

# Laguerre tessellations and polycrystalline microstructures: A fast algorithm for generating grains of given volumes

D.P. Bourne<sup>1</sup>, P.J.J. Kok<sup>2</sup>, S.M. Roper<sup>3</sup>, W.D.T. Spanjer<sup>2</sup>

<sup>1</sup>Department of Mathematics, Heriot-Watt University, Edinburgh Campus, Edinburgh, EH14 4AS, UK

<sup>2</sup>Tata Steel Research and Development, P.O. Box 10000, 1970 CA IJmuiden, The Netherlands

<sup>3</sup>School of Mathematics and Statistics, University of Glasgow, University Place, Glasgow, G12 8QQ, UK

June 22, 2020

## Abstract

We present a fast algorithm for generating Laguerre diagrams with cells of given volumes, which can be used for creating RVEs of polycrystalline materials for computational homogenisation, or for fitting Laguerre diagrams to EBSD or XRD measurements of metals. Given a list of desired cell volumes, we solve a convex optimisation problem to find a Laguerre diagram with cells of these volumes, up to any prescribed tolerance. The algorithm is built on tools from computational geometry and optimal transport theory which, as far as we are aware, have not been applied to microstructure modelling before. We illustrate the speed and accuracy of the algorithm by generating RVEs with user-defined volume distributions with up to 20,000 grains in 3D. We can achieve volume percentage errors of less than 1% in the order of minutes on a standard desktop PC. We also give examples of polydisperse microstructures with bands, clusters and size gradients, and of fitting a Laguerre diagram to 3D EBSD measurements of an IF steel.

**Keywords:** Laguerre diagrams; power diagrams; polycrystalline materials; grains; foams; volume distribution

## 1 Introduction

### 1.1 State of the art

Voronoi diagrams and their generalizations are often used to represent the microstructure of polycrystalline metals and foams, e.g., [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], with individual Voronoi cells representing grains in metals and pores or bubbles in foams. They can be used to generate complex microstructures quickly using a relatively small number of parameters, and they are often used as representative volume elements (RVEs) for computational homogenisation, e.g., [15, 16, 17].

In this paper we focus on the class of weighted Voronoi diagrams known as *Laguerre diagrams* (or *power diagrams* or *radical Voronoi tessellations*), which provide a more accurate description of the geometry of polycrystalline materials than classical Voronoi diagrams [14, 8]. However, Laguerre diagrams share the limitation of Voronoi diagrams that there is not an explicit relation between their generators and their geometric properties, such as the volumes of their cells. Consequently an active area of research is to develop algorithms for generating Laguerre diagrams with prescribed geometric properties.

One popular method for approximately controlling the grain size distribution of Laguerre diagrams is using random close packing of spheres and ellipsoids [4, 8, 10, 18]. This method is inexact, however, since it is impossible to tessellate Euclidean space with spheres or ellipsoids.

Several authors have developed methods for fitting Laguerre diagrams to image data (EBSD, XRD) of polycrystals. Here geometric properties such as grain size, centroid location, and aspect

ratio are fitted by minimising a measure of the fitting error using deterministic and stochastic optimisation methods, e.g., [3, 9, 11, 19, 20]. While optimisation methods can give very accurate results, they can also be computationally expensive. Heuristic methods such as [13] trade off fidelity against speed.

Recently several authors have generated RVEs with curved boundaries and non-convex cells grains using generalisations of Laguerre diagrams such as generalised balanced power diagrams [1, 12, 13, 21] and multilevel Voronoi diagrams [5, 17]. Complex geometries can also be created using the open-source software DREAM.3D [22] and Neper [23].

We will discuss some of these methods in more detail and compare them with ours in the Discussion section.

## 1.2 Goal of this paper

The goal of this paper is to develop algorithms for creating Laguerre diagrams with user-defined cell size distributions. Our motivation comes from steel modelling. We wish to generate realistic RVEs of single- and multi-phase steels for computational homogenisation simulations. Unlike much of the literature on Laguerre modelling of polycrystals [1, 24, 12, 13], our primary aim is not to fit Laguerre diagrams to EBSD or XRD data, but rather to create a tool for generating a rich family of (possibly never-observed) microstructures, which can be combined with multiscale simulations to optimise grain geometries and lead to the development of new alloys. Having said that, our algorithms are also very well suited for generating Laguerre diagrams with texture intensities that match EBSD data, as we demonstrate in Example 5.3. With these applications to steel in mind, we often refer to Laguerre cells as grains, although our results can be applied more generally to other polycrystalline metals and to foams.

## 1.3 Contributions and outline of the paper

In Section 2 we recall the definition and some important properties of Laguerre diagrams. In particular Property 2.3 forms the basis of our work. Section 3 includes our main result, Algorithm 2, for generating ‘regularised’ Laguerre diagrams with grains of prescribed volumes, up to any given tolerance. We also provide Algorithm 1 that can be used for fitting a Laguerre diagram to EBSD measurements of grain volumes and centroids (Example 5.3). We discuss practical issues about how the algorithms can be implemented in Section 4. Section 5 includes some large examples (10,000+ grains) and run time tests in 3D, including examples of RVEs of Interstitial Free (IF) steels.

The theory underlying the algorithms presented in Section 2 uses results from computational geometry and *optimal transport theory* [25, 26], a field of mathematics that has recently enormously grown in importance and found applications in a wide range of areas including data science, economics, image processing, partial differential equations and statistics. We believe however that this is its first application in the steel industry.

# 2 Laguerre diagrams

## 2.1 Notation and definitions

Let  $\Omega \subset \mathbb{R}^d$  be the region occupied by a metal. We consider both the 2- and 3-dimensional cases ( $d = 2$  and  $d = 3$ ). For simplicity we assume that  $\Omega$  is a convex polygon if  $d = 2$  or a convex polyhedron if  $d = 3$ . In principle the algorithms below can be used for non-convex regions with curved boundaries, but they become harder to implement. In all our examples below we take  $\Omega$  to be a rectangular box. If  $U$  is a subset of  $\Omega$ , let  $|U|$  denote its area if  $d = 2$  or its volume if  $d = 3$ .

**Definition 2.1** ([27, 28]). Let  $\mathbf{x}_1, \dots, \mathbf{x}_n$  be distinct points in  $\Omega$  and  $w_1, \dots, w_n$  be real numbers (not necessarily positive). The *Laguerre diagram* or *power diagram* generated by the weighted points  $(\mathbf{x}_1, w_1), \dots, (\mathbf{x}_n, w_n)$  is the tessellation  $\{L_i\}_{i=1}^n$  of  $\Omega$  defined by

$$L_i = \{\mathbf{x} \in \Omega : |\mathbf{x} - \mathbf{x}_i|^2 - w_i \leq |\mathbf{x} - \mathbf{x}_j|^2 - w_j \forall j \in \{1, \dots, n\}\}. \quad (2.1)$$

We refer to the sets  $L_i$  as *Laguerre cells* or *grains*.

Laguerre diagrams have the following basic properties [27, 28]:

- Laguerre cells are convex polygons if  $d = 2$  or convex polyhedra if  $d = 3$ .
- The Laguerre cells tessellate  $\Omega$ , which means that  $\bigcup_{i=1}^n L_i = \Omega$  and cells can only intersect along their boundaries.
- If all the weights are equal,  $w_1 = w_2 = \dots = w_n$ , then the Laguerre diagram is simply a Voronoi diagram.
- Adding a constant to all the weights does not affect the Laguerre diagram, i.e., the weighted points  $\{(\mathbf{x}_i, w_i)\}_{i=1}^n$  and  $\{(\mathbf{x}_i, w_i + c)\}_{i=1}^n$  generate the same diagram for any  $c \in \mathbb{R}$ .
- A generator  $\mathbf{x}_i$  needs not belong to its Laguerre cell  $L_i$ .
- There can be empty Laguerre cells,  $L_j = \emptyset$  for some  $j$ .

Now we recall two advanced properties of Laguerre diagrams, Properties 2.2 and 2.3. These are the key ingredients for generating RVEs with grains of given sizes (given areas if  $d = 2$  or given volumes if  $d = 3$ ). Property 2.2 states that there always exists a Laguerre diagram with grains of given sizes. Property 2.3 gives a constructive way of finding one.

**Property 2.2** ([27, p. 96, Corollary 6.1], [29]). *Let  $\mathbf{x}_1, \dots, \mathbf{x}_n$  be distinct points in  $\Omega$ . Let  $m_1, \dots, m_n$  be positive numbers with  $\sum_{i=1}^n m_i = |\Omega|$ . Then there exist weights  $w_1, \dots, w_n$  such that the Laguerre diagram  $\{L_i\}_{i=1}^n$  generated by  $(\mathbf{x}_1, w_1), \dots, (\mathbf{x}_n, w_n)$  has cells of size  $m_1, \dots, m_n$ :*

$$|L_i| = m_i \quad \text{for all } i \in \{1, \dots, n\}. \quad (2.2)$$

The weights  $w_i$  in Property 2.2 can be computed using the following result:

**Property 2.3** ([27, pp. 98-100], [29], [30, Theorem 2]). *Let  $\mathbf{x}_1, \dots, \mathbf{x}_n$  be distinct points in  $\Omega$ . Let  $m_1, \dots, m_n$  be positive numbers with  $\sum_{i=1}^n m_i = |\Omega|$ . Define the function  $g: \mathbb{R}^n \rightarrow \mathbb{R}$  by*

$$g(w_1, \dots, w_n) = \sum_{i=1}^n (m_i - |L_i|)w_i + \sum_{i=1}^n \int_{L_i} |\mathbf{x} - \mathbf{x}_i|^2 d\mathbf{x} \quad (2.3)$$

where  $\{L_i\}_{i=1}^n$  is the Laguerre diagram generated by  $(\mathbf{x}_1, w_1), \dots, (\mathbf{x}_n, w_n)$ . Then

- (i) *The function  $g$  is concave.*
- (ii) *The gradient of  $g$  has components*

$$\frac{\partial g}{\partial w_i} = m_i - |L_i|. \quad (2.4)$$

for all  $i \in \{1, \dots, n\}$ .

- (iii) *If  $\mathbf{w} = (w_1, \dots, w_n)$  is a critical point of  $g$ , i.e., if  $\nabla g(\mathbf{w}) = \mathbf{0}$ , then the Laguerre diagram  $\{L_i\}_{i=1}^n$  generated by  $(\mathbf{x}_1, w_1), \dots, (\mathbf{x}_n, w_n)$  has cells of size  $m_1, \dots, m_n$ :*

$$|L_i| = m_i \quad \text{for all } i \in \{1, \dots, n\}. \quad (2.5)$$

Property 2.3 forms the basis of Algorithms 1 and 2. It means that if we want to generate a Laguerre diagram with grains of given sizes, then we just need to find critical points of  $g$ . Since  $g$  is concave, this is equivalent to maximising  $g$ , or to minimising  $-g$ , which is a smooth, unconstrained, convex optimisation problem. Fast numerical methods are available for solving this [31].

## 2.2 Controlling the spatial distribution of grains

Property 2.3 not only allows one to control the size distribution of grains, it also gives some control over the spatial distribution. Given positive numbers  $m_1, \dots, m_n$  with  $\sum_{i=1}^n m_i = |\Omega|$ , there are infinitely many Laguerre diagrams  $\{L_i\}_{i=1}^n$  such that  $|L_i| = m_i$  for all  $i \in \{1, \dots, n\}$ . This can be seen from Property 2.3(iii); any choice of distinct points  $\mathbf{x}_1, \dots, \mathbf{x}_n$  can give a Laguerre diagram with grains of size  $m_1, \dots, m_n$ . In Section 4.3.1 we will show how to choose  $\mathbf{x}_1, \dots, \mathbf{x}_n$  to control the spatial distribution of the grains.

## 2.3 Connection with optimal transport theory

Properties 2.2 and 2.3 can also be stated in the language of semi-discrete optimal transport theory<sup>1</sup>; see, e.g., [26, Sec. 6.4.2], [32], [30]. This connection provides a way of finding critical points of  $g$  using fast modern methods from optimal transport theory [33, 32]. We discuss this connection further at the end of Section 3.2.3.

## 3 Main results

### 3.1 Statement of the algorithms

For concreteness we state the algorithms in three dimensions, but they can also be used in two dimensions (by substituting *volume* with *area* and *polyhedron* with *polygon* wherever they appear in Algorithms 1 and 2). Our main result is Algorithm 2. First however we consider a simplified version, Algorithm 1, which will help us to understand the importance of the regularisation step in Algorithm 2. Algorithm 1 can also be used for data-driven modelling to fit a Laguerre diagram to EBSD or XRD measurements of grain volumes and centroids (see Example 5.3). Algorithm 1 is not new and goes back at least as far as [29]. Our role is simply to bring it to the attention of the microstructure modelling community.

---

**Algorithm 1** Generate a Laguerre diagram with grains of given volumes

---

**Input:** A convex polyhedron  $\Omega$  representing a sample of metal, a list of volumes  $m_1, \dots, m_n$  such that  $m_i > 0$  and  $\sum_{i=1}^n m_i = |\Omega|$ , and a relative error tolerance  $\varepsilon$ .

**Output:** The generators  $(\mathbf{x}_1, w_1), \dots, (\mathbf{x}_n, w_n)$  of a Laguerre diagram  $\{L_i\}_{i=1}^n$  such that grain  $L_i$  has volume  $m_i$  up to  $\varepsilon$  relative error, i.e.,  $\frac{||L_i| - m_i|}{m_i} < \varepsilon$ , for all  $i \in \{1, \dots, n\}$ .

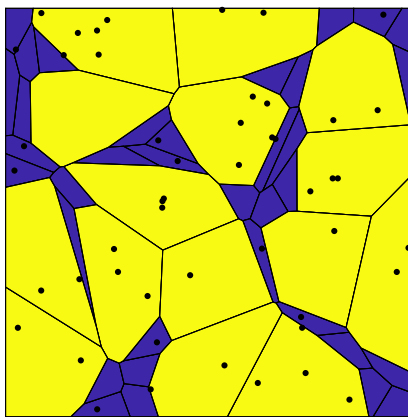
**Method:**

*Initialisation.* Choose or randomly select  $n$  distinct points  $\mathbf{x}_1, \dots, \mathbf{x}_n$  in  $\Omega$ .

*Optimisation step.* Use a numerical optimisation method to find  $\mathbf{w} = (w_1, \dots, w_n)$  that maximises the function  $g$  defined in equation (2.3). Initialise the optimisation method using the initial guess  $\mathbf{w}_{\text{init}} = \mathbf{0}$  and terminate it using the stopping criterion  $|\nabla g(\mathbf{w})| < \varepsilon \min_j m_j$ .

---

**Example 3.1** (Example of Algorithm 1). Figure 1 shows an example of Algorithm 1 implemented in MATLAB with  $n = 50$  grains in the square domain  $\Omega = [0, 1] \times [0, 1]$ .



**Figure 1:** An example of Algorithm 1 with  $n = 50$  grains in the unit square  $\Omega$ . There are 35 blue grains and 15 yellow grains. The blue grains have area  $x$  and the yellow grains have area  $10x$  to within 1% error, where  $x = 1/185$ . The black dots are the locations of the generators  $\{\mathbf{x}_i\}_{i=1}^{50}$ . Notice that not every grain contains its own generator.

<sup>1</sup>Technical remark: It can be shown that evaluating the optimal transport (Wasserstein) distance  $W_2(\mathcal{L}^d \llcorner \Omega, \sum_i m_i \delta_{\mathbf{x}_i})$  between the Lebesgue measure and a discrete measure generates a partition of  $\Omega$  into Laguerre cells of size  $m_1, \dots, m_n$ .

The grains have target areas  $m_i = x$  for  $i \in \{1, \dots, 35\}$  and  $m_i = 10x$  for  $i \in \{36, \dots, 50\}$ , where  $x = 1/185$  so that the total area of all the grains equals the area of  $\Omega$ . The actual areas of the grains returned by Algorithm 1 are correct to within 1% error ( $\varepsilon = 0.01$ ). The initialisation step of Algorithm 1 was performed using the MATLAB function *rand* to select  $\mathbf{x}_1, \dots, \mathbf{x}_{50}$  (pseudo)randomly from a uniform distribution. While the grains have the correct areas to within 1%, the microstructure is somewhat irregular and unrealistic, with some highly elongated grains. This leads us to Algorithm 2, which produces more regular microstructures; compare Figures 1 and 2(i).

---

**Algorithm 2** Generate a regularised Laguerre diagram with grains of given volumes

---

**Input:** A convex polyhedron  $\Omega$  representing a sample of metal, a list of volumes  $m_1, \dots, m_n$  such that  $m_i > 0$  and  $\sum_{i=1}^n m_i = |\Omega|$ , a relative error tolerance  $\varepsilon$ , and the number of regularisation steps  $K$ .

**Output:** The generators  $(\mathbf{x}_1, w_1), \dots, (\mathbf{x}_n, w_n)$  of a regularised Laguerre diagram  $\{L_i\}_{i=1}^n$  such that grain  $L_i$  has volume  $m_i$  up to  $\varepsilon$  relative error, i.e.,  $\frac{||L_i| - m_i|}{m_i} < \varepsilon$ , for all  $i \in \{1, \dots, n\}$ .

**Method:**

*Initialisation.* Choose or randomly select  $n$  distinct points  $\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_n^{(0)}$  in  $\Omega$ . Initialise the weights to be zero:  $\mathbf{w}^{(0)} = (w_1^{(0)}, \dots, w_n^{(0)}) = \mathbf{0}$ .

*Iteration.* For  $k = 1, \dots, K$  do:

1. *Regularisation step.* For  $i \in \{1, \dots, n\}$ , define  $\mathbf{x}_i^{(k)}$  to be the centroid of  $L_i^{(k-1)}$ :

$$\mathbf{x}_i^{(k)} := \frac{1}{|L_i^{(k-1)}|} \int_{L_i^{(k-1)}} \mathbf{x} \, d\mathbf{x} \quad (3.1)$$

where  $\{L_i^{(k-1)}\}_{i=1}^n$  is the Laguerre diagram obtained in the previous iteration, which is generated by  $(\mathbf{x}_1^{(k-1)}, w_1^{(k-1)}), \dots, (\mathbf{x}_n^{(k-1)}, w_n^{(k-1)})$ .

2. *Optimisation step.* Use a numerical optimisation method to find  $\mathbf{w}^{(k)} = (w_1^{(k)}, \dots, w_n^{(k)})$  that maximises the concave function

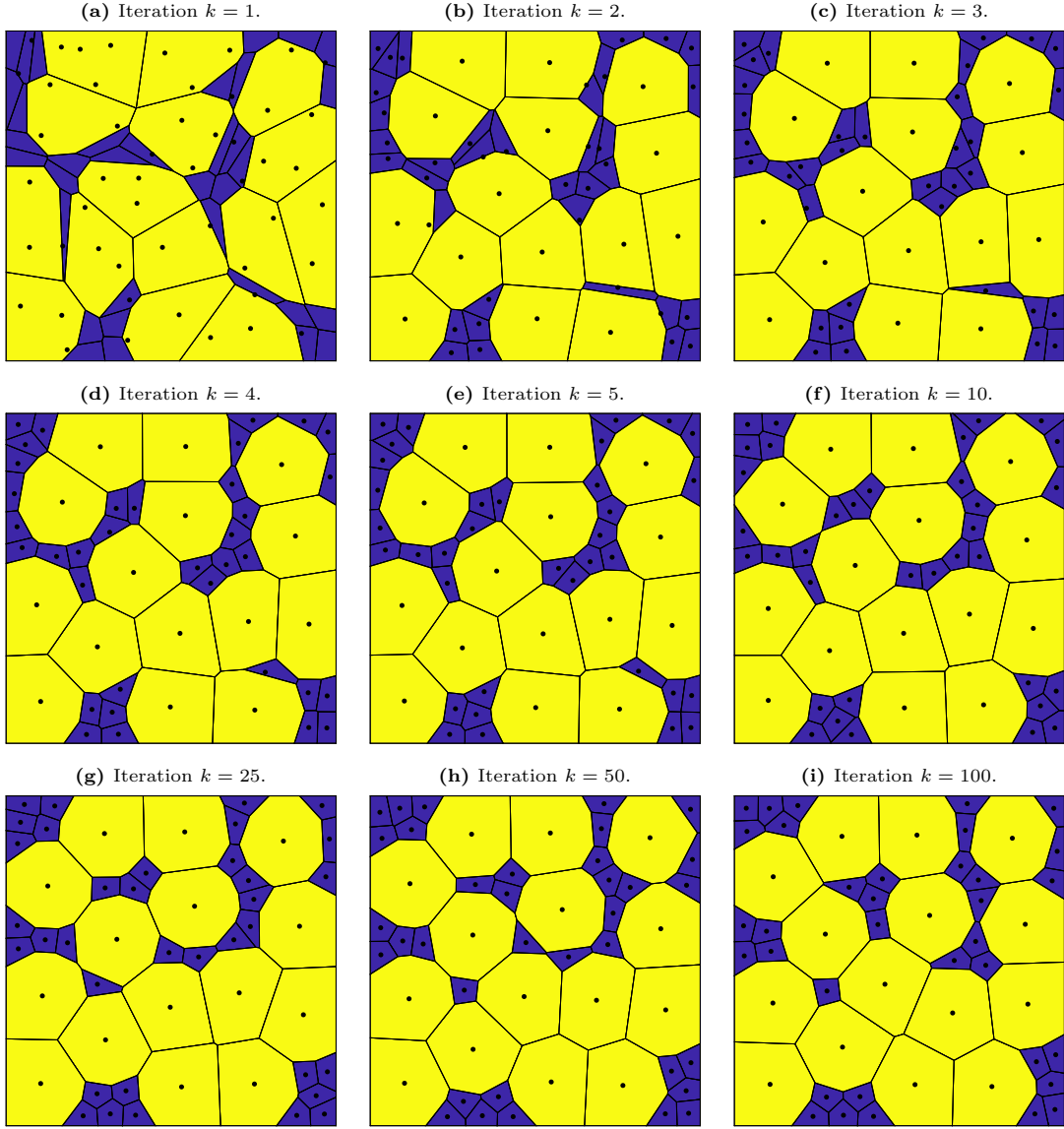
$$g_k(w_1, \dots, w_n) = \sum_{i=1}^n (m_i - |L_i|)w_i + \sum_{i=1}^n \int_{L_i} |\mathbf{x} - \mathbf{x}_i^{(k)}|^2 \, d\mathbf{x} \quad (3.2)$$

where  $\{L_i\}_{i=1}^n$  is the Laguerre diagram generated by  $(\mathbf{x}_1^{(k)}, w_1), \dots, (\mathbf{x}_n^{(k)}, w_n)$ . Initialise the optimisation method using the initial guess  $\mathbf{w}_{\text{init}} = \mathbf{w}^{(k-1)}$  and terminate it using the stopping criterion  $|\nabla g_k(\mathbf{w}^{(k)})| < \varepsilon \min_j m_j$ .

*Return.* Output the generators  $\{(\mathbf{x}_i, w_i)\}_{i=1}^n = \{(\mathbf{x}_i^{(K)}, w_i^{(K)})\}_{i=1}^n$ .

---

**Example 3.2** (Example of Algorithm 2). Figure 2 shows an example of Algorithm 2, using  $K = 100$  iterations, implemented in MATLAB with  $n = 50$  grains in the square domain  $\Omega = [0, 1] \times [0, 1]$ . The grains have target areas  $m_i = x$  for  $i \in \{1, \dots, 35\}$  and  $m_i = 10x$  for  $i \in \{36, \dots, 50\}$ , where  $x = 1/185$  so that the total area of all the grains equals the area of  $\Omega$ . The actual areas of the grains returned by Algorithm 2 are correct to within 1% error ( $\varepsilon = 0.01$ ). For the initialisation step we used exactly the same points  $\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_n^{(0)}$  that were used for the initialisation step in Example 3.1. Observe from Figure 2 how the Laguerre diagram becomes more regular as the number of iterations  $k$  increases, and how it appears to be converging. The diagram already looks quite regular after just 4 or 5 iterations and the user may be happy to take far fewer than  $K = 100$  iterations. We discuss how to choose  $K$  in the following section.



**Figure 2:** An example of  $K = 100$  iterations of Algorithm 2 with  $n = 50$  grains in the unit square  $\Omega$ . There are 35 blue grains and 15 yellow grains. The blue grains have area  $x$  and the yellow grains have area  $10x$  to within 1% error, where  $x = 1/185$ . The black dots are the locations of the generators  $\{\mathbf{x}_i\}_{i=1}^{50}$ . In Figure (i) the generators  $\mathbf{x}_i$  are located at the centroids of their cells  $L_i$  to within a distance of 0.002.

### 3.1.1 Periodic Laguerre diagrams

Algorithms 1 and 2 can be modified to create *periodic* Laguerre diagrams for use as RVEs for computational homogenisation (RVEs are usually taken to be periodic to avoid artificial boundary effects). To create periodic diagrams in a rectangular box  $\Omega$  of side lengths  $l_1, l_2, l_3$ , modify Algorithms 1 and 2 as follows. Define the periodic distance between  $x, y \in \Omega$  by

$$|x - y|_{\text{per}} = \min\{|x - y + (il_1, jl_2, kl_3)| : i, j, k \in \mathbb{Z}\}. \quad (3.3)$$

In Algorithms 1 and 2 replace the Laguerre cells  $L_i$  by periodic Laguerre cells  $\tilde{L}_i$ , which are defined by

$$\tilde{L}_i = \{\mathbf{x} \in \Omega : |\mathbf{x} - \mathbf{x}_i|_{\text{per}}^2 - w_i \leq |\mathbf{x} - \mathbf{x}_j|_{\text{per}}^2 - w_j \forall j \in \{1, \dots, n\}\}. \quad (3.4)$$

In Algorithm 1 replace  $g$  by

$$g_{\text{per}}(w_1, \dots, w_n) = \sum_{i=1}^n (m_i - |\tilde{L}_i|)w_i + \sum_{i=1}^n \int_{\tilde{L}_i} |\mathbf{x} - \mathbf{x}_i|_{\text{per}}^2 d\mathbf{x}. \quad (3.5)$$

Replace  $g_k$  in Algorithm 2 in an analogous way.

## 3.2 Properties of the algorithms

### 3.2.1 Convergence of Algorithm 2: centroidal Laguerre diagrams

In Appendix A we prove that, under a generic assumption, Algorithm 2 converges as  $K \rightarrow \infty$ . This means that the generator locations  $\mathbf{x}_i^{(k)}$  settle down with increasing iterations, like we see in Figure 2. To be more precise, there exist  $\mathbf{x}_1, \dots, \mathbf{x}_n$  such that  $\lim_{k \rightarrow \infty} \mathbf{x}_i^{(k)} = \mathbf{x}_i$  for all  $i \in \{1, \dots, n\}$ . By taking the limit  $k \rightarrow \infty$  in equation (3.1), we see that

$$\mathbf{x}_i = \frac{1}{|L_i|} \int_{L_i} \mathbf{x} d\mathbf{x}. \quad (3.6)$$

Therefore the generator  $\mathbf{x}_i$  is the centroid of its own Laguerre cell  $L_i$  for all  $i$ . Such a Laguerre diagram is known as a *centroidal Laguerre diagram* or a *centroidal power diagram*, a term introduced in [34]; see also [35, 36]. Centroidal Laguerre diagrams tend to be more regular than non-centroidal Laguerre diagrams, as illustrated by Figure 2(i) (centroidal) and Figure 1 (non-centroidal).

### 3.2.2 Connection with Lloyd's algorithm

If we omit the optimisation step in Algorithm 2 and set the weights to be zero for all iterations,  $\mathbf{w}^{(k)} = \mathbf{0}$  for all  $k$ , then we obtain the well-known *Lloyd's algorithm* for computing *centroidal Voronoi tessellations* (Voronoi diagrams where each generator is the centroid of its own Voronoi cell) [37]. Therefore Algorithm 2 can be interpreted as a *generalised Lloyd algorithm with capacity constraints* where cell  $L_i$  is constrained to have volume  $m_i$ . An alternative method for generating centroidal Laguerre diagrams with capacity constraints is given in [36, Sec. 4].

### 3.2.3 Energy-decreasing property of Algorithm 2

Algorithm 2 can also be interpreted as an energy-decreasing optimisation method. Given  $m_1, \dots, m_n$  with  $\sum_i m_i = |\Omega|$ , define

$$E(\mathbf{x}_1, \dots, \mathbf{x}_n) = \min \left\{ \sum_{i=1}^n \int_{U_i} |\mathbf{x} - \mathbf{x}_i|^2 d\mathbf{x} : \{U_i\}_{i=1}^n \text{ is a partition of } |\Omega|, |U_i| = m_i \right\}. \quad (3.7)$$

Here the minimum is taken over all possible partitions of  $\Omega$ , not just Laguerre diagrams. This is an example of an *optimal transport problem*. For example, in two dimensions  $E$  could represent the minimum (squared) cost of transporting the recyclable waste generated by a population uniformly distributed over a country  $\Omega$  to recycling centres located at  $\{\mathbf{x}_i\}_{i=1}^n$  with capacities  $\{m_i\}_{i=1}^n$ . It can be shown [27, Sec. 6.4.1] that

$$E(\mathbf{x}_1, \dots, \mathbf{x}_n) = \sum_{i=1}^n \int_{L_i} |\mathbf{x} - \mathbf{x}_i|^2 d\mathbf{x} \quad (3.8)$$

where  $\{L_i\}_{i=1}^n$  is the Laguerre diagram with generators  $\{(\mathbf{x}_i, w_i)\}_{i=1}^n$ , where  $(w_1, \dots, w_n)$  is a maximum point of  $g$  (defined in (2.3)). In other words,  $\{L_i\}_{i=1}^n$  is the solution of the optimal transport problem and all the recyclable waste generated in region  $L_i$  should be sent to the recycling centre  $\mathbf{x}_i$ .

We could further ask what are the best locations of the recycling centres by considering the optimisation problem

$$\min_{\mathbf{x}_1, \dots, \mathbf{x}_n} E(\mathbf{x}_1, \dots, \mathbf{x}_n). \quad (3.9)$$

This is known as the *optimal location problem* in the economics literature [38] and the *quantization problem* in the discrete geometry [39], electrical engineering [40] and probability literature [41]. It can be shown that

$$\frac{\partial E}{\partial \mathbf{x}_i}(\mathbf{x}_1, \dots, \mathbf{x}_n) = 2 \int_{L_i} (\mathbf{x}_i - \mathbf{x}) \, d\mathbf{x}. \quad (3.10)$$

See for example [36]. Therefore  $\nabla E(\mathbf{x}_1, \dots, \mathbf{x}_n) = 0$  if and only if  $\{\mathbf{x}_i\}_{i=1}^n$  generate a centroidal Laguerre diagram.

Thanks to its regularisation step, Algorithm 2 is energy-decreasing in the sense that

$$E(\mathbf{x}_1^{(k+1)}, \dots, \mathbf{x}_n^{(k+1)}) \leq E(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)}). \quad (3.11)$$

Moreover, under the generic assumption (A.5), the sequence  $(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)})$  converges to a critical point of  $E$  (to a local minimum point or saddle point). In other words, it converges to a centroidal Laguerre diagram. We prove these statements in Appendix A. In general, Algorithm 2 does not converge to a *global* minimum point of  $E$  since  $E$  is highly non-convex with many critical points; Figure 4 illustrates four different (approximate) critical points of  $E$ , corresponding to different choices of  $(\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_n^{(0)})$ .

An alternative method for finding local minima of  $E$  is given in [36, Sec. 4] where, instead of updating  $x_i^{(k)}$  using our regularisation step, they update it using a quasi-Newton (L-BFGS) optimisation step applied to  $E$ .

## 4 Implementation

In this section we discuss different options for implementing Algorithms 1 and 2, which we did using MATLAB and Voro++ [42].

### 4.1 Computing Laguerre diagrams

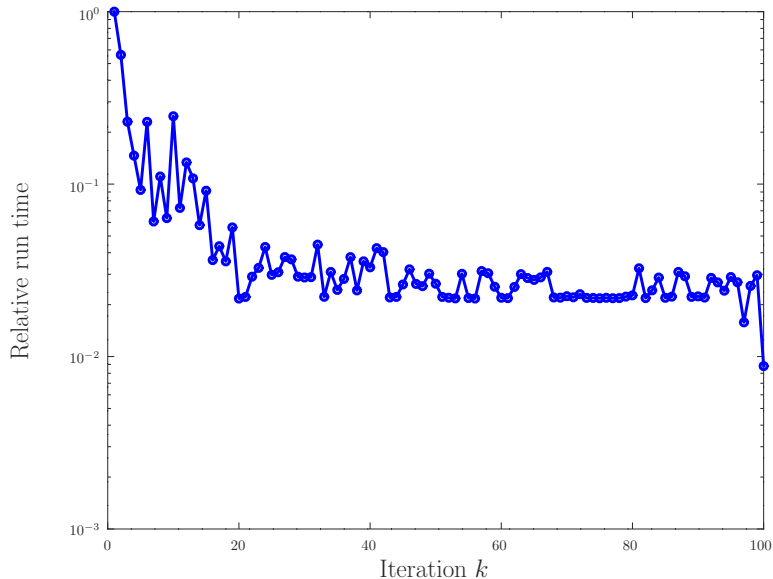
One of the main expenses of Algorithms 1 and 2 is the computation of Laguerre diagrams. This happens whenever the objective function  $g$  or  $g_k$  is evaluated, which could happen many times within a single optimisation step. A Laguerre diagram of  $n$  generators can be computed in  $\mathcal{O}(n \log n)$  flops in 2D and  $\mathcal{O}(n^2)$  flops in 3D [27, p. 85]. (Note that these are *worst-case* optimal run times and in practice the complexity may be better, as we observed in Example 5.1. For example, the complexity can be better if each cell has only  $\mathcal{O}(n)$  faces instead of the worst-case  $\mathcal{O}(n^2)$  [27, Theorem 6.1].) In applications  $n$  could be 10,000 or more, and hundreds or thousands of Laguerre diagrams could be computed in a single run of either algorithm. Therefore it is important to use efficient software.

For our 2D computations we used the MATLAB function `power_bounded` from the MATLAB File Exchange [43], which implements Aurenhammer’s lifting method [44] and crops the diagram to a rectangular box  $\Omega$ .

The function `power_bounded` is limited to 2D, and so for our 3D computations we used (a slightly modified version of) the C++ library Voro++ [42] combined with a MEX file so that we could run Voro++ via MATLAB. In 3D we also tried the MATLAB function `powerDiagramWrapper` from the MATLAB File Exchange [45], combined with our own code to crop the diagram to a cuboid  $\Omega$ , but we found Voro++ to be faster. Another advantage of Voro++ is that it can create periodic Laguerre diagrams.

We also used Tata Steel’s own in-house Laguerre diagram code to visualise Laguerre diagrams in 2D and 3D.





**Figure 3:** The relative run times for each optimisation step of Algorithm 2. In this example, the parameters are the same as those in Example 3.2, except that the number of grains is  $n = 500$ . The relative proportion of small and large grains is the same as in Example 3.2 and the ratio of the size of the largest grain to the size of the smallest grain is 10. The initial seed locations are randomly distributed in the unit square. The  $y$ -axis displays  $t_k/t_1$ , where  $t_k$  is the run time for the optimisation step of iteration  $k$ . After 12 iterations the time per iteration falls to below one-tenth the time of the first iteration, and after 18 iterations it falls to below one-twentieth. This is because the initial guess for the weights in the inner iteration is taken from the output of the previous outer iteration, and the quality of this guess improves as the number of iterations increases.

## 4.2 Optimisation methods

The other main expense of Algorithms 1 and 2 is the optimisation step. For each algorithm this is a smooth, unconstrained, concave maximisation problem and so is very tractable. We used the MATLAB function `fminunc` to minimise  $-g$  and  $-g_k$  (and hence maximise  $g$  and  $g_k$ ), which uses the BFGS quasi-Newton method by default [31]. This requires an initial guess  $\mathbf{w}_{\text{init}}$  for the minimum point.

### 4.2.1 Choice of the initial guess

For Algorithm 1 we recommend the initial guess  $\mathbf{w}_{\text{init}} = \mathbf{0}$ . For data fitting (like Example 5.3, where the seeds  $\mathbf{x}_i$  are taken from EBSD measurements), if the target grains are relatively spherical, then a better choice may be  $(\mathbf{w}_{\text{init}})_i = m_i/\pi$  in 2D or  $(\mathbf{w}_{\text{init}})_i = (3m_i/(4\pi))^{2/3}$  in 3D. In other words,  $(\mathbf{w}_{\text{init}})_i = r_i^2$  where  $r_i$  is the radius of a ball of area  $m_i$  in 2D or volume  $m_i$  in 3D. This is motivated by sphere-packing methods [4, 8, 10, 18].

For Algorithm 2 the initial guess should depend on the iteration  $k$ . For the first iteration  $k = 1$  we recommend  $\mathbf{w}_{\text{init}} = \mathbf{0}$ . For iterations  $k \geq 2$  we recommend  $\mathbf{w}_{\text{init}} = \mathbf{w}^{(k-1)}$ , the solution of the optimisation step from the previous iteration. As the number of iterations increases and the points  $\mathbf{x}_i^{(k)}$  converge, the initial guess  $\mathbf{w}_{\text{init}} = \mathbf{w}^{(k-1)}$  becomes better and better and consequently the optimisation step becomes quicker and quicker. This is illustrated in Figure 3, which shows the relative run time of each iteration for an example in which the relative sizes and relative proportions of small and large grains are the same as in Example 3.2 but the number of grains is  $n = 500$ . We see that the total runtime of the algorithm is not proportional to the number of iterations  $K$ ; most of the expense is in the first few iterations.

Note that for the first iteration,  $k = 1$ , the initial guess  $\mathbf{w}_{\text{init}} = \mathbf{0}$  does not incorporate any

information about the locations  $\mathbf{x}_i^{(0)}$ . It is possible to improve the speed of the first iteration by using a more sophisticated choice of  $\mathbf{w}_{\text{init}}$ , e.g., using the *multilevel methods* of Mérigot [30] and Lévy [32], which generate a good initial guess  $\mathbf{w}_{\text{init}}$  by solving a sequence of smaller optimisation problems with fewer grains. (For example, you can obtain a good initial guess  $\mathbf{w}_{\text{init}}$  for  $n$  grains by first solving a coarser problem with  $n/2$  grains; you can obtain a good initial guess for  $n/2$  grains by first solving a coarser problem with  $n/4$  grains, etc.) We found that Mérigot’s multilevel method [30] in 2D could halve the run time of iteration  $k = 1$  when there are  $n = 10,000$  grains. It is reported that Lévy’s multilevel program GEOGRAM can handle one million grains in 3D [32, Table 4].

It is also possible to obtain a better initial guess  $\mathbf{w}_{\text{init}}$  for iterations  $k \geq 2$  as follows. The Lloyd step (3.1) of Algorithm 2 could be replaced with a *damped Lloyd step* of the form

$$\mathbf{x}_i^{(k)} := (1 - \lambda)\mathbf{x}_i^{(k-1)} + \lambda \frac{1}{|L_i^{(k-1)}|} \int_{L_i^{(k-1)}} \mathbf{x} \, d\mathbf{x} \quad (4.1)$$

where  $\lambda$  is a damping parameter between 0 and 1. The choice  $\lambda = 1$  corresponds to the Lloyd step (3.1). The closer  $\lambda$  is to 0, the closer  $\mathbf{x}_i^{(k)}$  is to  $\mathbf{x}_i^{(k-1)}$ , and so the better the associated initial guess  $\mathbf{w}_{\text{init}} = \mathbf{w}^{(k-1)}$ . Therefore the optimisation step is faster for smaller  $\lambda$ . On the other hand, the regularisation step has less effect for smaller  $\lambda$ , and it is necessary to increase the number of iterations  $K$  to achieve the same amount of regularisation. For our purposes the full Lloyd step  $\lambda = 1$  was sufficiently fast and so we did not try to optimise the choice of  $\lambda$ .

#### 4.2.2 Choice of the tolerance

For simplicity we chose the tolerance  $\varepsilon$  of the optimisation step of Algorithm 2 to be fixed at each iteration  $k$  (recall that the optimisation step terminates when  $|\nabla g_k(\mathbf{w}^{(k)})| < \varepsilon \min_j m_j$ ). The algorithm could be sped up, however, by taking  $\varepsilon = \varepsilon_k$  to depend on  $k$ . In order for Algorithm 2 to produce a Laguerre diagram with grains of given volumes up to a relative error of  $\varepsilon$ , we only need the tolerance to be  $\varepsilon$  at the final iteration,  $\varepsilon_K = \varepsilon$ . For previous iterations we could use a cruder tolerance:  $\varepsilon = \varepsilon_K < \varepsilon_{K-1} < \dots < \varepsilon_2 < \varepsilon_1$ . It is tempting to think that the larger the tolerance, the faster the optimisation step. On the other hand, if  $\varepsilon_{k-1}$  is larger than  $\varepsilon_k$ , then the initial guess  $\mathbf{w}_{\text{init}} = \mathbf{w}^{(k-1)}$  at iteration  $k$  may be worse, and the optimisation step at iteration  $k$  may be slower. So the tolerances  $\varepsilon_k$  must be chosen carefully. The choice of fixed tolerance  $\varepsilon_k = \varepsilon$  for all  $k$  is a simple, reliable option, which is why we used it.

#### 4.2.3 Choice of the optimisation algorithm

The speed of the optimisation step depends of course not only on the choice of the initial guess  $\mathbf{w}_{\text{init}}$  and the tolerance  $\varepsilon$ , but also on the choice of the optimisation algorithm. For example, instead of using a quasi-Newton method like we did, one could use Newton’s method. Newton’s method tends to converge faster than quasi-Newton methods (quadratically rather than superlinearly), although it is harder to implement since it requires the second derivative of  $g$  (whereas quasi-Newton methods only require the first derivative) [31].

It can be shown (see, e.g., [35]) that

$$\frac{\partial^2 g}{\partial w_i \partial w_j} = -\frac{\partial |L_i|}{\partial w_j} = \begin{cases} -\sum_{k \in N_i} \frac{a_{ik}}{2|\mathbf{x}_i - \mathbf{x}_k|} & \text{if } i = j, \\ \frac{a_{ij}}{2|\mathbf{x}_i - \mathbf{x}_j|} & \text{if } j \in N_i, \\ 0 & \text{otherwise,} \end{cases} \quad (4.2)$$

where  $a_{ij}$  is the area of the face between cell  $i$  and cell  $j$  and  $N_i$  is the index set of the neighbours of cell  $i$  (that is  $j \in N_i$  if and only if cell  $j$  and cell  $i$  share a face). The pseudo-inverse of this

Hessian matrix is used in a damped Newton method in [33, Algorithm 1], where fractions of a full Newton step are used in order to control the error reduction and the minimum volume of a cell (to stop cells disappearing). The authors prove that their damped Newton method converges globally with order 1 and locally with order 2 [33, Theorem 1.5].

### 4.3 Initialisation: Effect on the spatial distribution

In this section we discuss the initialisation step of Algorithms 1 and 2.

#### 4.3.1 Initialisation of the seeds

The locations of the generators  $\mathbf{x}_1, \dots, \mathbf{x}_n$  at the termination of Algorithm 2 is a strong function of the initial choice  $\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_n^{(0)}$ . This simple observation gives us a great deal of control over the spatial distribution of the different sized grains. Examples of Algorithm 2 with different initial distributions of the generators  $\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_n^{(0)}$  are shown in Figure 4. In these examples  $\Omega = [0, 3] \times [0, 2]$  and there are  $n = 1000$  grains. There are  $n_1 = 800$  grains of size  $x$  and  $n_2 = 200$  grains of size  $20x$ . The tolerance is  $\varepsilon = 0.01$  and the number of iterations of Algorithm 2 is  $K = 20$ . The figure shows the output of Algorithm 2. The final spatial distribution of grains has some features in common with the spatial distribution of the initial generator locations.

A further example of controlling the spatial distribution of grains can be seen in Figure 5. In this example  $n = 1000$  grains have areas drawn from a random distribution such that the ratio of the largest to the smallest grain size is at most 100. The Laguerre diagram in Figure 5(a) has the property that the grain sizes tend to increase from left to right. A variety of spatial distributions of grain sizes can be simulated by first distributing the seed locations appropriately. Figure 5(b) shows how a more complicated distribution can be produced.

#### 4.3.2 Initialisation of the weights

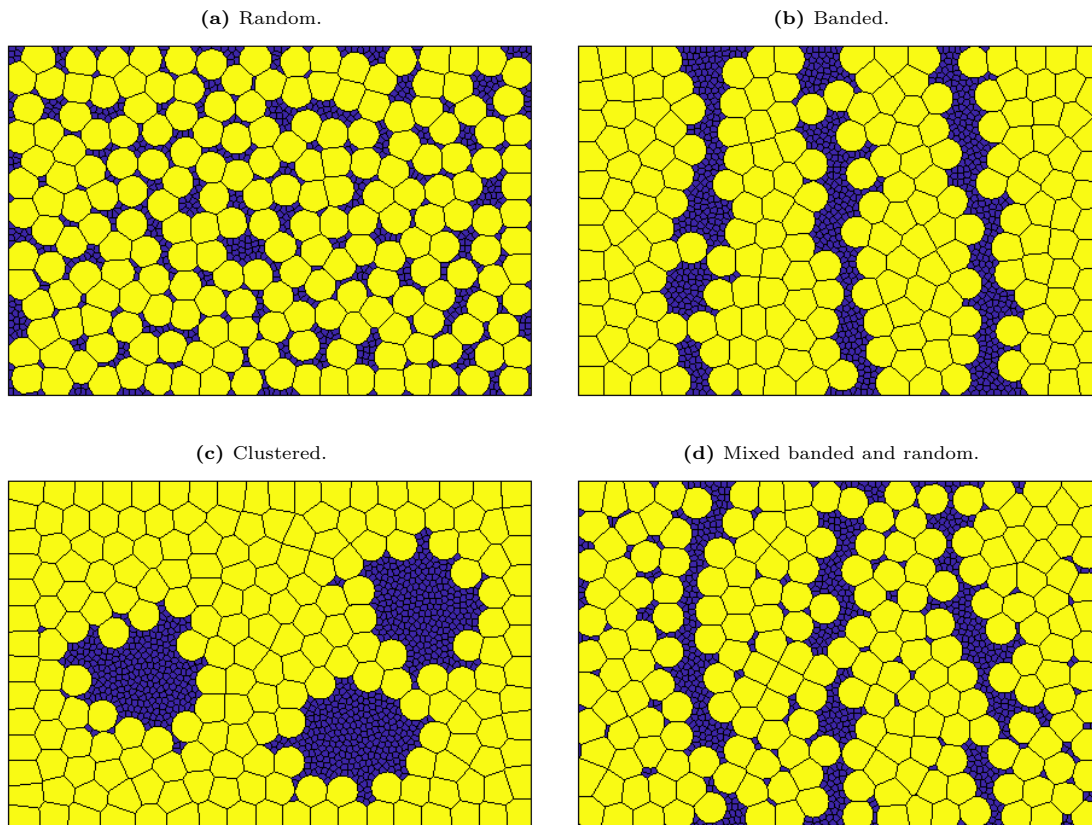
The choice of  $\mathbf{w}^{(0)}$  in the initialisation step of Algorithm 2 is also important. One should choose  $w_1^{(0)}, \dots, w_n^{(0)}$  so that the Laguerre diagram generated by  $(\mathbf{x}_1^{(0)}, w_1^{(0)}), \dots, (\mathbf{x}_n^{(0)}, w_n^{(0)})$  has no empty Laguerre cells. If there are empty cells then the regularisation step is not defined (there is division by zero in equation (3.1) if  $L_i^{(0)}$  is empty). A good choice is  $\mathbf{w}^{(0)} = \mathbf{0}$  since then the Laguerre diagram generated by  $(\mathbf{x}_1^{(0)}, w_1^{(0)}), \dots, (\mathbf{x}_n^{(0)}, w_n^{(0)})$  is a Voronoi diagram and so has no empty cells, whatever the choice of  $\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_n^{(0)}$ .

### 4.4 Stopping criteria

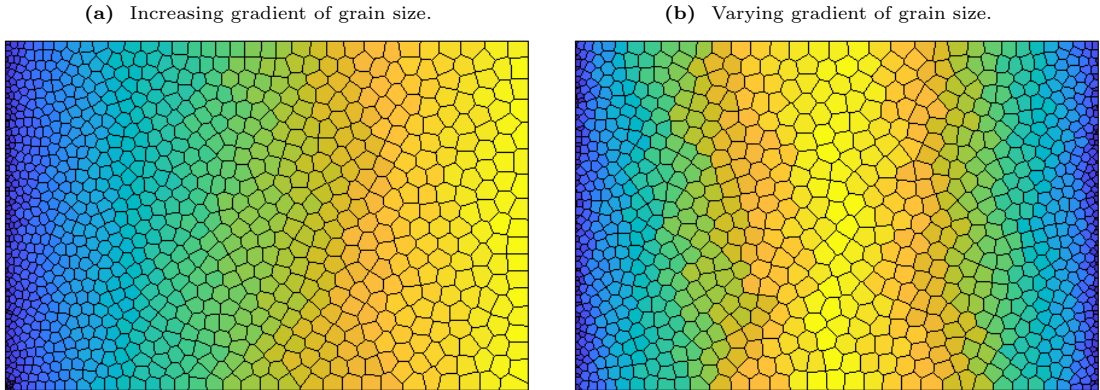
In Algorithm 2 the user must specify the number of regularisation steps  $K$ . As we discussed in Section 3.1, for large values of  $K$  the Laguerre diagram resulting from Algorithm 2 is approximately a *centroidal Laguerre diagram*, which means that each seed  $\mathbf{x}_i^{(K)}$  is approximately the centre of mass of its Laguerre cell  $L_i^{(K)}$ . Centroidal Laguerre diagrams tend to have very regular-shaped cells, e.g., in 2D if the grains all have the same target areas,  $m_i = 1/n$ , and if  $n$  and  $K$  are large, then the Laguerre diagram looks locally like a regular hexagonal tiling. Indeed for steel microstructures we found that if  $K$  is too large, then Algorithm 2 tends to produce grains that are too ‘round’ compared to EBSD measurements of grain aspect ratios.

Instead of fixing the number of steps  $K$  in advance, the user could terminate the algorithm whenever some measure of the maximum grain aspect ratio falls below a critical threshold. For example, the aspect ratio of a grain can be measured using the ratio of its largest and smallest principal moments of inertia, or using the ratio of the radii of circumscribed and inscribed balls, or using its sphericity [46], which is the ratio of the surface area of the volume-equivalent sphere to the surface area of the grain.

The user may also want to terminate the algorithm if the distance  $|\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)}|$  moved by the seeds from one iteration to the next falls below some threshold. The Laguerre diagram  $\{L_i^{(k)}\}$



**Figure 4:** Examples of more advanced microstructures: Coupling of size and spatial distributions with Algorithm 2. This figure shows the output of Algorithm 2 after  $K = 20$  iterations with different initial distributions for the seed locations. In all cases there are  $n = 1000$  grains with  $n_1 = 800$  grains of size  $x$  (in dark blue) and  $n_2 = 200$  grains of size  $20x$  (in yellow). a Random distribution of initial locations. Here the initial generator locations of the large and small grains are uniformly distributed over  $\Omega$ . b Banded distribution of initial locations. Here the different sized grains have initial generator locations that lie inside bands within  $\Omega$ . The sizes of the bands have been chosen so that there are approximately equal numbers of small grains within each blue band and approximately equal numbers of large grains within each yellow band. c Clustered distribution of initial locations. Here the smaller grains have initial generator locations that lie inside non-overlapping discs. d A mixed distribution: the initial generators are such that the larger grains are arranged in bands and the smaller grains are a combination of the banded and random distributions.



**Figure 5:** Experiments to demonstrate a gradient in the distribution of grain sizes. Here  $n = 1000$  grains have areas drawn from a uniform distribution such that the ratio of the largest to the smallest grain size is at most 100. The domain is  $\Omega = [0, 3] \times [0, 2]$ . The results are shown after  $K = 20$  iterations and the grains are coloured according to their area. a The initial seed locations  $\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_n^{(0)}$  are distributed such that the  $x$ -coordinate increases with grain size. b The initial seed locations are distributed such that the larger grains are found in the middle of  $\Omega$ .

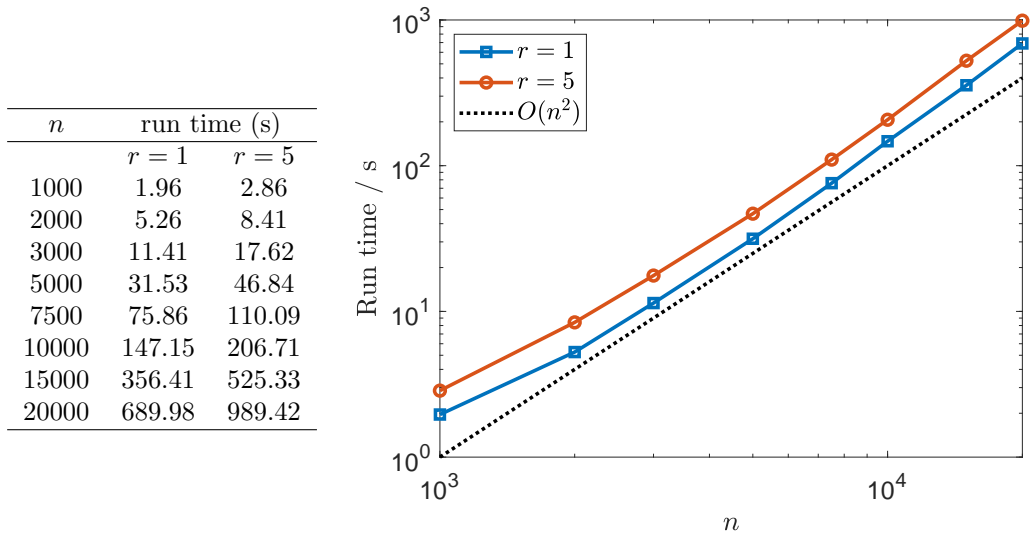
tends to evolve slowly with  $k$  when  $k$  is large, as illustrated in Figure 2, and the evolution can slow down dramatically when there is a T1 topological transition (to borrow a term from foam dynamics). This topological transition involves a change of cell neighbour relations; in 2D this is via coalescence of two triple junctions of cell boundaries. So in general there is little to be gained from performing a large number of regularisation steps, especially since our aim is not to produce a centroidal Laguerre diagram, but rather to produce a physically realistic microstructure. (If on the other hand the user’s aim is to produce centroidal Laguerre diagrams, then it would be better to use the quasi-Newton method of [36], which converges superlinearly as opposed to the linear convergence of Lloyd’s algorithm.)

## 5 Examples

This section includes some large examples in 3D to illustrate the capabilities of our algorithms.

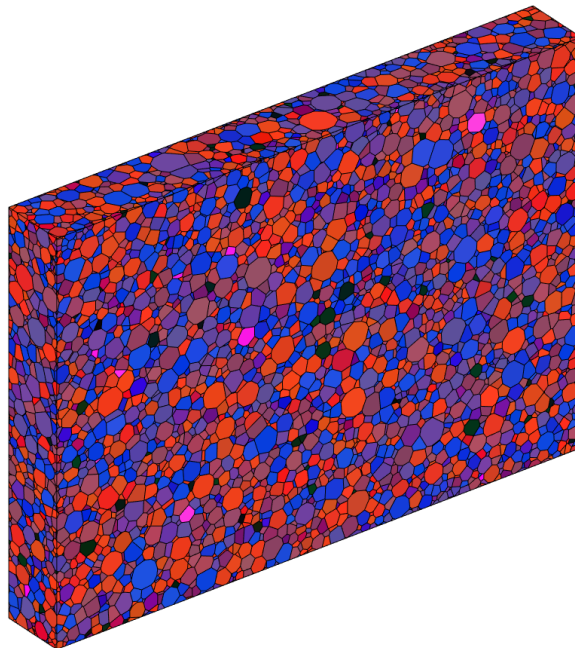
**Example 5.1** (Run time tests). Figure 6 gives some run times of Algorithm 2 for creating monodisperse and polydisperse periodic Laguerre diagrams in 3D. Here Algorithm 2 has been used to create  $n/2$  grains of volume  $x$  and  $n/2$  grains of volume  $rx$ , for  $r = 1$  and  $r = 5$ , in a cube of side length 100 with error tolerance  $\varepsilon = 0.01$ . We see that the run time grows roughly quadratically in the range  $n = 2000$  to  $n = 5000$ , for both the monodisperse (single phase) case  $r = 1$  and the polydisperse (dual phase) case  $r = 5$ . This could be expected since the cost of each iteration of the BFGS method used by *fminunc* is  $\mathcal{O}(n^2)$  (a matrix-vector multiplication). Also the worst-case optimal time it takes to compute a Laguerre diagram of  $n$  cells is  $\mathcal{O}(n^2)$  in 3D, although we found that in these examples the cost of computing the Laguerre diagrams was sub-quadratic (but super-linear); see also the discussion in Section 4.1. For  $n > 5000$  the run time grows a little faster than  $\mathcal{O}(n^2)$ . Observe also from Figure 6 that it is about 50% more expensive to compute dual phase RVEs ( $r = 5$ ) than single phase RVEs ( $r = 1$ ).

**Example 5.2** (Generating a periodic RVE of an IF steel). Figure 7 shows an example of a periodic Laguerre diagram created using Algorithm 2. The target volumes  $m_i$  are taken from a 3D EBSD measurement of an IF (interstitial free) steel (EN 10130 grade DC06). There are  $n = 9211$  grains in a box of dimensions  $670\mu\text{m} \times 80\mu\text{m} \times 480\mu\text{m}$ . We took the initial seed locations  $\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_n^{(0)}$  to be the centres of mass of the grains from the EBSD data, and performed  $K = 10$  regularisation steps with a tolerance of 0.5% ( $\varepsilon = 0.005$ ). The grains in Figure 7 are coloured according to their lattice orientations by mapping the three Euler angles linearly to RGB values. The orientations

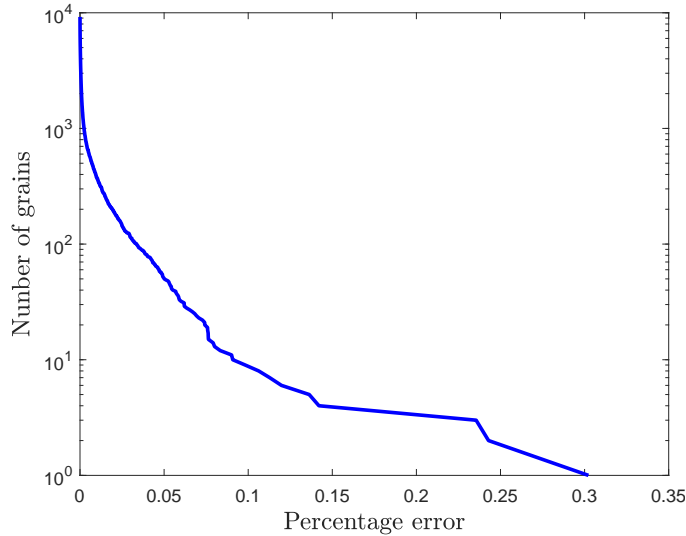


**Figure 6:** Run times in seconds of Algorithm 2 for creating monodisperse (single phase) and polydisperse (dual phase) periodic RVEs in 3D. There are  $n/2$  grains of volume  $x$  and  $n/2$  grains of volume  $rx$  in a cube of side length 100 with at most 1% error ( $\varepsilon = 0.01$ ) in the volumes of the grains ( $r = 1$  corresponds to a single phase material,  $r = 5$  corresponds to a dual phase material,  $x$  is chosen such that the total volume of the grains equals the volume of the box). We used  $K = 5$  regularisation steps, and the initial seed locations  $\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_n^{(0)}$  were chosen randomly from a uniform distribution. The simulations were performed on an Intel Xeon E5-1620V3 (3.5GHz, 4 cores, 8 threads). The graph on the right has a log-log scale. The black dotted line is the graph of the function  $cn^2$ , where  $c$  is a constant. It is included to illustrate how the run times grow with  $n$ .

were taken from the EBSD data. Figure 8 shows that the volumes of all the grains are correct to within 0.5%, and most volumes are correct to within 0.05%.



**Figure 7:** A periodic RVE of an IF steel with  $n = 9211$  grains, generated using Algorithm 2 (see Example 5.2). The grains are coloured according to their lattice orientations.



**Figure 8:** The complementary cumulative number distribution of the percentage error of the volumes of the grains for Example 5.2. For a percentage error  $x$  we plot the number of grains that have volume percentage error at least  $x$ . Most of the grain volumes have percentage error less than 0.05%, which is an order of magnitude below the tolerance of 0.5% ( $\varepsilon = 0.005$ ). In this example, the maximum percentage error is 0.30%.

**Example 5.3** (Fitting a Laguerre diagram to EBSD measurements). The main aim of this paper is to create Laguerre diagrams with given volume distributions for use in computational homogenisation simulations. We briefly mention, however, how Algorithm 1 can be used to fit a Laguerre diagram to EBSD data of grain volumes and centroids. Figure 9 is an example of a non-periodic Laguerre diagram fitted to a 3D EBSD measurement of an IF steel (EN 10130 grade DC06) with  $n = 9211$  grains in a box of dimensions  $670\mu m \times 80\mu m \times 480\mu m$  (this is the same EBSD data used in the previous example). In the initialisation step of Algorithm 1 we took  $\mathbf{x}_1, \dots, \mathbf{x}_n$  to be the centroids of the grains from the EBSD data. The target volumes  $m_i$  were also taken from the EBSD data. We used a tolerance of 1% ( $\varepsilon = 0.01$ ). As in the previous example, the grains in Figure 9 are coloured according to their lattice orientation. Observe that Figure 9 is less regular than Figure 7, which is because Algorithm 1 is missing the regularisation steps of Algorithm 2.

Figure 10 shows that the volumes of all the grains are correct to within 0.56%, and most volumes are correct to within 0.1%.

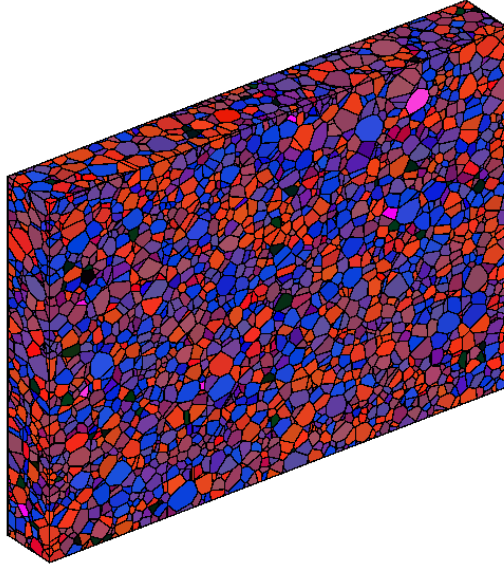
Figure 11 shows the complementary cumulative number distribution of the relative errors of the centroids. The relative error for grain  $i$  is defined by

$$\frac{|\mathbf{x}_i - \mathbf{c}_i|}{r_i} \quad (5.1)$$

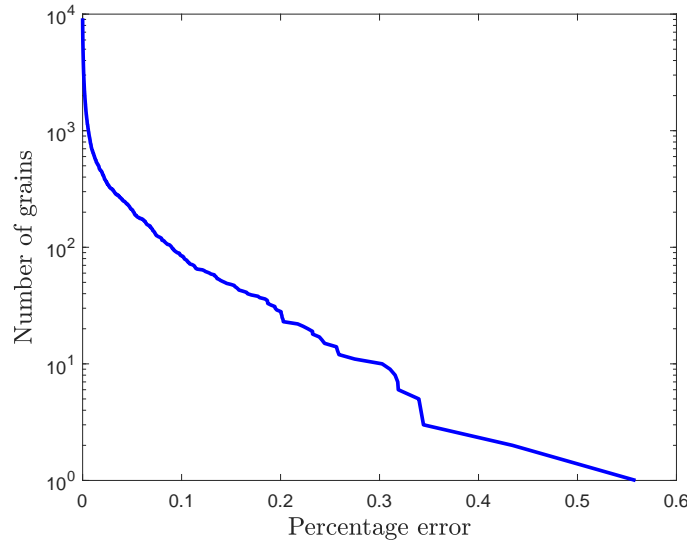
where  $\mathbf{x}_i$  is the centroid of grain  $i$  from the EBSD data,  $\mathbf{c}_i$  is the centroid of the Laguerre cell  $L_i$ , and  $r_i$  is the radius of a sphere of volume  $m_i$ , where  $m_i$  is the target volume of grain  $i$ . This definition of relative error was proposed by [9]. As expected, the centroid errors are higher than the volume errors since Algorithm 1 does not directly try to fit the centroids. To be precise, the optimisation step of Algorithm 1 only minimises the fitting error of the volumes; the centroid fitting error is not minimised (the centroids do not appear in the objective function  $g$ ). However, the centroid measurements are used in the initialisation step of Algorithm 1. The relative error of 79% of the grain centroids is less than 1 and the relative error of 93% of the grain centroids is less than 2.

The run time for this example was 376 seconds on an Intel Xeon E5-1620V3 (3.5GHz, 4 cores, 8 threads) with the initial guess  $(\mathbf{w}_{\text{init}})_i = (3m_i/(4\pi))^{2/3}$ , which was inspired by sphere-packing methods [4, 8, 10, 18].



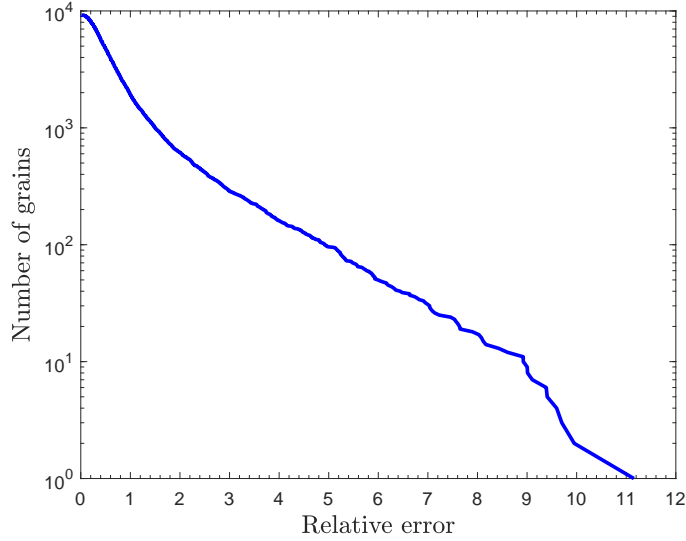


**Figure 9:** A non-periodic Laguerre diagram fitted to 3D EBSD data of an IF steel using Algorithm 1 (see Example 5.3). The grains are coloured according to their lattice orientations. The volume distribution has a fitting error of less than 1%. The texture intensity inherits the same fitting error.



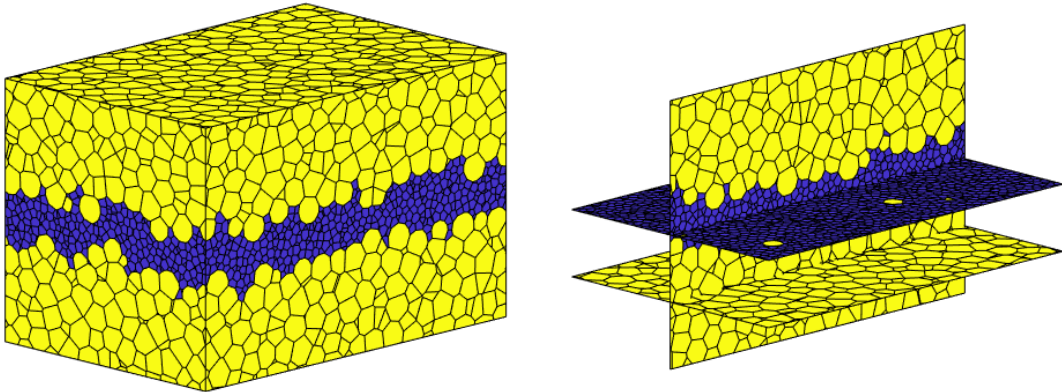
**Figure 10:** The complementary cumulative number distribution of the percentage error of the volumes of the grains for Example 5.3. For a percentage error  $x$  we plot the number of grains with volume percentage error at least  $x$ . The largest percentage error is 0.56% and the second largest is 0.43%. All the other percentage errors are below 0.34%.





**Figure 11:** The complementary cumulative number distribution of the relative error of the centroids of the grains for Example 5.3. For a relative error  $x$  we plot the number of grains with centroid relative error at least  $x$ . The maximum relative error is 11.16%, which is worse than the result given in [9, Fig. 9], although for most of the grains the relative errors are comparable: 7278 of the 9211 grains have relative error less than 1, and 8596 of the 9211 grains have relative error less than 2.

**Example 5.4** (Generating a dual phase RVE with a banded microstructure). Figure 12 shows an example of a periodic, dual phase Laguerre diagram with a band of small grains in the centre, generated using Algorithm 2. There are  $n = 10,000$  grains: 8000 grains of volume  $x$  and 2000 grains of volume  $20x$  (where  $x$  was chosen so that the total volume of the grains equals the volume of the box). We used  $K = 20$  regularisation steps and a volume tolerance of 1%. The grains are coloured according to their volume. In order to obtain the banded structure, we placed the initial seeds  $\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_n^{(0)}$  at random within bands of the correct volume. We see from Figure 12 that these bands were largely preserved by the regularisation steps.



**Figure 12:** A periodic RVE of a dual phase material with a banded microstructure (see Example 5.4).

**Example 5.5** (Generating an RVE with a log-normal distribution of grain volumes). Figure 13 shows an example of a periodic Laguerre diagram generated by Algorithm 2, in which the grains have volumes that are log-normally distributed. There are  $n = 10,000$  grains. We used  $K = 5$

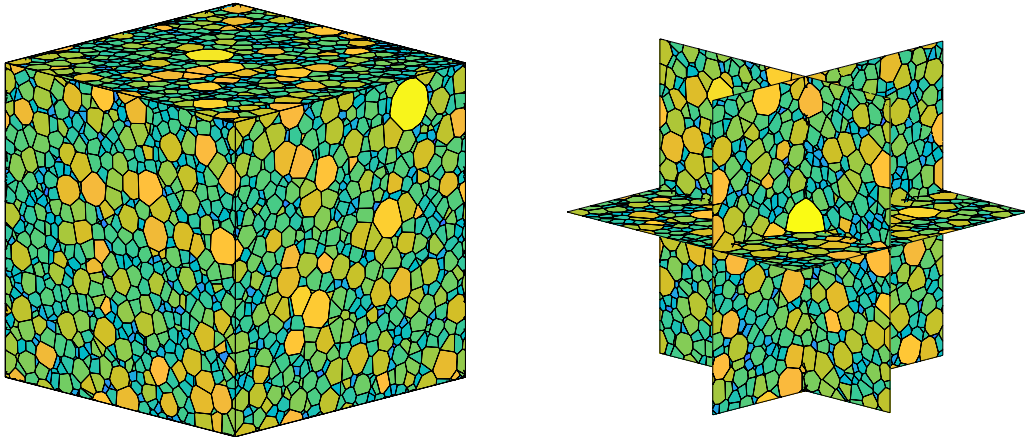
regularisation steps and a volume tolerance of 1%. The grains are coloured by their volume, using a log scale. We placed the initial seeds  $\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_n^{(0)}$  at random. The target volumes were generated by drawing 10,000 samples  $r_i$  from a log-normal distribution with mean 1 and standard deviation 0.35 (these correspond to the log-normal parameters  $\sigma = (\log(1 + 0.35^2))^{1/2} = 0.34$  and  $\mu = -\sigma^2/2$ ). The target volumes were defined by

$$m_i = \frac{r_i^3}{L_1 L_2 L_3 \sum_{j=1}^n r_j^3}, \quad (5.2)$$

where the  $L_i$  are the side-lengths of the domain  $\Omega = [0, L_1] \times [0, L_2] \times [0, L_3]$ . For large  $n$  these target volumes are approximately log-normally distributed with coefficient of variation (the ratio of standard variation to mean) of

$$\frac{1}{3} \sqrt{\exp(9\sigma^2) - 1}. \quad (5.3)$$

The algorithm took 669 seconds using the same machine as above. Observe from Example 5.1 that Algorithm 2 took only 147 seconds to produce a monodisperse RVE and 207 seconds to produce a dual phase RVE with the same number of grains ( $n = 10,000$ ) and the same number of regularisation steps ( $K = 5$ ). Therefore the run time of Algorithm 2 increases as the RVE becomes more polydisperse.



**Figure 13:** An RVE in which the grain volumes have approximately log-normal distribution (see Example 5.5). There are  $n = 10,000$  grains in the cubic cell. The coefficient of variation of the volumes (the ratio of the standard deviation to the mean) is 1.4. Cross-sections of the cell are also shown, showing the distribution of sizes throughout the cell.

## 6 Discussion

The advantages of our method are

- it is fast;
- it can create Laguerre diagrams with grains of *exact* volumes, in principle of any desired tolerance up to machine precision;
- it gives some control over the spatial distribution of the grains;
- it can create periodic and non-periodic Laguerre diagrams.

The disadvantages of our method are

- it provides no direct control over the centroids of the grains or their morphology, e.g., their aspect ratio;
- it is currently limited to Laguerre diagrams and so the grains cannot have curved boundaries or be non-convex.

We now discuss these advantages and disadvantages in more detail, give evidence in support of our claims, and compare our method with others in the literature.

## 6.1 Controlling grain volumes: Speed and accuracy

Figure 6 shows that we can create Laguerre diagrams in 3D with up to 20,000 grains in around 10 minutes on a standard desktop PC (without using parallel computation), where the volumes of the grains are correct to within 1%. For 10,000 grains we require as little as 2.5 minutes (see Figure 6), although the time depends on the regularity of the microstructure and whether the material is monodisperse or polydisperse; in Example 5.3 it took 6.25 minutes for 9211 grains and in Example 5.5 it took about 11 minutes for 10,000 grains.

In our implementation of Algorithm 2 we simply used MATLAB’s all-purpose *fminunc* optimisation algorithm. With modern, customised optimal transport optimisation algorithms such as [33, 32] it should be possible to use our method to generate Laguerre diagrams with given volume distributions with 100,000 grains in a few minutes [32, Table 3] or even one million grains in less than an hour [32, Table 4].

We now compare this with the speed of other algorithms. It is difficult to make a direct comparison in some cases since different methods fit different geometric properties.

In [9] a stochastic optimisation method (the cross-entropy method) is used to solve a non-convex optimisation problem to fit a Laguerre diagram to 3D XRD measurements of grain volumes *and centroids*. The authors report simulation times (performed using parallel computation) of 19.2 hours for 1439 grains and 122.3 hours for 8063 grains. Note that it is hard to compare their run times with ours since they are also fitting centroids; their method does not try to fit the volumes exactly like we do, but rather obtain a good fit for both volumes and centroids, and their method can produce empty Laguerre cells (grains with volume zero). While the main focus of our paper is to fit volumes only, we gave an example of fitting volumes and centroids in Example 5.3, where we fit a Laguerre diagram to 3D EBSD measurements of an IF steel with 9211 grains. The run time is 6.25 minutes and the volumes are correct to within 0.56%. The centroid errors of most of the grains are comparable to those given in [9, Fig. 9], although overall our method does worse than [9] at centroid fitting, as expected.

Sphere-packing methods are popular for fitting Laguerre diagrams to volume distributions [4, 8, 10, 18]. For  $n$  non-overlapping spheres  $S_1, \dots, S_n$  with centres  $\mathbf{x}_1, \dots, \mathbf{x}_n$  and radii  $r_1, \dots, r_n$ , the Laguerre diagram with seeds  $\mathbf{x}_i$  and weights  $w_i = r_i^2$  has the property that cell  $L_i$  contains sphere  $S_i$ . Therefore the volume of  $L_i$  is at least the volume of  $S_i$ . The idea of sphere-packing methods is that if the spheres are close-packed, then the volumes of the Laguerre cells are approximately equal to the volumes of the solid spheres. The disadvantage of this method is that it is inexact and computationally expensive since the sphere-packing problem is NP hard [47]. Nevertheless, this method provided us with inspiration for a good initial guess for the optimisation simulation in Example 5.3 (see also Section 4.2.1).

In [1] a method is proposed for fitting grain measurements with *generalised balanced power diagrams* (GBPDs), which are a generalization of Laguerre diagrams. GBPDs are generated by triples  $(\mathbf{x}_i, w_i, A_i)$  of seeds  $\mathbf{x}_i$ , weights  $w_i$ , and positive definite matrices  $A_i$ ; the matrices  $A_i$  give some control over the aspect ratio of the generalised Laguerre cells; the case  $A_i = I$  for all  $i$  corresponds to a standard Laguerre diagram. The advantage of GBPDs is that they give a high degree of control over the morphology of the grains [1, Figs. 1-6]. The disadvantage is that they are hard to compute. In [1] the authors approximate GBPDs by voxels, and they fit discretised GBPDs to grain measurements by solving a high-dimensional linear programming problem, where

the number of unknowns equals the number of grains multiplied by the number of voxels. It is reported that to fit a discretised GBPD to 109 grains in 3D took around 6 hours on a standard laptop (this involved solving a linear programming problem with over 77 million variables and 78 million constraints) [1, Sec. 5.3]. Again, it is not possible to make a direct comparison of these run times with the ones presented here since grain volumes and morphology are fitted in [1], not only grain volumes like here.

A heuristic method for approximately fitting GBPDs to measurements of grain volumes, centroids and aspect ratios was proposed in [13]. Their method entirely avoids solving an optimisation problem; it includes explicit formulas for the generators  $(\mathbf{x}_i, w_i, A_i)$  in terms of the data. It is reported in [13] that the method is comparable in accuracy with the optimisation methods of [1, 11, 21] but takes a small fraction of the computation time. No run times or volume errors are reported in [13] precluding a more precise comparison with our method. Like the sphere-packing method, this heuristic method could be used for generating good initial guesses for optimisation methods.

## 6.2 Controlling grain geometry

The main goal of our method is to fit grain volumes quickly and accurately. Unlike other methods [1, 9, 21, 24, 12, 11, 13] it is not specifically designed for fitting grain morphology. We now discuss to what extent we can control the geometry of Laguerre diagrams.

Our method gives some control over the spatial distribution of the grains, as shown in Figures 4, 5 and 12, where we create microstructures with bands, clusters, and size gradients.

Several authors use grain centroids as a measure of fitting-error when fitting Laguerre diagrams to data measurements, e.g., [9, 13]. We show how we can approximately fit grain centroids to 3D EBSD data in Example 5.3, although the accuracy is much lower than the volume accuracy.

In its current form, our method gives no direct control over the aspect ratio of the grains. Like the sphere-packing method, Algorithm 2 tends to produce grains that are too round compared to grains typically seen in metals; see Section 4.4.

Nevertheless there are several ways how our method could be generalised to give more control over the morphology of the grains. For example, by combining our method with *multilevel Voronoi diagrams* [5, 17] we could maintain control over the volume of the grains while producing more realistic RVEs with non-convex and elongated grains. The idea would be to first use Algorithm 2 to create a Laguerre diagram with  $N$  ‘micro-grains’ of equal volume for large  $N$ . Then we would glue together the micro-grains into non-convex ‘macro-grains’. By choosing which micro-grains to glue, we would control the volume and the morphology of the macro-grains. (The multilevel Voronoi method glues together two micro-grains if their generators lie in the same Voronoi cell of a ‘coarser’ Voronoi diagram with fewer generators.)

In principle our algorithms can also be generalised very easily to produce GBPDs with grains of given volumes (up to any desired tolerance) by modifying the objective functions  $g$  and  $g_k$  in Algorithms 1 and 2 (simply replace the Laguerre cells  $L_i$  with generalised Laguerre cells, and replace the isotropic distances  $|\mathbf{x} - \mathbf{x}_i|$  with anisotropic distances  $|\mathbf{x} - \mathbf{x}_i|_{A_i}$ ). This would again allow us to control both the volumes and the aspect ratio of the grains. In practice, however, it is expensive to compute GBPDs to high accuracy; discretizing them with voxels greatly increases the cost of the algorithm. Without developing new computational geometry algorithms for the efficient computation of GBPDs, this limits the method to a small number of grains or greatly increases the run time (cf. the run time of 6 hours for 109 grains in 3D in [1]).

Since our method is currently limited to Laguerre diagrams, the grains cannot have curved boundaries or be non-convex. Curved grain boundaries can be created using additively-weighted Voronoi diagrams (Apollonius diagrams) [27], anisotropic diagrams [2], or GBPDs [1, 21, 12, 13], although these are all more costly to compute than Laguerre diagrams. Algorithms 1 and 2 can also be modified to produce Apollonius diagrams with grains of given volumes (in the definition of the objective functions  $g$  and  $g_k$  simply replace the Laguerre cells  $L_i$  with Apollonius cells, and replace the squared distances  $|\mathbf{x} - \mathbf{x}_i|^2$  with non-squared distances  $|\mathbf{x} - \mathbf{x}_i|$ ) but again the

implementation cost is an obstacle at the present time. We plan to explore this and the above generalizations in a future paper.

## 7 Conclusions

In this paper we introduced a fast optimisation method for generating Laguerre diagrams with user-defined grain size distributions. The volumes of the grains can be controlled exactly (to within any given tolerance). We produced industrially-relevant examples of RVEs with up to 20,000 grains with only 1% volume error in the order of minutes on a standard desktop PC. We also demonstrated how the spatial and texture distribution of the grains can be partially controlled. Our algorithms can create both non-periodic Laguerre diagrams (for data fitting) and periodic Laguerre diagrams (for generating RVEs of polycrystalline metals or solid foams for computational homogenisation).

## Acknowledgments

The authors would like to thank Carola Celada-Casero for useful discussions. DPB would like to thank the EPSRC for financial support via the grant EP/R013527/1, EP/R013527/2 Designer Microstructure via Optimal Transport Theory. Some of the work of DPB was carried out at Durham University. The work on generating 3D EBSD data has received funding from the European Union's Horizon 2020 research and innovation programme *Euratom research and training programme 2014-2018* under grant agreement No 709418 MuSTMeF.

## A Proof that Algorithm 2 is energy-decreasing and convergent

Throughout this section we assume that  $\Omega \subset \mathbb{R}^d$  is compact.

First we prove that Algorithm 2 is energy-decreasing, equation (3.11). Recall that if  $U$  is a compact subset of  $\mathbb{R}^d$  with centroid  $\mathbf{c}(U)$ , then

$$\min_{\mathbf{z} \in \mathbb{R}^d} \int_U |\mathbf{x} - \mathbf{z}|^2 d\mathbf{x} = \int_U |\mathbf{x} - \mathbf{c}(U)|^2 d\mathbf{x}. \quad (\text{A.1})$$

This follows from the fact that the function  $\mathbf{z} \mapsto \int_U |\mathbf{x} - \mathbf{z}|^2 d\mathbf{x}$  is strictly convex with unique critical point  $\mathbf{c}(U)$ . By equation (3.8) and by the way  $\{(\mathbf{x}_i^{(k)}, w_i^{(k)})\}_{i=1}^n$  is constructed using Algorithm 2,

$$E(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)}) = \sum_{i=1}^n \int_{L_i^{(k)}} |\mathbf{x} - \mathbf{x}_i^{(k)}|^2 d\mathbf{x} \quad (\text{A.2})$$

where  $\{L_i^{(k)}\}_{i=1}^n$  is the Laguerre diagram with generators  $\{(\mathbf{x}_i^{(k)}, w_i^{(k)})\}_{i=1}^n$ . Therefore

$$\begin{aligned} E(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)}) &\stackrel{(\text{A.1})}{\geq} \sum_{i=1}^n \int_{L_i^{(k)}} |\mathbf{x} - \mathbf{c}(L_i^{(k)})|^2 d\mathbf{x} \\ &= \sum_{i=1}^n \int_{L_i^{(k)}} |\mathbf{x} - \mathbf{x}_i^{(k+1)}|^2 d\mathbf{x} \quad (\text{regularisation step of Alg. 2}) \\ &\stackrel{(3.7)}{\geq} E(\mathbf{x}_1^{(k+1)}, \dots, \mathbf{x}_n^{(k+1)}). \end{aligned} \quad (\text{A.3})$$

This proves (3.11). The inequalities above are strict unless  $\mathbf{x}_i^{(k)} = \mathbf{x}_i^{(k+1)}$  for all  $i$ , which means that  $(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)})$  is a fixed point of Algorithm 2.

Next we prove a *weak global convergence* result of the form [48, Theorem 3.8], where convergence of the classical Lloyd algorithm was proved. Weak global convergence means that  $\nabla E(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)}) \rightarrow \mathbf{0}$  as  $k \rightarrow \infty$  and that any convergent subsequence of  $(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)})$  converges to a critical point of  $E$ , namely to a centroidal Laguerre diagram. This convergence is called *weak* because it does not guarantee convergence of the whole sequence  $(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)})$  (different subsequences could converge to different critical points).

By construction  $\mathbf{x}_i^{(k)}$  is the centroid of the convex set  $L_i^{(k-1)}$ , which has volume  $m_i$ . Therefore by [48, Lemma 3.2] the distance between  $\mathbf{x}_i^{(k)}$  and  $\partial L_i^{(k-1)}$  has a lower bound of  $Cm_i^2/\text{diam}(\Omega)^{2d-1}$  where  $C = 1/1024$ . Therefore the closest that two generators  $\mathbf{x}_i^{(k)}$  and  $\mathbf{x}_j^{(k)}$  can be is  $2Cm^2/\text{diam}(\Omega)^{2d-1}$  where  $m = \min_i m_i$ . Note that this bound is independent of the iteration number  $k$ . Therefore the iterates  $(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)})$  lie in the compact set

$$\{(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \Omega^n : |\mathbf{x}_i - \mathbf{x}_j| \geq 2Cm^2/\text{diam}(\Omega)^{2d-1} \forall i, j\}. \quad (\text{A.4})$$

Owing to this compactness and the energy-decreasing property of Algorithm 2, we have weak global convergence of  $(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)})$ ; see the Global Convergence Theorem in [49, p. 206] or [35, proof of Theorem 3.3].

Finally we prove a *strong convergence* result, namely that the whole sequence  $(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)})$  produced by Algorithm 2 converges to a critical point of  $E$ . We are only able to prove this, however, under the following generic assumption: There are only finitely many centroidal Laguerre diagrams with the same energy  $E$ . More precisely we assume that, for all  $M > 0$ ,

$$\#\{(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \Omega^n : \nabla E(\mathbf{x}_1, \dots, \mathbf{x}_n) = \mathbf{0}, E(\mathbf{x}_1, \dots, \mathbf{x}_n) = M\} < \infty. \quad (\text{A.5})$$

This assumption is expected to hold for ‘generic’ domains  $\Omega$  and masses  $m_1, \dots, m_n$  [50, p. 107], [35, Remark 3.4], however there are examples where it fails. For example, if  $\Omega$  is a disc,  $n = 2$  and  $m_1 = m_2$ , then there are infinitely many critical points of  $E$  with the same energy by the rotational symmetry of  $\Omega$ . Note that if  $E$  satisfies the generic condition of being a Morse function (having no degenerate critical points), then its critical points are isolated. Since they lie in the compact set (A.4) there are only finitely many of them, and so assumption (A.5) is satisfied.

The Monotone Convergence Theorem implies that the whole sequence  $E(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)})$  converges (because Algorithm 2 is energy-decreasing and  $E$  is bounded below by zero). In addition  $E$  is continuous, and so every accumulation point of  $(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)})$  has the same energy  $E$ . Moreover, by the global weak convergence result above, every accumulation point is a critical point of  $E$ . Therefore assumption (A.5) ensures there are only finitely many accumulation points.

We now complete the proof following the idea from [50, proof of Theorem 2.5]. Assume for contradiction that the sequence  $\mathbf{X}^{(k)} := (\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)})$  does not converge. Since it only has finitely many accumulation points, there exist distinct accumulation points  $\mathbf{Y} := (\mathbf{y}_1, \dots, \mathbf{y}_n)$ ,  $\mathbf{Z} := (\mathbf{z}_1, \dots, \mathbf{z}_n)$  and distinct subsequences  $\mathbf{X}^{(k_j)} := (\mathbf{x}_1^{(k_j)}, \dots, \mathbf{x}_n^{(k_j)})$ ,  $\mathbf{X}^{(k_l)} := (\mathbf{x}_1^{(k_l)}, \dots, \mathbf{x}_n^{(k_l)})$  such that  $\mathbf{X}^{(k_j)} \rightarrow \mathbf{Y}$ ,  $\mathbf{X}^{(k_l)} \rightarrow \mathbf{Z}$ , and  $k_l = k_j + 1$ , i.e., the sequence  $\mathbf{X}^{(k)}$  ‘jumps’ between the two subsequences infinitely many times. (Note that such subsequences may not exist if there are infinitely many accumulation points.) Let  $\delta = |\mathbf{Y} - \mathbf{Z}| > 0$  and let  $T : \Omega^n \rightarrow \Omega^n$  denote the continuous map that sends  $(\mathbf{x}_1, \dots, \mathbf{x}_n)$  to  $(\mathbf{c}(L_1), \dots, \mathbf{c}(L_n))$ , where  $\{L_i\}$  is the Laguerre diagram with seeds  $(\mathbf{x}_1, \dots, \mathbf{x}_n)$  and cells of volume  $m_1, \dots, m_n$ . In other words,  $\mathbf{X}^{(k+1)} = T(\mathbf{X}^{(k)})$ . Moreover,  $T(\mathbf{Y}) = \mathbf{Y}$  and  $T(\mathbf{Z}) = \mathbf{Z}$ . Then

$$\begin{aligned} \delta &= |\mathbf{Y} - \mathbf{Z}| \\ &\leq |\mathbf{Y} - \mathbf{X}^{(k_j)}| + |\mathbf{X}^{(k_j)} - \mathbf{X}^{(k_l)}| + |\mathbf{X}^{(k_l)} - \mathbf{Z}| \\ &= |\mathbf{Y} - \mathbf{X}^{(k_j)}| + |\mathbf{X}^{(k_j)} - T(\mathbf{X}^{(k_j)})| + |\mathbf{X}^{(k_l)} - \mathbf{Z}|. \end{aligned} \quad (\text{A.6})$$

By taking  $j, l \rightarrow \infty$  in the right-hand side, and using the continuity of  $T$ , we find that  $\delta = 0$ , which is a contradiction. Therefore the whole sequence  $(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_n^{(k)})$  converges to a critical point of  $E$ . Moreover, this critical point must a local minimum point or a saddle point of  $E$  by the energy-decreasing property. This completes the proof.

## References

- [1] A. Alpers, A. Brieden, P. Gritzmann, A. Lyckegaard, and H.F. Poulsen, *Generalized balanced power diagrams for 3D representations of polycrystals*, Philosophical Magazine 95 (2015), pp. 1016–1028, Available at <https://doi.org/10.1080/14786435.2015.1015469>.
- [2] H. Altendorf, F. Latourte, D. Jeulin, M. Faessel, and L. Saintoyant, *3D reconstruction of a multiscale microstructure by anisotropic tessellation models*, Image Analysis & Stereology 33 (2014), pp. 121–130, Available at <https://www.ias-iss.org/ojs/IAS/article/view/1090>.
- [3] J. Barker, G. Bollerhey, and J. Hamaekers, *A multilevel approach to the evolutionary generation of polycrystalline structures*, Computational Materials Science 114 (2016), pp. 54–63, Available at <http://www.sciencedirect.com/science/article/pii/S0927025615007259>.
- [4] D. Depriester and R. Kubler, *Radical Voronoi tessellation from random pack of polydisperse spheres: Prediction of the cells' size distribution*, Computer-Aided Design 107 (2019), pp. 37–49, Available at <https://doi.org/10.1016/j.cad.2018.09.001>.
- [5] P.J.J. Kok and F.N.M. Korver, *Modelling of complex microstructures in multi phase steels: geometric considerations for building an RVE*, in *Proceedings of the X International Conference on Computational Plasticity*. 2009.
- [6] A. Liebscher, *Laguerre approximation of random foams*, Philosophical Magazine 95 (2015), pp. 2777–2792, Available at <https://doi.org/10.1080/14786435.2015.1078511>.
- [7] A. Leonardi, P. Scardi, and M. Leoni, *Realistic nano-polycrystalline microstructures: beyond the classical Voronoi tessellation*, Philosophical Magazine 92 (2012), pp. 986–1005, Available at <https://doi.org/10.1080/14786435.2011.637984>.
- [8] A. Lyckegaard, E.M. Lauridsen, W. Ludwig, R.W. Fonda, and H.F. Poulsen, *On the use of Laguerre tessellations for representations of 3d grain structures*, Advanced Engineering Materials 13 (2011), pp. 165–170, Available at <https://onlinelibrary.wiley.com/doi/abs/10.1002/adem.201000258>.
- [9] L. Petrich, J. Staněk, M. Wang, D. Westhoff, L. Heller, P. Šittner, C.E. Krill III, V. Beneš, and V. Schmidt, *Reconstruction of grains in polycrystalline materials from incomplete data using Laguerre tessellations*, Microscopy and Microanalysis 25 (2019), pp. 743–752, Available at <https://doi.org/10.1017/S1431927619000485>.
- [10] I. Pérez, M. Muniz de Farias, M. Castro, R. Roselló, C. Recarey Morfa, L. Medina, and E. Oñate, *Modeling polycrystalline materials with elongated grains*, International Journal for Numerical Methods in Engineering 118 (2019), pp. 121–131, Available at <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6004>.
- [11] A. Spetl, T. Brereton, Q. Duan, T. Werz, C.E. Krill III, D.P. Kroese, and V. Schmidt, *Fitting Laguerre tessellation approximations to tomographic image data*, Philosophical Magazine 96 (2016), pp. 166–189, Available at <https://doi.org/10.1080/14786435.2015.1125540>.
- [12] O. Šedivý, D. Westhoff, J. Kopeček, C.E. Krill III, and V. Schmidt, *Data-driven selection of tessellation models describing polycrystalline microstructures*, Journal of Statistical Physics 172 (2018), pp. 1223–1246, Available at <https://doi.org/10.1007/s10955-018-2096-8>.
- [13] K. Teferra and D.J. Rowenhorst, *Direct parameter estimation for generalised balanced power diagrams*, Philosophical Magazine Letters 98 (2018), pp. 79–87, Available at <https://doi.org/10.1080/09500839.2018.1472399>.
- [14] Y. Wu, J. Cao, and Z. Fan, *Chord length distribution of Voronoi diagram in Laguerre geometry with lognormal-like volume distribution*, Materials Characterization 55 (2005), pp. 332–339, Available at <http://www.sciencedirect.com/science/article/pii/S1044580305001695>.

- [15] J. Alsayednoor, P. Harrison, and Z. Guo, *Large strain compressive response of 2-D periodic representative volume element for random foam microstructures*, *Mechanics of Materials* 66 (2013), pp. 7–20, Available at <http://www.sciencedirect.com/science/article/pii/S016766361300118X>.
- [16] S. Ghosh and D. Dimiduk, *Computational Methods for Microstructure-Property Relationships*, Springer, 2011.
- [17] S. Yadegari, S. Turteltaub, A.S.J. Suiker, and P.J.J. Kok, *Analysis of banded microstructures in multiphase steels assisted by transformation-induced plasticity*, *Computational Materials Science* 84 (2014), pp. 339–349, Available at <http://www.sciencedirect.com/science/article/pii/S0927025613007519>.
- [18] Y. Wu, W. Zhou, B. Wang, and F. Yang, *Modeling and characterization of two-phase composites by Voronoi diagram in the Laguerre geometry based on random close packing of spheres*, *Computational Materials Science* 47 (2010), pp. 951–961, Available at <http://www.sciencedirect.com/science/article/pii/S092702560900439X>.
- [19] Q. Duan, D. Kroese, T. Brereton, A. Spettl, and V. Schmidt, *Inverting Laguerre tessellations*, *The Computer Journal* 57 (2014), pp. 1431–1440.
- [20] K. Teferra and L. Graham-Brady, *Tessellation growth models for polycrystalline microstructures*, *Computational Materials Science* 102 (2015), pp. 57–67, Available at <https://doi.org/10.1016/j.commatsci.2015.02.006>.
- [21] O. Šedivý, T. Brereton, D. Westhoff, L. Polívka, V. Beneš, V. Schmidt, and A. Jäger, *3D reconstruction of grains in polycrystalline materials using a tessellation model with curved grain boundaries*, *Philosophical Magazine* 96 (2016), pp. 1926–1949, Available at <https://doi.org/10.1080/14786435.2016.1183829>.
- [22] M. Groeber and M. Jackson, *DREAM.3D: A digital representation environment for the analysis of microstructure in 3D*, *Integrating Materials* 3 (2014), pp. 56–72, Available at <https://doi.org/10.1186/2193-9772-3-5>.
- [23] 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 (2018), pp. 308–333, Available at <http://www.sciencedirect.com/science/article/pii/S0045782517307028>.
- [24] O. Šedivý, J. Dake, C.E. Krill III, V. Schmidt, and A. Jäger, *Description of the 3D morphology of grain boundaries in aluminum alloys using tessellation models generated by ellipsoids*, *Image Analysis & Stereology* 36 (2017), pp. 5–13, Available at <https://www.ias-iss.org/ojs/IAS/article/view/1656>.
- [25] B. Lévy and E.L. Schwindt, *Notions of optimal transport theory and how to implement them on a computer*, *Computers & Graphics* 72 (2018), pp. 135–148, Available at <http://www.sciencedirect.com/science/article/pii/S0097849318300098>.
- [26] F. Santambrogio, *Optimal transport for applied mathematicians*, Birkhäuser/Springer, 2015, Available at <https://doi.org/10.1007/978-3-319-20828-2>.
- [27] F. Aurenhammer, R. Klein, and D.-T. Lee, *Voronoi diagrams and Delaunay triangulations*, World Scientific, 2013, Available at <https://doi.org/10.1142/8685>.
- [28] A. Okabe, B. Boots, K. Sugihara, and S.N. Chiu, *Spatial tessellations: Concepts and applications of Voronoi diagrams*, 2nd ed., Wiley, 2000, Available at <https://doi.org/10.1002/9780470317013>.



- [29] F. Aurenhammer, F. Hoffmann, and B. Aronov, *Minkowski-type theorems and least-squares clustering*, *Algorithmica* 20 (1998), pp. 61–76, Available at <https://doi.org/10.1007/PL00009187>.
- [30] Q. Mérigot, *A multiscale approach to optimal transport*, *Computer Graphics Forum* 30 (2011), pp. 1583–1592, Available at <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2011.02032.x>.
- [31] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, 2004, Available at <https://doi.org/10.1017/CB09780511804441>.
- [32] B. Lévy, *A numerical algorithm for  $L_2$  semi-discrete optimal transport in 3D*, *ESAIM. Mathematical Modelling and Numerical Analysis* 49 (2015), pp. 1693–1715, Available at <https://doi.org/10.1051/m2an/2015055>.
- [33] J. Kitagawa, Q. Mérigot, and B. Thibert, *Convergence of a Newton algorithm for semi-discrete optimal transport*, *Journal of the European Mathematical Society (JEMS)* 21 (2019), pp. 2603–2651, Available at <https://doi.org/10.4171/JEMS/889>.
- [34] A. Brieden and P. Gritzmann, *On optimal weighted balanced clusterings: Gravity bodies and power diagrams*, *SIAM Journal on Discrete Mathematics* 26 (2012), pp. 415–434, Available at <https://doi.org/10.1137/110832707>.
- [35] D.P. Bourne and S.M. Roper, *Centroidal power diagrams, Lloyd’s algorithm, and applications to optimal location problems*, *SIAM Journal on Numerical Analysis* 53 (2015), pp. 2545–2569, Available at <https://doi.org/10.1137/141000993>.
- [36] S.-Q. Xin, B. Lévy, Z. Chen, L. Chu, Y. Yu, C. Tu, and W. Wang, *Centroidal power diagrams with capacity constraints: Computation, applications, and extension*, *ACM Transactions on Graphics* 35 (2016), pp. 244:1–244:12, Available at <http://doi.acm.org/10.1145/2980179.2982428>.
- [37] Q. Du, V. Faber, and M. Gunzburger, *Centroidal Voronoi tessellations: applications and algorithms*, *SIAM Review* 41 (1999), pp. 637–676, Available at <https://doi.org/10.1137/S0036144599352836>.
- [38] B. Bollobás and N. Stern, *The optimal structure of market areas*, *Journal of Economic Theory* 4 (1972), pp. 174–179, Available at [https://doi.org/10.1016/0022-0531\(72\)90147-0](https://doi.org/10.1016/0022-0531(72)90147-0).
- [39] P.M. Gruber, *Optimum quantization and its applications*, *Advances in Mathematics* 186 (2004), pp. 456–497, Available at <https://doi.org/10.1016/j.aim.2003.07.017>.
- [40] R.M. Gray and D.L. Neuhoff, *Quantization*, *Institute of Electrical and Electronics Engineers. Transactions on Information Theory* 44 (1998), pp. 2325–2383, Available at <https://doi.org/10.1109/18.720541>, information theory: 1948–1998.
- [41] S. Graf and H. Luschgy, *Foundations of quantization for probability distributions*, Vol. 1730, Springer, 2000, Available at <https://doi.org/10.1007/BFb0103945>.
- [42] C.H. Rycroft, *VORO++: A three-dimensional Voronoi cell library in C++*, *Chaos: An Interdisciplinary Journal of Nonlinear Science* 19 (2009), p. 041111, Available at <https://doi.org/10.1063/1.3215722>.
- [43] Firman, *Fast bounded power diagram*, <https://uk.mathworks.com/matlabcentral/fileexchange/56633-fast-bounded-power-diagram>, MATLAB Central File Exchange.
- [44] F. Aurenhammer, *Power diagrams: properties, algorithms and applications*, *SIAM Journal on Computing* 16 (1987), pp. 78–96, Available at <https://doi.org/10.1137/0216006>.

- [45] F. McCollum, *Power diagrams*, <https://uk.mathworks.com/matlabcentral/fileexchange/44385-power-diagrams>, MATLAB Central File Exchange.
- [46] A. Spettl, T. Werz, C. Krill III, and V. Schmidt, *Parametric representation of 3d grain ensembles in polycrystalline microstructures*, *Journal of Statistical Physics* 154 (2014), pp. 913–928, Available at <https://doi.org/10.1007/s10955-013-0893-7>.
- [47] M. Hifi and R. M’Hallah, *A literature review on circle and sphere packing problems: Models and methodologies*, *Advances in Operations Research* 2009 (2009), pp. 150624:1–150624:22.
- [48] M. Emelianenko, L. Ju, and A. Rand, *Nondegeneracy and weak global convergence of the Lloyd algorithm in  $\mathbb{R}^d$* , *SIAM Journal on Numerical Analysis* 46 (2008), pp. 1423–1441, Available at <https://doi.org/10.1137/070691334>.
- [49] D. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 3rd ed., Springer, 2008.
- [50] Q. Du, M. Emelianenko, and L. Ju, *Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations*, *SIAM Journal on Numerical Analysis* 44 (2006), pp. 102–119, Available at <https://doi.org/10.1137/040617364617364>.