

EIKONAL-BASED MODELS OF RANDOM TESSELLATIONS

BRUNO FIGLIUZZI

Centre for Mathematical Morphology, Mines ParisTech – PSL Research University, Fontainebleau, 77300, France

e-mail: bruno.figliuzzi@mines-paristech.fr

(Received November 20, 2018; accepted March 7, 2019)

ABSTRACT

In this article, we propose a novel, efficient method for computing a random tessellation from its level sets representation at each voxel of a discretized domain. This method is based upon the solution of the Eikonal equation and has a complexity in $O(N \log N)$, N being the number of voxels used to discretize the domain. By contrast, evaluating the implicit functions of the level sets representation at each voxel location has a complexity of $O(N^2)$ in the general case. The method also enables us to consider the generation of tessellations with rough interfaces between cells by simulating the growth of the germs on a domain where the velocity varies locally. This aspect constitutes the main contribution of the article. A final contribution is the development of an algorithm for estimating the multi-scale tortuosity of the boundaries of the tessellation cells. The algorithm computes the tortuosity of the boundary at several scales by iteratively deforming the boundary until it becomes a straight line. Using this algorithm, we demonstrate that depending on the local velocity model, it is possible to control the roughness amplitude of the cells boundaries.

Keywords: eikonal equation, fast marching algorithm, stochastic geometry, Voronoi tessellations.

INTRODUCTION

Most materials used in contemporary life and industry are heterogeneous and exhibit a complex internal microstructure. The microstructure is a key feature of the global material, which largely determines most of its physical properties at the macroscopic level (Jeulin, 1991; Ohser and Schladitz, 2009; Torquato, 2013). As a consequence, one is often interested in generating random microstructures that reproduce accurately some geometrical features of the original material (Moreaud *et al.*, 2012; Wang *et al.*, 2015). The simulated microstructures can in turn serve as a basis to investigate the physical or mechanical properties of heterogeneous materials through extensive numerical simulations (Gasnier *et al.*, 2015; Figliuzzi *et al.*, 2016; Koishi *et al.*, 2017). This approach can notably be used to understand the influence of the microstructure on the physical properties of the material at the macroscopic scale, or to determine microstructures yielding optimized functional properties for some considered application.

Stochastic geometry models provide useful tools to describe the complex microstructures observed in heterogeneous materials (Chiu *et al.*, 2013). A significant amount of work can therefore be found in the literature related to microstructure simulation based upon classical morphological models (Jeulin, 1991; 2012; 2015). Morphological models can for instance deal with the simulation of microstructures including random inclusions in a matrix or random

crystal structures. A general introduction to this topic can be found in Jeulin (1991). An efficient way to generate a random geometric model of a material microstructure is to rely on implicit functions and level sets to describe the geometry. Implicit functions and level sets cannot be directly used to conduct a numerical computation of the physical properties of the materials. The computation is indeed usually performed using a finite elements method or a fast Fourier transform method, which both require the geometry to be represented on a discretized grid of voxels. In practice, the discretization is computed by evaluating the implicit functions representing the microstructure at each voxel location, which can lead to expensive computations.

In this article, we propose a method for computing a random tessellation from its level sets representation at each voxel of a discretized domain based upon the solution of the Eikonal equation. The Eikonal equation describes the propagation of a wave from an initial set of boundaries on a domain where the local propagation velocity can vary (Malladi *et al.*, 1995; Sethian, 1996). Efficient algorithms have been developed over the years to solve the Eikonal equation on a domain. The fast marching algorithm has for instance a complexity in $O(N \log N)$, N being the number of voxels used to discretize the domain. By contrast, evaluating the implicit functions of the level sets representation of the tessellation at each voxel location has a complexity of $O(N^2)$ in the general case. In addition, by relying on the developed methodology,

we were able to develop a novel stochastic model enabling the generation of tessellations with rough interfaces between cells (Lee and Cowan, 1994; Lee, 1999; Jeulin, 2015). Models of tessellation with rough boundaries can find applications to describe microstructures for which the area of contact between components is a key geometrical feature that strongly influences the effective physical properties of the materials at the macroscopic scale (Bortolussi *et al.*, 2018).

The article is organized as follows. In section “Mathematical model”, we present classical stochastic models used to generate random tessellations, before introducing a new generation model based upon the solution of the Eikonal equation. Finally, we present the application of the model to the generation of tessellations with rough interfaces. In section “Numerical experiments”, we present and discuss the results of the Eikonal based model. To that end, we introduce a novel multi-scale tortuosity descriptor, which allows us to quantify the roughness at the boundaries of the cells of the tessellation. Conclusions are drawn in the last section.

MATHEMATICAL MODEL

VORONOI AND JOHNSON-MEHL TESSELLATIONS

Let Ω denote a compact observation window in \mathbb{R}^D . We consider a Poisson point process \mathcal{P} in Ω with intensity θ . A Poisson point process on Ω is a point process such that the number $N(K)$ of points contained in any compact subset K of Ω is a Poisson random variable with parameter $\theta V(K)$:

$$P\{N(K) = k\} = \frac{(\theta V(K))^k}{k!} \exp(-\theta V(K)), \quad (1)$$

where $V(K)$ denotes the volume of the compact subset K . For a given realization of \mathcal{P} in Ω , we denote by $\{p_1, p_2, \dots, p_n\}$ the implanted Poisson points. It follows from the definition of the Poisson distribution (Eq. 1) that the average number N of Poisson points in the domain Ω is

$$N = \theta V(\Omega). \quad (2)$$

It is possible to construct a tessellation based upon the Poisson point process \mathcal{P} by constructing cells which associate each point of Ω to the closest point p_i , $i = 1, \dots, N$ of the point process. In mathematical terms, if the set of points $\{p_1, p_2, \dots, p_n\}$ is a realization of \mathcal{P} , then we can associate to each point p_i with $i = 1, \dots, n$ the cell C_i defined by

$$C_i = \{p \in \Omega, \forall j \neq i, \|p_i - p\| \leq \|p_j - p\|\}. \quad (3)$$

Obviously, we have

$$\bigcup_{i=1}^n C_i = \Omega, \quad (4)$$

so that the cells C_i , $i = 1, \dots, n$ form a tessellation of Ω . This tessellation is referred to as the Voronoi tessellation in the stochastic geometry literature. Voronoi tessellations have been extensively studied in the literature. We refer the reader interested in their properties to the references Jeulin (1991) and Chiu *et al.* (2013).

Other types of tessellations can be constructed from a Poisson point process \mathcal{P} . For instance, let us assume that each point of \mathcal{P} carries a mark t_i . For sake of simplicity, we assume that the mark t_i is sampled according to the uniform distribution on the interval $[0, T]$ for some $T > 0$. Then, an alternative to the Voronoi tessellation is to consider the tessellation constructed with the cells

$$C_i = \{p \in \Omega, \forall j \neq i, t_i + \|p_i - p\| \leq t_j + \|p_j - p\|\}. \quad (5)$$

The tessellation resulting from this construction is known as the Johnson-Mehl tessellation in the literature. The Johnson-Mehl tessellation was originally introduced to model the growth of crystals. It is now a classical model in stochastic geometry (Johnson and Mehl, 1939; Møller, 1989; 1992; 1994), which has been employed to model numerous microstructures in materials engineering or geophysics applications (Ohser and Schladitz, 2009; Figliuzzi *et al.*, 2016; Belhadj *et al.*, 2018; Bortolussi *et al.*, 2018). The Johnson-Mehl tessellation can be seen as a sequential version of the Voronoi model, for which the Poisson points are implanted sequentially according to their mark t . From this perspective, it is convenient to interpret the marks as implantation times. All cells grow then isotropically with the same velocity. The growth of the cell boundaries is stopped when two boundaries meet. When a Poisson point from \mathcal{P} is implanted in an already existing cell, it is immediately removed.

Simulations of random tessellations are often performed on a grid of points. One can however rely on a completely different approach based upon level sets. In this approach, the tessellation cells are described by implicit functions, which are real valued functions defined in the ambient space. The level sets of an implicit function Φ are described by an equation of the form $\Phi(p) = c$, for some constant c . A surface is described as a level set of the function Φ , most commonly the set of points for which $\Phi(p) = 0$. In this case, the points for which $\Phi(p) < 0$ correspond to the interior of the cell associated to the implicit function, the points for which $\Phi(p) > 0$ to its complementary

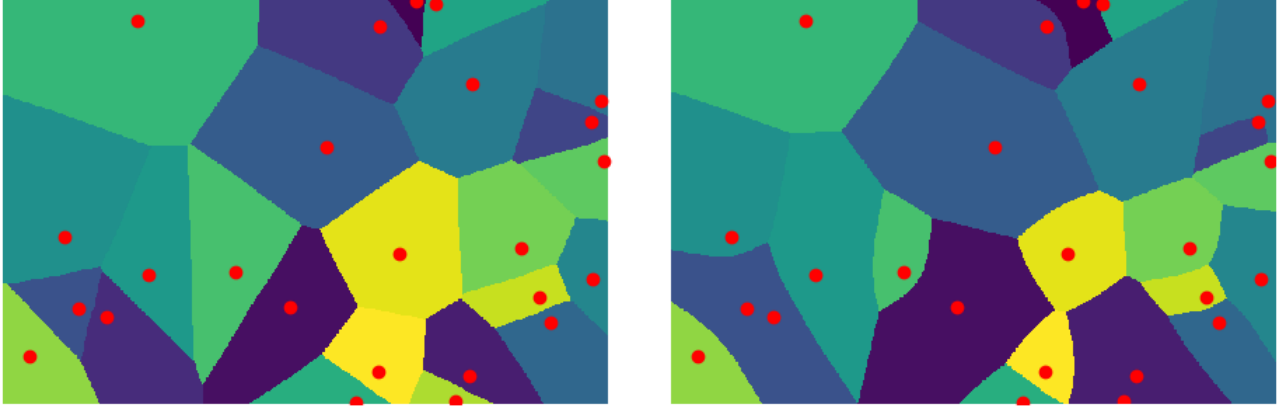


Fig. 1. Simulation of a random Voronoi tessellation (left) and of a Johnson-Mehl tessellation (right). We can note that while the boundaries of the Voronoi tessellation are straight lines, the boundaries of the Johnson-Mehl are portions of hyperboles.

and the level set $\Phi(p) = 0$ to the boundary of the tessellation cell. In the Voronoi case, the implicit function associated to cell i can be expressed as

$$\Phi(p) = \|p - p_i\|^2 - \inf_{1 \leq j \leq N, j \neq i} \|p - p_j\|^2.$$

Overall, using implicit functions to perform the simulation allows us to build complex combinations of simulations that we could not easily process with a pixel based method. Furthermore, implicit functions based simulations do not require a large amount of computer resources. However, for materials engineering applications, it is usually necessary to simulate images of the generated tessellation. To that end, the domain Ω is discretized and the implicit functions associated to the cells of the tessellation are evaluated at each voxel location. Generating an image from a vectorial simulation can be computationally expensive, especially when high resolution is required. If we denote by P the number of voxels in the image used to discretize the domain Ω , the average number of Poisson points in \mathcal{P} is proportional to P . Hence, the complexity of the image generation algorithm is on the order of $O(P^2)$.

EIKONAL BASED TESSELLATIONS

In this section, we relate the problem of generating an image from a vectorial simulation of a tessellation to the one of solving the Eikonal equation on a domain. The Eikonal equation is a non-linear partial differential equation which describes the propagation of waves in a medium. The Eikonal equation has the following form at each point p of the considered domain Ω :

$$\|\nabla t(p)\| = \frac{1}{u(p)}, \quad (6)$$

where ∇ is the gradient operator. t and u can be conveniently interpreted as the propagation time of the waves and as the local velocity, respectively. Eq. 6 requires initial conditions to be specified. Let us assume that we have sampled a vectorial simulation of the Johnson-Mehl tessellation model. We denote by N the number of Poisson points in \mathcal{P} , $N = |\mathcal{P}|$, and by (τ_1, \dots, τ_N) the corresponding germination times. Then, we consider the following problem:

$$\begin{cases} \|\nabla t(p)\| = \frac{1}{u(p)} \\ t(p_i) = \tau_i, \forall i = 1, \dots, N. \end{cases} \quad (7)$$

For all p in Ω , we denote by d the signed distance function defined by

$$d(p, \mathcal{P}) = \inf_{i=1, \dots, N} \|p - p_i\|. \quad (8)$$

Proposition 1 below (Sethian, 1996) relates the signed distance function d to the solution of the Eikonal equation on the domain Ω

Proposition 1 Let Ω be a subset of the Euclidean space \mathbb{R}^D . Then, the distance function is differentiable almost everywhere, and its gradient satisfies the Eikonal equation

$$\|\nabla t(p)\| = 1,$$

with initial conditions $t(p_i) = 0, \forall i = 1, \dots, N$.

An immediate consequence of proposition 1 is provided by the following corollary:

Corollary 1 On the domain Ω , when the velocity u is set to be constant on Ω , the solution t of the initial values problem (Eq. 7) satisfies, for all $p \in \Omega$,

$$t(p) = \inf_{i=1,\dots,N} (\tau_i + |p - p_i|). \quad (9)$$

In other words, the solution of Eq. 7 exactly matches the distance used for constructing the Johnson-Mehl tessellation. Solving the Eikonal equation therefore provides an efficient way to compute the Johnson-Mehl distance on a grid of points. Efficient algorithms are available for this task, including the fast marching algorithm described in the next section of the article.

FAST MARCHING ALGORITHM

The fast marching algorithm solves the Eikonal equation by iteratively computing the arrival times outwards from the initial conditions. In this article, we introduce a slightly modified version of the fast marching algorithm that computes the Johnson-Mehl distance from the germs in \mathcal{P} while keeping track of the labels of the initial germs during the propagation. To keep notations simple, we present the algorithm in a 2D setup. However, it is straightforward to apply the algorithm for higher dimensions. The algorithm is initialized as follows:

1. An arrival time map is initialized: each voxel p is associated the arrival time $t = \infty$, except if it contains the germ p_i in \mathcal{P} . In this case, the voxel p is associated the corresponding germination time τ_i .
2. A label map is initialized: the voxels p containing one germ p_i are associated the label i . The other voxels are associated the value 0 as label.
3. All voxels that contain one of the germs in \mathcal{P} and that have consequently been associated a finite time are regrouped in a set referred to as *narrow band*.

At each iteration, the voxel $P := (X, Y)$ of the narrow band with the smallest associated time is extracted and labeled as *frozen*. Arrival times are computed for its neighbors in a 4-neighborhood, which are in turn added to the narrow band. Note that frozen voxels are used to compute the arrival times in other voxels, but their arrival times is never recomputed. At a neighbor location $p := (x, y)$ of $P := (X, Y)$, the arrival time is computed by solving the Eikonal equation

$$\|\nabla t\|(x, y)u(x, y) = 1. \quad (10)$$

The difficulty here is to propose a correct way to approximate the gradient term. The following formula

is classically used to compute the squared length of the gradient at $p := (x, y)$ (Sethian, 1996):

$$\|\nabla t(x, y)\|^2 = \max(t_{x,y} - t_{x,y+1}, t_{x,y} - t_{x,y-1}, 0)^2 + \max(t_{x,y} - t_{x+1,y}, t_{x,y} - t_{x-1,y}, 0)^2. \quad (11)$$

This leads to the equation

$$\max(t_{x,y} - t_{x,y+1}, t_{x,y} - t_{x,y-1}, 0)^2 + \max(t_{x,y} - t_{x+1,y}, t_{x,y} - t_{x-1,y}, 0)^2 = \frac{1}{u(x, y)^2}. \quad (12)$$

In this equation, the only unknown is the arrival time $t(x, y)$ at point $p := (x, y)$. At this point, it is clear that

$$\max(t_{x,y} - t_{x,y+1}, t_{x,y} - t_{x,y-1}, 0)^2 = (t_{x,y} - \min(t_{x,y+1}, t_{x,y-1}))^2. \quad (13)$$

Hence, the arrival time $t(x, y)$ at point $p := (x, y)$ is solution of the quadratic equation

$$(t_{x,y} - \min(t_{x,y+1}, t_{x,y-1}))^2 + (t_{x,y} - \min(t_{x+1,y}, t_{x-1,y}))^2 = \frac{1}{u(x, y)^2}. \quad (14)$$

Eq 14 has two distinct solutions. However, to respect the consistency of the scheme, the arrival time must be higher than the time $t(X, Y)$ at the selected point P in the narrow band. Hence, the arrival time $t(x, y)$ is necessarily the largest solution of Eq. 14.

Once the arrival time $t(x, y)$ has been computed, two situations can be encountered. When the neighbor point (x, y) is in the narrow band, it has already been associated an arrival time $t_{old}(x, y)$. If $t(x, y) < t_{old}(x, y)$, then we affect the arrival time $t(x, y)$ to $p := (x, y)$ as well as the label of point $P := (X, Y)$. By contrast, if $t(x, y) > t_{old}(x, y)$, then the label and the arrival time at (x, y) remains unchanged. When the neighbor point (x, y) is not in the narrow band, we affect to it the arrival time $t(x, y)$, the label of (X, Y) and we add it to the narrow band.

At each iteration of the algorithm, it is necessary to extract the element of the narrow band with the smallest arrival time. To reduce the complexity of the algorithm, the elements of the narrow band are stored in a binary heap. We refer the reader interested in more details on the fast marching algorithm implementation to the original articles (Malladi *et al.*, 1995; Sethian, 1996; 1999a;b).

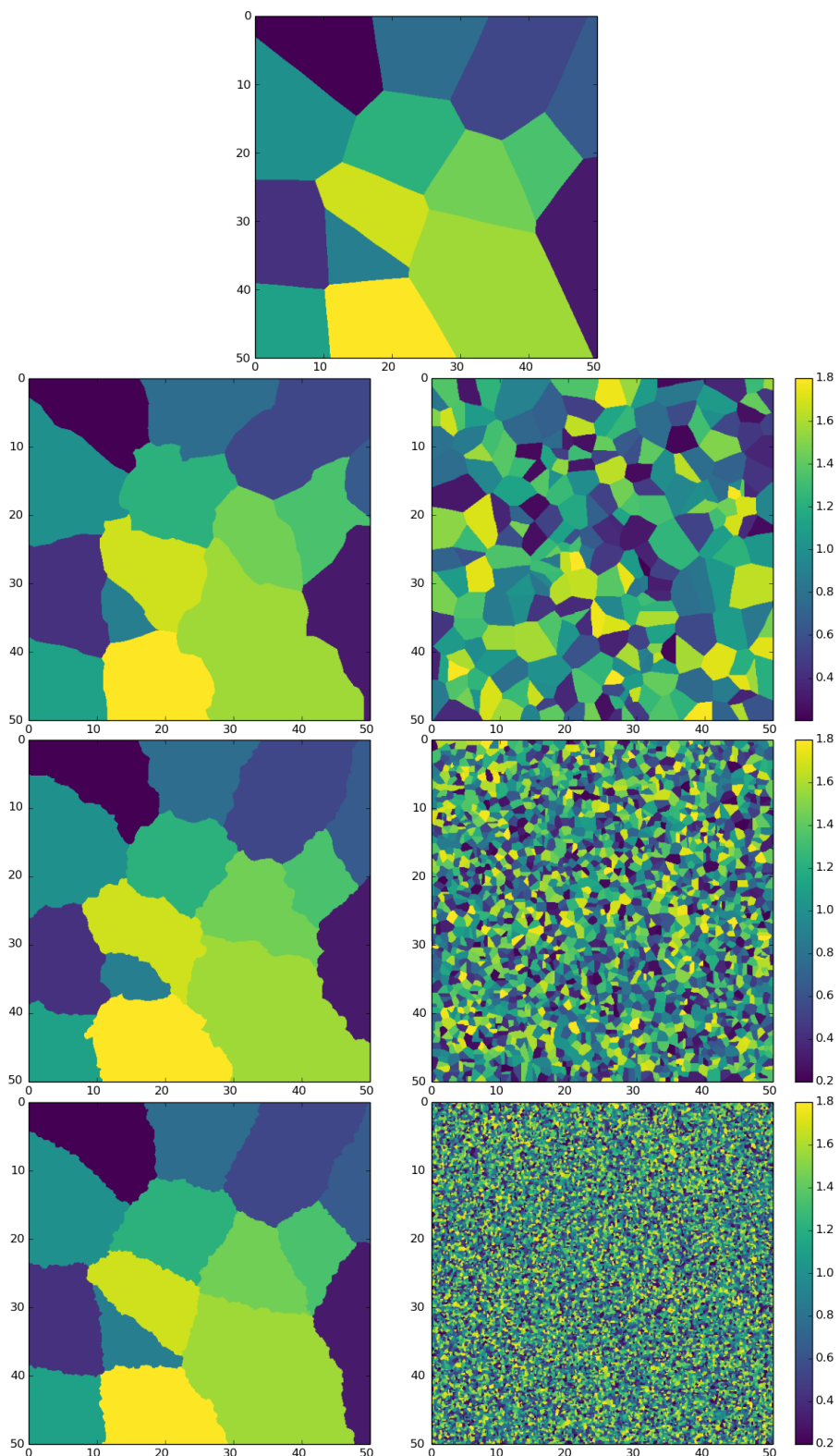


Fig. 2. Rough Voronoi tessellations (left) and corresponding local velocity field (right). All tessellations have been generated using the same Poisson point process \mathcal{P} on a 50×50 domain \mathcal{D} , discretized on a 500×500 regular grid. The local velocity fields are realizations of Voronoi tessellations with respective intensities $\theta = 0$, $\theta = 0.1$, $\theta = 1$ and $\theta = 10$. To account for edges effects, the level sets simulation of these tessellations are conducted on a domain larger than \mathcal{D} .

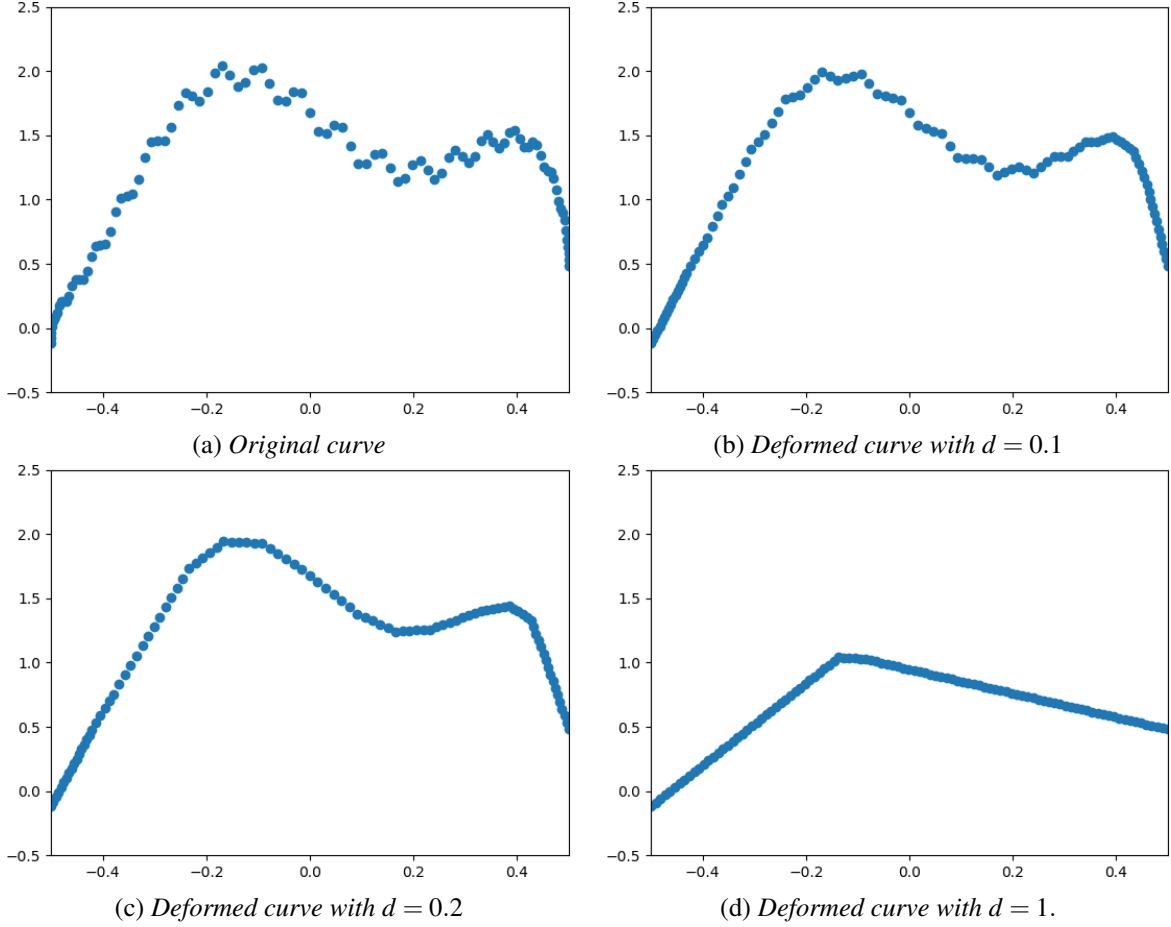


Fig. 3. *Illustration of the multi-scale tortuosity computation on a toy example.*

PROPAGATION ON A RANDOM VELOCITY FIELD

Up to this point, we have assumed that the velocity field used to compute the Johnson-Mehl tessellation was constant. However, solving the Eikonal equation also enables us to generate a tessellation of space built by growing germs on a field where the velocity varies from point to point. This observation opens the way to the development of tessellation models that cannot be directly obtained with classical approaches including the Voronoi and the Johnson-Mehl tessellations described previously. A potential application is for instance the generation of tessellations of space with rough boundaries between adjacent cells.

Let \mathcal{P} be the realization of a marked point process with intensity θ on an open domain Ω , for which the marks are drawn according to the uniform distribution $\mathcal{U}([0, L])$ on an interval $[0, L] \subset \mathbb{R}_+$. The points in \mathcal{P} as well as their respective marks allow to compute a Johnson-Mehl tessellation on the domain Ω by solving problem in Eq. 7. In this section, we construct a random velocity field u on domain Ω by

1. computing a Voronoi tessellation of Ω based upon a Poisson point process \mathcal{P}_v with intensity θ_v , and
2. setting a random velocity drawn according to some distribution \mathcal{D} in each cell of the Voronoi tessellation.

By solving the problem

$$\begin{cases} \|\nabla t(p)\| = \frac{1}{u(p)}, \\ t(p_i) = \tau_i, \forall i = 1, \dots, N, \end{cases} \quad (15)$$

we obtain a tessellation of the domain Ω resembling the Johnson-Mehl tessellation generated by solving Eq. 7, but with rough boundaries between adjacent cells. Moreover, the roughness at each boundary is strongly related to the characteristic length of the velocity field fluctuations. This characteristic length depends upon the intensity θ_v of the Voronoi tessellation used to compute the velocity field. A large value of θ_v yields for instance a large number of small Voronoi cells, which results in velocity variations over small distances. Hence, the generated Johnson-Mehl tessellation exhibits boundaries with a high density of

roughness patterns with relatively small amplitude. By contrast, Johnson-Mehl tessellations generated from velocity fields constructed with a small value of θ_v , exhibit boundaries with a low density of roughness patterns with larger amplitudes.

NUMERICAL EXPERIMENTS

We present in this section numerical simulations of Voronoi tessellations with rough boundaries computed with different values of the velocity parameter θ_v . We quantify the roughness of the cells boundaries using a multi-scale tortuosity descriptor, which iteratively computes the tortuosity of smoothed versions of the boundary. The multi-scale tortuosity descriptor is introduced in the first part of this section.

MULTI-SCALE TORTUOSITY

Let \mathcal{C} be a connected curve in a domain $\Omega \subset \mathbb{R}^2$. We discretize the domain Ω on a regular grid with equal spacing h in both directions. The curve \mathcal{C} is also discretized and can be represented as a set of points of the grid:

$$\mathcal{C} = \{c_1, c_2, \dots, c_N\}. \quad (16)$$

We consider that two points of the curve are neighbors when they are connected by an edge of the regular grid. The discrete curve \mathcal{C} can then be conveniently represented by an adjacency graph \mathcal{G} connecting together the points $\{c_1, c_2, \dots, c_N\}$. From now on, we assume

1. that the graph \mathcal{G} is connected (C.1),
2. that each point c_i is at most connected to two points (C.2),
3. that there are no cycles in the graph (C.3).

Necessarily, when $N > 2$, two points in $\{c_1, c_2, \dots, c_N\}$ have a single neighbor, while the others have exactly two neighbors. We refer to the points with a single neighbor as the extremities of the curve \mathcal{C} . For convenience, we can re-index the set of points $\{c_1, c_2, \dots, c_N\}$ so that c_1 and c_N are the extremities of the curve, and so that for all n such that $1 < n < N$, the point c_n is connected to c_{n-1} and c_{n+1} . In this case, the length of the curve \mathcal{C} is given by the expression

$$L_{\mathcal{C}} := \sum_{i=1}^{N-1} \|c_{i+1} - c_i\|. \quad (17)$$

A straightforward method to estimate the tortuosity of \mathcal{C} is to consider the ratio of the length L of the curve and of the distance D_c between the extremities of \mathcal{C} ,

$$\tau = \frac{L}{D_c}. \quad (18)$$

A limitation of this approach is that it does not involve the notion of scale in the tortuosity description. Hence, the tortuosity computed on a curve with repeated small roughness patterns can be the same as the tortuosity computed on a curve whose tortuosity arise at a larger scale. To address this issue, we propose to rely on a novel multiscale descriptor of the contour tortuosity computed by iteratively deforming the original contour.

To that end, let us denote by $d > 0$ some arbitrary distance. We consider the problem of finding the curve $\mathcal{Z}^d = \{z_1, \dots, z_N\}$ with minimal length sharing the same endpoints as \mathcal{C} and such that each point $z_{i,i=2,\dots,N-1}$ remains at a distance less than d from the corresponding point c_i of \mathcal{C} . In mathematical term, this can be formulated as the optimization problem

$$\begin{aligned} \mathcal{Z}^d = \arg \min \sum_{i=1}^{N-1} \|z_{i+1} - z_i\|^2, \\ \text{subject to } z_1 = c_1, z_N = c_N, \\ \forall n = 2, \dots, N-1, \|z_n - c_n\| \leq d. \end{aligned} \quad (19)$$

It is important to note that the objective function considered in Eq. 19 differs from the length of the curve, given by Eq. 17. The length of the curve is not a strictly convex function of the variables $z_i, i = 1, \dots, N$. In particular, the minimum of the function is not unique. By contrast, it is straightforward to note that the objective function used in Eq. 19 as well as the constraints are strictly convex, and that the minimum of this objective function is included in the set of the global minima of the function (Eq. 17). Hence, the considered optimization problem has a unique minimizer that can easily be estimated using the proximal gradient descent method or the projected gradient method. We refer the reader interested in more details on these algorithms to the article of Parikh and Boyd (2014).

Table 1. *Multiscale tortuosity parameters of the tessellations displayed in Fig 2. Each parameter has been computed on a basis of 20 simulations*

	$\theta_v = 0$	$\theta_v = 0.1$	$\theta_v = 1$	$\theta_v = 10$
$d = 1.$	1.23	1.36	1.41	1.44
$d = 2.$	1.	1.08	1.11	1.04
$d = 3.$	1.	1.06	1.05	1.01
$d = 4.$	1.	1.05	1.03	1.
$d = 5.$	1.	1.04	1.01	1.
$d = 6.$	1.	1.03	1.	1.
$d = 7.$	1.	1.02	1.	1.
$d = 8.$	1.	1.02	1.	1.
$d = 9.$	1.	1.01	1.	1.
$d = 10.$	1.	1.01	1.	1.
$d = 11.$	1.	1.	1.	1.

Let us consider boundary \mathbf{c}_{ij} separating cells i and j of the generated tessellation. We denote by $d_0 > 0$ some arbitrary distance. d_0 can for instance be chosen to be the voxel size. We define the tortuosity at scale $s = 0$ to be the tortuosity of the boundary $\mathbf{c}^{(0)} := \mathbf{c}_{ij}$:

$$\tau_0 = \frac{L(\mathbf{c}^{(0)})}{D(\mathbf{c}^{(0)})}, \quad (20)$$

where $D(\mathbf{c})$ is the Euclidean distance between the extremities of \mathbf{c} and $L(\mathbf{c})$ is the length of \mathbf{c} . The multi-scale tortuosity is computed iteratively as follows.

1. At step n , compute a smoother version $\mathbf{c}^{(n)}$ of the original boundary $\mathbf{c}^{(0)}$ by solving the optimization problem (19) with $d := d_n$.
2. Compute the tortuosity

$$\tau_n = \frac{L(\mathbf{c}^{(n)})}{D(\mathbf{c}^{(n)})}. \quad (21)$$

3. Update the distance, $d_{n+1} = d_n + d_0$.

The algorithm runs until the tortuosity is below $1 + \varepsilon$, $\varepsilon > 0$ being arbitrarily fixed.

The algorithm is illustrated on a toy example in Fig. 3. We can note that each iteration removes distinct scales of the roughness observed on the boundary.

NUMERICAL EXPERIMENT

The Eikonal based generation of rough Voronoi tessellations is illustrated in Fig. 2. In this figure, we generated four distinct Voronoi tessellations on a domain with size 50×50 discretized on a 500×500 regular grid from the same realization of a Poisson point process \mathcal{P} with intensity $\theta = 5 \times 10^{-2}$. We generated random velocity fields by first constructing a Voronoi tessellation of the domain, and then selecting a random velocity drawn from the uniform distribution \mathcal{U} on the interval $[0.2, 1.8]$ for each cell of the tessellation. The characteristic length λ_v of the velocity variation can be related to the intensity θ_v of the point process used to construct the germs of the tessellation. More precisely, the average area of the cell is given by

$$\bar{S} = \frac{1}{\theta_v}.$$

Hence, we can define the characteristic length of the velocity variations to be

$$\lambda_v = \sqrt{\frac{1}{\theta_v}}.$$

The fast marching algorithm returns a label image of the tessellation. We can extract the cells boundaries

by scanning the label image and by retaining the edges of the discretization grid separating voxels with distinct labels. This procedure yields a contour $\mathbf{c}_{i,j}$ for each couple of adjacent regions i and j . We obtain a representation of the boundaries as an adjacency graph by retaining the vertices of the edges of the discretization grid that constitute the boundary. Two vertices are considered to be adjacent if there is an edge of the discretization grid linking them. The corresponding adjacency graph is not necessarily connected. However, each connected component of the graph represents a connected portion of the boundary between two adjacent cells which satisfies conditions C.1, C.2 and C.3, and it is therefore possible to characterize the multi-scale tortuosity of the cells boundaries using the algorithm described in the previous section.

We computed the multi-scale tortuosity parameters for the tessellations displayed in Fig. 2 in Table 1. As expected, the tortuosity at the smallest scale is maximal for the tessellation with $\theta_v = 10$. By contrast, the largest tortuosity measured at subsequent scales correspond to the tessellation with $\theta_v = 0.1$. We note that the tortuosity of the original tessellation is not one at scale $d = 0$, despite the fact that the boundary of a Voronoi tessellation should be a straight line. This is due to artefacts caused by the discretization of the diagonal elements of the boundaries on a regular grid with preferential directions \mathbf{e}_x and \mathbf{e}_y .

CONCLUSION

In this article, we introduced a novel method for computing a random tessellation based upon the solution of the Eikonal equation. This method yields a complexity in $O(N \log N)$, N being the number of voxels used to discretize the simulation domain, and enabled us to generate tessellations with rough interfaces. In addition, we introduced an algorithm for characterizing the roughness of the cells boundaries at different scales. Using this algorithm, we showed that the local velocity model could be parameterized to control the roughness amplitude on the cells boundaries. A natural extension of the model presented in this article is the generation of random tessellations growing over an anisotropic velocity field. Another line of research could also be to generalize the multiscale characterization algorithm to the 3D setting. The difficulty here would be to propose a strictly convex objective function accounting for the smoothness of the curve. This will be the object of further work.

REFERENCES

- Belhadj J, Romary T, Gesret A, Noble M, Figliuzzi B (2018). New parameterizations for Bayesian seismic tomography. *Inverse Probl* 34(6):065007.
- Bortolussi V, Figliuzzi B, Willot F, Faessel M, Jeulin D (2018). Morphological modeling of cold spray coatings. *Image Anal Stereol* 37(2):145–58.
- Chiu SN, Stoyan D, Kendall WS, Mecke J (2013). *Stochastic geometry and its applications*. Chichester: Wiley.
- Figliuzzi B, Jeulin D, Faessel M, Willot F, Koishi M, Kowatari N (2016). Modelling the microstructure and the viscoelastic behaviour of carbon black filled rubber materials from 3D simulations. *Tech Mechanik* 32(1-2):22–46.
- Gasnier J-B, Willot F, Trumel H, Figliuzzi B, Jeulin D, Biessy M (2015). A Fourier-based numerical homogenization tool for an explosive material. *Mater Tech* 103(3):308.
- Jeulin D (1991). *Modèles morphologiques de structures aléatoires et de changement d'échelle*, PhD Thesis. Caen: Université de Caen Normandie.
- Jeulin D (2012). Morphology and effective properties of multi-scale random sets: A review. *C R Mécanique* 340(4-5):219–29.
- Jeulin D (2015). Boolean random functions. In: Schmidt V, eds. *Stochastic geometry, spatial statistics and random fields*. *Lect Not Math* 2120:143–69.
- Jeulin D (2017). Morphological models. In: Altenbach H, Öchsner A eds. *Encyclopedia of continuum mechanics*. Berlin: Springer.
- Johnson W, Mehl R (1939). Reaction kinetics in processes of nucleation and growth. *Trans Metal Soc AIME* 135(8):396–415
- Koishi M, Kowatari N, Figliuzzi B, Faessel M, Willot F, Jeulin D (2017). Computational material design of filled rubbers using multi-objective design exploration. In: Lion A, Johlitz M, eds. *Proc 10th Eur Conf Constit Model Rubbers (ECCMR)*.
- Lee T, Cowan R (1994). A stochastic tessellation of digital space. In: Serra J, Soille P, eds. *Mathematical morphology and its applications to image processing*. Dordrecht: Springer, pp 217–24.
- Lee T (1999). A stochastic tessellation for modelling and simulating colour aluminium grain images. *J Microsc* 193(2):109–26.
- Malladi R, Sethian J, Vemuri B (1995). Shape modeling with front propagation: A level set approach. *IEEE T Pattern Anal* 17(2):158–75.
- Møller J (1989). Random tessellations in \mathbb{R}^d . *Adv Appl Probab* 21(1):37–73.
- Møller J (1992). Random Johnson-Mehl tessellations. *Adv Appl Probab* 24(4):814–44.
- Møller J (1994). *Lectures on random Voronoi tessellations*. New York: Springer.
- Moreaud M, Jeulin D, Morard V, Revel R (2012). TEM image analysis and modelling: application to boehmite nanoparticles. *J Microsc* 245(2):186–99.
- Ohser J, Schladitz K (2009). *3D images of materials structures: processing and analysis*. Weinheim: Wiley-VCH.
- Parikh N, Boyd S (2014). Proximal algorithms. *Found Trends Optim* 1(3):127–239.
- Sethian J (1996). A fast marching level set method for monotonically advancing fronts. *P Natl Acad Sci USA* 93(4):1591–95.
- Sethian J (1999). *Level set methods and fast marching methods: Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge: Cambridge University Press.
- Sethian J (1999). Fast marching methods. *SIAM Rev* 41(2):199–235.
- Torquato S (2013). *Random heterogeneous materials: microstructure and macroscopic properties*. New York: Springer.
- Wang H, Pietrasanta A, Jeulin D, Willot F, Faessel M, Sorbier L, Moreaud M (2015). Modelling mesoporous alumina microstructure with 3D random models of platelets. *J Microsc* 260(3):287–301.