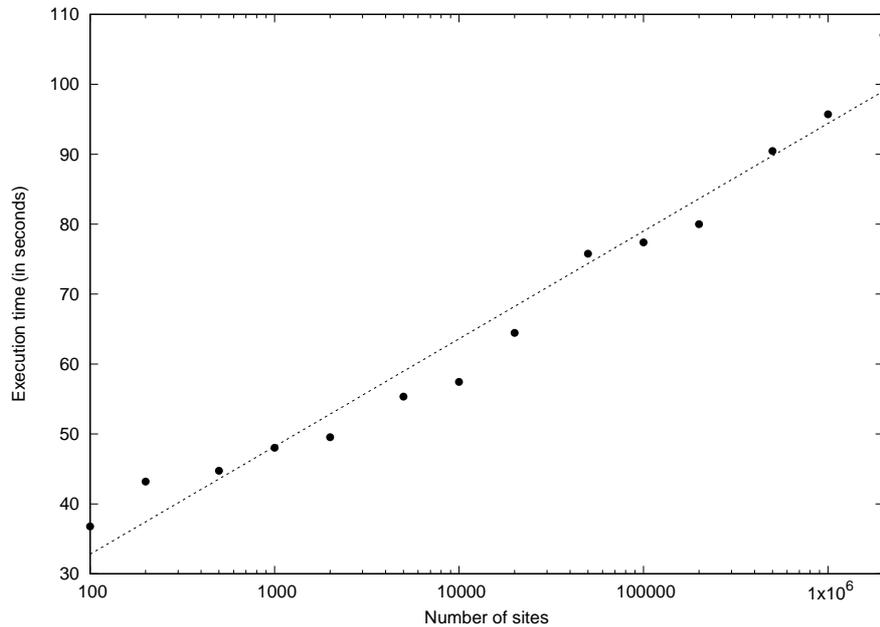


Figure 7: Total execution times  $t_e$  of the accelerated algorithm for the creation of an image of  $1000 \times 1000 \times 1000$  voxels of a Laguerre tessellation (radial growth  $G = 0.01$ ) for different numbers of sites  $N_s$  on a 8-core processor Xeon Silver 4208 (2.10 GHz) Dashed line: least square regression on the data ( $\ln N_s, t_e$ ).

The brute force algorithm described in section 3.1, the algorithm outlined in section 3.2 and the code Neper were applied to create a three-dimensional image of a Voronoi tessellation containing a number of sites varying from 100 to 2 000 000. In each case, the same set of sites was used by the three algorithms. The dimension of the images has been fixed to  $500 \times 500 \times 500$ , which is a realistic size for common calculations using an FFT-based homogenisation method. Figure 8 shows the execution times of the different algorithms, all three parallelised with OpenMP, on a 8-core processor Xeon Silver 4208 (2.10 GHz).

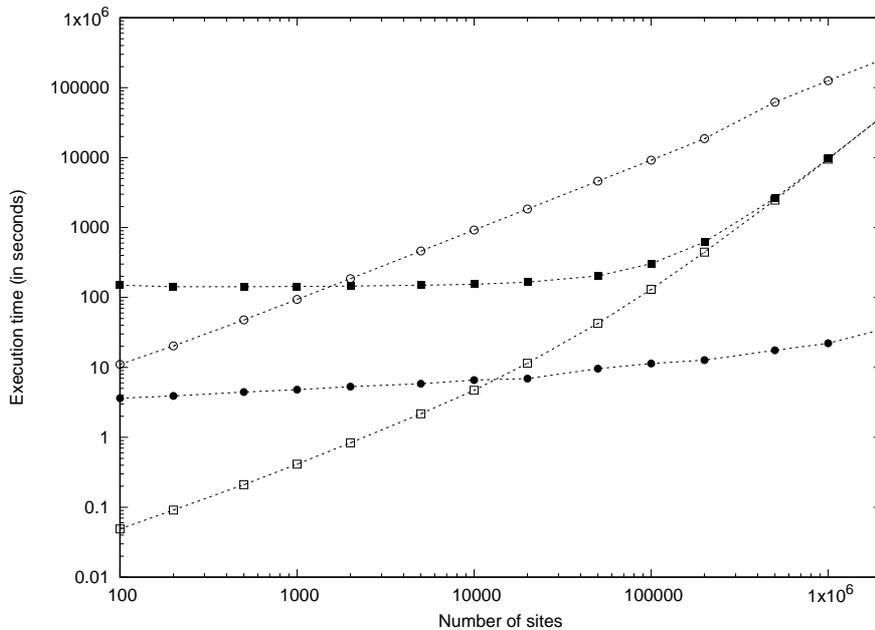


Figure 8: Code performance comparison on a 8-core processor Xeon Silver 4208 (2.10 GHz). Generation of an image of  $500 \times 500 \times 500$  voxels of a Voronoi tessellation with different numbers of sites. Total execution time in seconds for the brute force algorithm (empty circles), for the accelerated algorithm (black circles), for Neper code (black squares), for Neper code without image rasterisation (empty squares).

The code Neper was applied in two configurations depending on whether only the step of creating the geometrical description of the cells was carried out (empty squares on Figure 8), or whether this step was followed by the creation of an image by “rasterisation” (black squares on Figure 8). It can be seen that the time to create the image from the geometrical description of the cells remains approximately constant at around 140 seconds for the different site numbers, but that the time taken to create the geometrical description increases with the number of sites: beyond about 10 000 sites, this time alone exceeds the time it takes to execute the accelerated Algorithm 3.2.

Unsurprisingly, the brute force algorithm has low performance and the time it takes to run is proportional to the number of sites. Nevertheless, for a low number of sites (less than 1000), and in this specific case, it can compete with Neper.

The accelerated code is better in all cases and, as expected, takes an execution time of the order of the logarithm of the number of sites, which increases its relative performance as the number of sites increases. It took less than 30 seconds to create an image of 2 000 000 cells.

## 5. Conclusion

In this paper, an algorithm has been presented for creating digital images of Voronoi, Johnson-Mehl and Laguerre tessellations of punctual sites assumed to be randomly distributed in the domain considered. The principle of the algorithm is to assign the voxels belonging to balls centred around the sites of the tessellation, to the cell corresponding to the closest of these sites. The choice of the radii of these investigation balls plays a predominant role on the

performance of the algorithm. An analytical development enables to find a close to optimal choice, whose validity has been verified by numerical tests.

Incidentally, a relevant definition of the distance is introduced for the case of a periodic microstructure and a simple expression is given to calculate it.

### Available code

A beta version of the code applying the algorithms presented in this article is freely available on the site:  
<https://lma-software-craft.cnrs.fr/voronoi-johnson-mehl-and-laguerre-tessellations>.

### Acknowledgments

The author thanks the “Institut de Radioprotection et de Sûreté Nucléaire”, and especially Dr Pierre-Guy Vincent, who has supported the project for which the method proposed in this article has been developed.

The author also thanks Romain Quey for his interest in this study and for his pertinent advice.

### Appendix A. L-periodic distance

#### Appendix A.1. Periodicity

In the Euclidean space of dimension  $d$ , a given property  $f$  defined at each point  $\mathbf{x}$  is periodic of periods  $\mathbf{L}_i$ , with  $i = 1, 2, \dots, d$ , when

$$f(\mathbf{x}) = f\left(\mathbf{x} + \sum_{i=1}^d k_i \mathbf{L}_i\right) \quad \forall k_i \in \mathbb{Z}, \quad (\text{A.1})$$

where  $\mathbf{L}_i$  are given  $d$ -dimensional vectors supposed to be linearly independent of each other.

The property  $f$  is completely determined by its definition in the unit cell  $\mathcal{U}$

$$\mathcal{U} = \left\{ \mathbf{x} = \sum_{i=1}^d \alpha_i \mathbf{L}_i \quad \mid \quad \alpha_i \in [0, 1) \right\},$$

or in any translation of  $\mathcal{U}$  by a vector  $\mathbf{o}$ :

$$\mathcal{L} = \left\{ \mathbf{x} = \mathbf{o} + \sum_{i=1}^d \alpha_i \mathbf{L}_i \quad \mid \quad \alpha_i \in [0, 1) \right\}. \quad (\text{A.2})$$

#### Appendix A.2. Definition of the L-periodic distance

To introduce the definition of the L-periodic distance, the simple example of a two-dimensional microstructure containing two circular inclusions periodically repeated in a matrix is considered. An illustration of the resulting microstructure is presented in Figure A.9 where a period of the pattern is repeated three times in each direction. The left picture of Figure A.10 represents a unit cell of the microstructure. By comparison with the microstructure with the same two discs but without periodic conditions (Figure A.10 right), it can be seen that the parts of the discs that leave the image by one edge (here on the left and bottom edge) reappear on the opposite edge.

Thus, building an image of a unit cell under non-periodic conditions, consists in determining, for each of its voxels, if it belongs to a ball (or disc in 2D) by comparing the Euclidean distance between the voxel and the centre of the ball to the radius of the latter. In the case of periodic conditions, one has to compare the distance between the location  $\mathbf{x}$  of each voxel and every periodically repeated centres of the balls, to the radius. In other words, one has to evaluate

$$d'(\mathbf{x}, \mathbf{x}_s) = \min_{\mathbf{k} \in \mathbb{Z}^d} (d_E(\mathbf{x}, \mathbf{x}_s + \sum_{i=1}^d k_i \mathbf{L}_i)), \quad (\text{A.3})$$

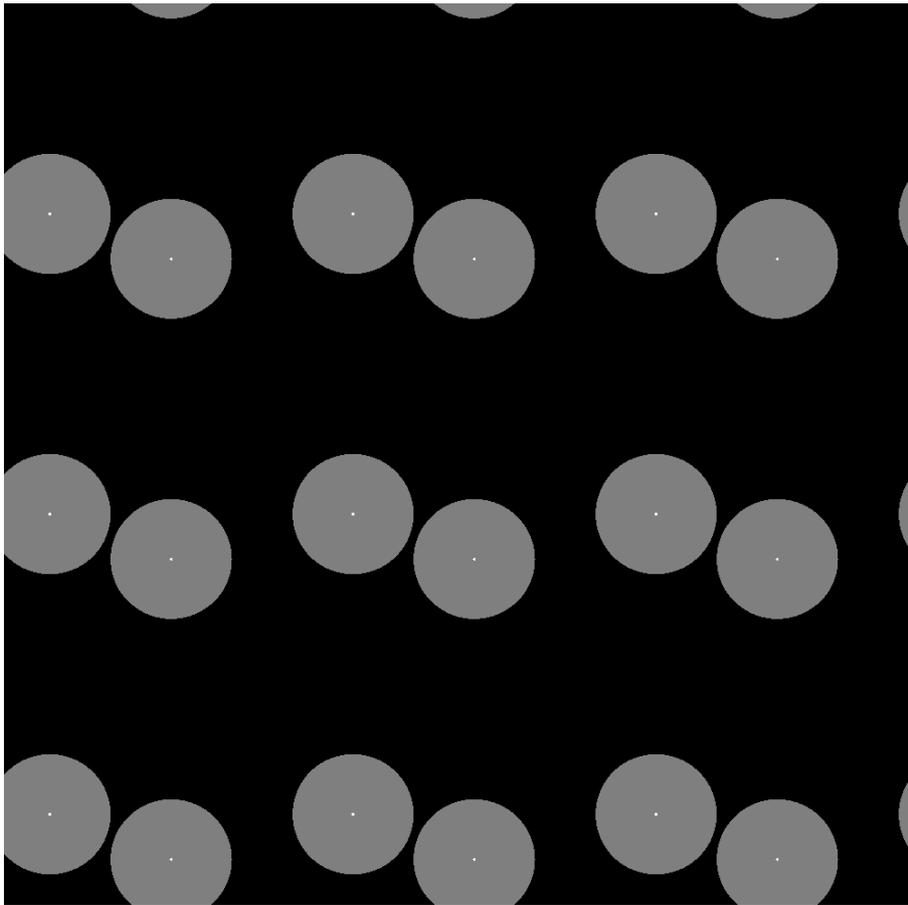


Figure A.9: Two-dimensional microstructure consisting in two similar discs (in gray) periodically repeated in a matrix (black). The centres of the discs are represented in small white circles. This figure reproduces  $3 \times 3$  periods of the pattern.

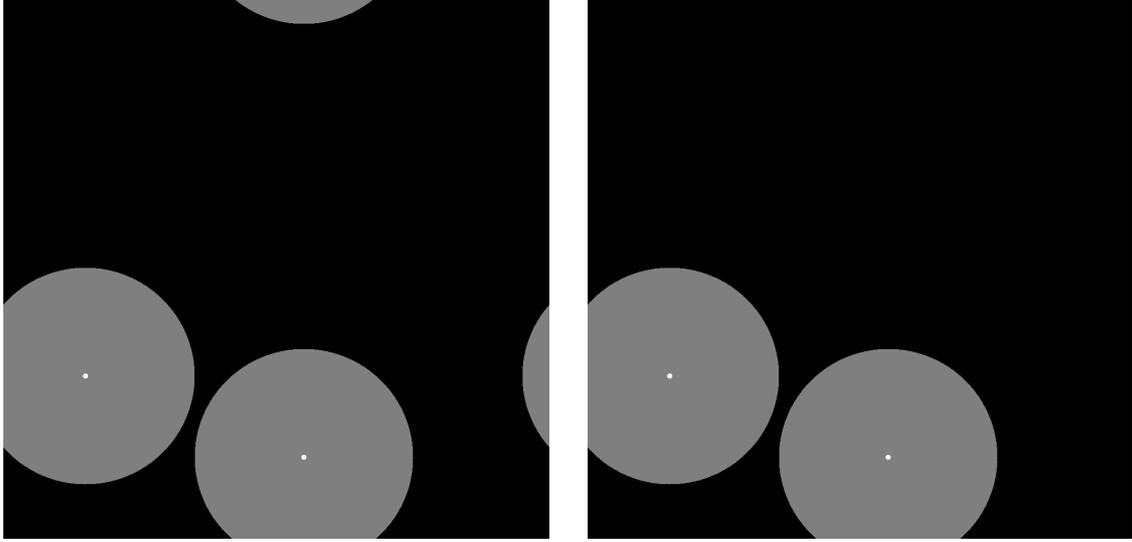


Figure A.10: Left: unit cell of the two-dimensional microstructure A.9. Right: image of the same discs *without* periodic conditions.

where  $\mathbf{x}$  denotes the position of the voxel and  $\mathbf{x}_s$  the one of the site or centre, where  $\mathbf{k}$  is a  $d$ -dimensional vector of integer components  $k_i$ , and where  $d_E$  denotes the usual Euclidean distance:

$$d_E(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}, \quad (\text{A.4})$$

where  $x_i$  and  $y_i$  are the components of vectors  $\mathbf{x}$  and  $\mathbf{y}$  in an orthonormal basis.

In the particular case where the sites are inside the unit cell and that the ball diameter is smaller than the dimensions of the unit cell, it is sufficient to estimate the distances between the voxels of the image of the unit cell and the ball centres which has been repeated by one period forward and backward in every directions (Fritzen et al. (2009), Bargmann et al. (2018) or Yan et al. (2011)), and one has

$$d'(\mathbf{x}, \mathbf{x}_s) = \min_{\mathbf{k} \in \{-1, 0, 1\}^d} (d_E(\mathbf{x}, \mathbf{x}_s + \sum_{i=1}^d k_i \mathbf{L}_i)) \quad (\text{A.5})$$

(each component  $k_i$  of  $\mathbf{k}$  only takes the value  $-1$ ,  $0$  or  $1$ ). Thus, to evaluate expression A.5, one has to calculate 9 Euclidean distances in a two-dimensional problem, and 27 Euclidean distances in a three-dimensional problem.

However, this approach does not work in the more general case where the points whose distance is to be calculated can be located outside the unit cell. This typically occurs in the case of a microstructure that contains inclusions whose size exceeds one of the dimensions of the unit cell (for example long fibers in a periodic microstructure).

Therefore, we define the  $L$ -periodic distance between two points of coordinates  $\mathbf{x}$  and  $\mathbf{y}$  of the Euclidean space - which can be located outside the unit cell - where the property considered is supposed to be periodic, with periods  $\mathbf{L}_i$ , with  $i = 1, 2, \dots, d$ , by

$$d_L(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} \min_{\mathbf{k} \in \mathbb{Z}^d, \mathbf{k}' \in \mathbb{Z}^d} (d_E(\mathbf{x} + \sum_{i=1}^d k_i \mathbf{L}_i, \mathbf{y} + \sum_{i=1}^d k'_i \mathbf{L}_i)). \quad (\text{A.6})$$

### Appendix A.3. Calculation of the $L$ -periodic distance

In this paragraph, the expression of the  $L$ -periodic distance is developed to obtain a formulation convenient to use in practice.

It is straightforward that relation A.6 simplifies into

$$d_L(\mathbf{x}, \mathbf{y}) = \min_{k \in \mathbb{Z}^d} \left( d_E(\mathbf{x}, \mathbf{y} + \sum_{i=1}^d k_i \mathbf{L}_i) \right).$$

In the case when the vectors  $\mathbf{L}_i$  are all orthogonal to each other, they can be noted

$$\mathbf{L}_i = L_i \mathbf{e}_i,$$

where  $L_i$  are positive scalars and  $\mathbf{e}_i$  are the unit vectors of an orthonormal basis. Hence,  $d_L$  can be written

$$d_L^2(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^d \min_{k_i \in \mathbb{Z}} (y_i - x_i + k_i L_i)^2,$$

where  $x_i$  and  $y_i$  are the components of  $\mathbf{x}$  and  $\mathbf{y}$  in the basis  $(\mathbf{e}_i)$ .

With no real difficulties, it can be established that

$$\min_{k_i \in \mathbb{Z}} (y_i - x_i + k_i L_i)^2 = \left( y_i - x_i - N\left(\frac{y_i - x_i}{L_i}\right) L_i \right)^2,$$

where  $N()$  is the “nearest integer function” defined as

$$N(x) = \lfloor x + \frac{1}{2} \rfloor$$

where  $\lfloor \cdot \rfloor$  denotes the “floor function”, that gives the greatest integer value lower or equal than its real number argument. The nearest integer function can be calculated using the function `rint()` of the C mathematical library.

Finally, the L-periodic distance  $d_L$  can be written

$$d_L(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^d \left( y_i - x_i - N\left(\frac{y_i - x_i}{L_i}\right) L_i \right)^2}. \quad (\text{A.7})$$

The numerical cost of the calculation of the L-periodic distance using expression A.7 is only slightly higher than the one of the Euclidean distance with expression A.4. In addition to being an exact method, it is very simple to implement and its numerical cost is much lower than evaluating the minimum of 27 (respectively 9) Euclidean distances in 3D (respectively 2D) problems, as proposed by Fritzen et al. (2009) and Bargmann et al. (2018).

## Appendix B. Alternative definition of Laguerre tessellations

As already presented in section 2.3, a Laguerre tessellation associated with the set of sites  $\mathcal{S} = \{(\mathbf{x}_s, r_s)\}_{s=1,2,\dots,N_s}$  where  $\mathbf{x}_s$  is the position of site  $s$  and  $r_s$  is an associated positive scalar, is a division of the domain  $\Omega$  of the Euclidean space into  $N_s$  cells defined by

$$C_s = \{\mathbf{x} \in \Omega \mid p_d(\mathbf{x}, s) \leq p_d(\mathbf{x}, s'), \forall s' \in \mathcal{S}\} \quad (\text{B.1})$$

with the so-called power-distance  $p_d$  being defined by

$$p_d(\mathbf{x}, s) = d^2(\mathbf{x}, \mathbf{x}_s) - r_s^2. \quad (\text{B.2})$$

For a given growth constant  $G$ , one can define a time  $t_s$  for each site  $s$  as

$$t_s = -r_s^2/G$$

and thus each cell  $C_s$  can be explicitated as

$$C_s = \{\mathbf{x} \in \Omega \mid d^2(\mathbf{x}, \mathbf{x}_s) + Gt_s \leq d^2(\mathbf{x}, \mathbf{x}_{s'}) + Gt_{s'}, \forall s' \in \mathcal{S}\}. \quad (\text{B.3})$$

The time  $t_s$ , at which a site “appears”, if one takes again the analogy with the crystallisation process, does not necessarily have to be negative if one remarks that the partition obtained does not depend on the choice of the time reference, i.e. the partition is unchanged if one considers time  $t' = t - t_{ref}$  instead of  $t$ , the power-distance being modified into

$$d^2(\mathbf{x}, \mathbf{x}_s) + Gt'_s = d^2(\mathbf{x}, \mathbf{x}_s) + Gt_s - Gt_{ref} \quad (\text{B.4})$$

because

$$\min_s(d^2(\mathbf{x}, \mathbf{x}_s) + Gt'_s) = \min_s(d^2(\mathbf{x}, \mathbf{x}_s) + Gt_s) - Gt_{ref}. \quad (\text{B.5})$$

The relation between the radius  $r_s$  and time  $t_s$  can then be expressed as

$$G(t_s - t_{ref}) = -r_s^2 \quad (\text{B.6})$$

with  $t_{ref}$  chosen adequately.

## References

- F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, Sept. 1991. ISSN 0360-0300. doi: 10.1145/116873.116880. URL <https://doi-org.insis.bib.cnrs.fr/10.1145/116873.116880>.
- S. Bargmann, B. Klusemann, J. Markmann, J. E. Schnabel, K. Schneider, C. Soyarslan, and J. Wilmers. Generation of 3d representative volume elements for heterogeneous materials: A review. *Progress in Materials Science*, 96:322–384, 2018. ISSN 0079-6425. doi: <https://doi.org/10.1016/j.pmatsci.2018.02.003>. URL <https://www.sciencedirect.com/science/article/pii/S0079642518300161>.
- G. Boittin, P.-G. Vincent, H. Moulinec, and M. Gărăjeu. Numerical simulations and modeling of the effective plastic flow surface of a biporous material with pressurized intergranular voids. *Computer Methods in Applied Mechanics and Engineering*, 323:174–201, 2017. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2017.05.004>. URL <https://www.sciencedirect.com/science/article/pii/S0045782516302705>.
- H. Breu, J. Gil, D. Kirkpatrick, and M. Werman. Linear-Time Euclidean Distance Transform Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):529–533, MAY 1995. ISSN 0162-8828. doi: {10.1109/34.391389}.
- Y. Chen, L. Gélébart, C. Chateau, M. Bornert, C. Sauder, and A. King. Analysis of the damage initiation in a sic/sic composite tube from a direct comparison between large-scale numerical simulation and synchrotron x-ray micro-computed tomography. *International Journal of Solids and Structures*, 161:111–126, 2019. ISSN 0020-7683. doi: <https://doi.org/10.1016/j.ijsostr.2018.11.009>. URL <https://www.sciencedirect.com/science/article/pii/S0020768318304554>.
- R. Fabbri, L. D. F. Costa, J. C. Torelli, and O. M. Bruno. 2D Euclidean distance transform algorithms: A comparative survey. *ACM COMPUTING SURVEYS*, 40(1), FEB 2008. ISSN 0360-0300. doi: {10.1145/1322432.1322434}.
- C. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32(6):1205–1227, 1991. doi: <https://doi.org/10.1002/nme.1620320604>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620320604>.
- F. Feyel. A multilevel finite element method (fe2) to describe the response of highly non-linear structures using generalized continua. *Computer Methods in Applied Mechanics and Engineering*, 192(28):3233–3244, 2003. ISSN 0045-7825. doi: [https://doi.org/10.1016/S0045-7825\(03\)00348-7](https://doi.org/10.1016/S0045-7825(03)00348-7). URL <https://www.sciencedirect.com/science/article/pii/S0045782503003487>. Multiscale Computational Mechanics for Materials and Structures.
- F. Fritzen, T. Böhlke, and E. Schnack. Periodic three-dimensional mesh generation for crystalline aggregates based on voronoi tessellations. *Computational Mechanics*, 43(5):701–713, Apr 2009. ISSN 1432-0924. doi: 10.1007/s00466-008-0339-2. URL <https://doi.org/10.1007/s00466-008-0339-2>.
- P. Gosselet and C. Rey. Non-overlapping domain decomposition methods in structural mechanics. *ARCHIVES OF COMPUTATIONAL METHODS IN ENGINEERING*, 13(4):515–572, 2006. ISSN 1134-3060. doi: {10.1007/BF02905857}.
- A. Gusev. Representative volume element size for elastic composites: A numerical study. *Journal of the Mechanics and Physics of Solids*, 45(9):1449–1459, 1997. ISSN 0022-5096. doi: [https://doi.org/10.1016/S0022-5096\(97\)00016-1](https://doi.org/10.1016/S0022-5096(97)00016-1). URL <https://www.sciencedirect.com/science/article/pii/S0022509697000161>.
- T. Hirata. A unified linear-time algorithm for computing distance maps. *INFORMATION PROCESSING LETTERS*, 58(3):129–133, MAY 13 1996. ISSN 0020-0190. doi: {10.1016/0020-0190(96)00049-X}.
- W. Johnson and R. Mehl. Reaction kinetics in processes of nucleation and growth. *TRANSACTIONS OF THE AMERICAN INSTITUTE OF MINING AND METALLURGICAL ENGINEERS*, 135:416–442, 1939.
- T. Kanit, S. Forest, I. Galliet, V. Mounoury, and D. Jeulin. Determination of the size of the representative volume element for random composites: statistical and numerical approach. *International Journal of Solids and Structures*, 40(13):3647–3679, 2003. ISSN 0020-7683. doi: [https://doi.org/10.1016/S0020-7683\(03\)00143-4](https://doi.org/10.1016/S0020-7683(03)00143-4). URL <https://www.sciencedirect.com/science/article/pii/S0020768303001434>.
- C. Lautensack and S. Zuyev. Random laguerre tessellations. *Advances in Applied Probability*, 40(3):630–650, 2008. doi: 10.1239/aap/1222868179.
- I. Lee, K. Lee, and C. Torpelund-Bruin. Raster voronoi tessellation and its application to emergency modeling. *Geo-spatial Information Science*, 14(4):235–245, 2011. doi: 10.1007/s11806-011-0569-x. URL <https://doi.org/10.1007/s11806-011-0569-x>.
- C. Li, J. Chen, and Z. Li. A raster-based method for computing Voronoi diagrams of spatial objects using dynamic distance transformation. *INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE*, 13(3):209–225, APR-MAY 1999. ISSN 1365-8816.
- A. Marano, L. Gélébart, and S. Forest. Fit-based simulations of slip and kink bands formation in 3d polycrystals: Influence of strain gradient crystal plasticity. *Journal of the Mechanics and Physics of Solids*, 149:104295, 2021. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2021.104295>. URL <https://www.sciencedirect.com/science/article/pii/S0022509621000041>.

- H. Moulinec and P. Suquet. A fast numerical-method for computing the linear and nonlinear mechanical-properties of composites. *C. R. Acad. Sc.*, II(318):1417–1423, 1994.
- H. Moulinec and P. Suquet. A numerical method for computing the overall response of nonlinear composites with complex microstructure. *Computer Methods in Applied Mechanics and Engineering*, 157(1):69–94, 1998. ISSN 0045-7825. doi: [https://doi.org/10.1016/S0045-7825\(97\)00218-1](https://doi.org/10.1016/S0045-7825(97)00218-1). URL <https://www.sciencedirect.com/science/article/pii/S0045782597002181>.
- J. Møller. Random johnson-mehl tessellations. *Advances in Applied Probability*, 24(4):814–844, 1992. doi: 10.2307/1427714.
- V. Müller, M. Kabel, H. Andrä, and T. Böhlke. Homogenization of linear elastic properties of short-fiber reinforced composites – a comparison of mean field and voxel-based methods. *International Journal of Solids and Structures*, 67-68:56–70, 2015. ISSN 0020-7683. doi: <https://doi.org/10.1016/j.ijsolstr.2015.02.030>. URL <https://www.sciencedirect.com/science/article/pii/S0020768315000761>.
- A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, 2nd edition, September 2000. ISBN 978-0-470-31785-3.
- W. K. Pratt. Image sampling and reconstruction. In *Digital Image Processing*, chapter 4, pages 91–126. John Wiley & Sons, Ltd, 2007. ISBN 9780470097434. doi: <https://doi.org/10.1002/9780470097434.ch4>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470097434.ch4>.
- R. Quey. Neper: Polycrystal generation and meshing, 2021a. URL <https://neper.info/>.
- R. Quey. Centre National de la Recherche Scientifique, September 2021b. personal communication.
- R. Quey and L. Renversade. Optimal polyhedral description of 3d polycrystals: Method and application to statistical and synchrotron x-ray diffraction data. *Computer Methods in Applied Mechanics and Engineering*, 330:308–333, 2018. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2017.10.029>. URL <https://www.sciencedirect.com/science/article/pii/S0045782517307028>.
- R. Quey, P. Dawson, and F. Barbe. Large-scale 3d random polycrystals for the finite element method: Generation, meshing and remeshing. *Computer Methods in Applied Mechanics and Engineering*, 200(17):1729–1745, 2011. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2011.01.002>. URL <https://www.sciencedirect.com/science/article/pii/S004578251100003X>.
- T. Saito and J. Toriwake. New Algorithms Euclidean Distance Transformation of an N-Dimensional Digitized Picture With Applications. *Pattern Recognition*, 27(11):1551–1565, NOV 1994. ISSN 0031-3203. doi: {10.1016/0031-3203(94)90133-3}.
- D. Stoyan, W. S. Kendall, and J. Mecke. *Stochastic geometry and its applications*. Wiley, 2d edition, 1995. ISBN 0-471-95099-8.
- P. Tallec, Y. Roeck, and M. Vidrascu. Domain decomposition methods for large linearly elliptic three-dimensional problems. *Journal of Computational and Applied Mathematics*, 34(1):93–117, 1991. ISSN 0377-0427. doi: [https://doi.org/10.1016/0377-0427\(91\)90150-I](https://doi.org/10.1016/0377-0427(91)90150-I). URL <https://www.sciencedirect.com/science/article/pii/037704279190150I>.
- T. van Nuland, J. van Dommelen, and M. Geers. An anisotropic voronoi algorithm for generating polycrystalline microstructures with preferred growth directions. *Computational Materials Science*, 186:109947, 2021. ISSN 0927-0256. doi: <https://doi.org/10.1016/j.commatsci.2020.109947>. URL <https://www.sciencedirect.com/science/article/pii/S0927025620304389>.
- M. Velić, D. May, and L. Moresi. A fast robust algorithm for computing discrete voronoi diagrams. *Journal of Mathematical Modelling and Algorithms*, 8(3):343–355, Aug 2009. ISSN 1572-9214. doi: 10.1007/s10852-008-9097-6. URL <https://doi.org/10.1007/s10852-008-9097-6>.
- P.-G. Vincent, H. Moulinec, L. Joëssel, M. I. Idiart, and M. Gărăjeu. Porous polycrystal plasticity modeling of neutron-irradiated austenitic stainless steels. *Journal of Nuclear Materials*, 542:152463, 2020. ISSN 0022-3115. doi: <https://doi.org/10.1016/j.jnucmat.2020.152463>. URL <https://www.sciencedirect.com/science/article/pii/S0022311520310710>.
- G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les paralléloèdres primitifs. *J. Reine Angew. Math.*, 134:198–287, 1908.
- K. Wojtacki, P.-G. Vincent, P. Suquet, H. Moulinec, and G. Boittin. A micromechanical model for the secondary creep of elasto-viscoplastic porous materials with two rate-sensitivity exponents: Application to a mixed oxide fuel. *International Journal of Solids and Structures*, 184:99–113, 2020. ISSN 0020-7683. doi: <https://doi.org/10.1016/j.ijsolstr.2018.12.026>. URL <https://www.sciencedirect.com/science/article/pii/S0020768318305237>. Physics and Mechanics of Random Structures: From Morphology to Material Properties.
- D.-M. Yan, K. Wang, B. Levy, and L. Alonso. Computing 2d periodic centroidal voronoi tessellation. In *2011 Eighth International Symposium on Voronoi Diagrams in Science and Engineering*, pages 177–184, 2011. doi: 10.1109/ISVD.2011.31.