Automatic construction of fractal structures with locally controlled lacunarity

Boris Bordeaux
LIB, Université de Bourgogne
9 Avenue Alain Savary
BP 47870
21000, Dijon, France
boris.bordeaux@u-bourgogne.fr

Christian Gentil
LIB, Université de Bourgogne
9 Avenue Alain Savary
BP 47870
21000, Dijon, France
christian.gentil@u-bourgogne.fr

ABSTRACT

Lacunar fractal structures reduce the material quantity and weight while improving some physics properties, such as heat transfers, and preserving good mechanical properties. Nowadays, it is possible to construct such shapes thanks to additive manufacturing. This paper focuses on automatically generating subdivision rules for fractal lacunar structures with local topology control. The first main difficulty is guaranteeing topological consistency while assembling different cells to build a complicated multi-lacuna structure. The second is the adaptation of such shapes to geometric constraints like imposed boundaries. We address these questions throughout the formalism of the Boundary Controlled Iterated Function System. Then, we analyze the lacunarity and complexity of these structures from various geometric, topologic, and fractal measures.

Keywords

Fractals, Geometric Modeling, Lacunar Control, Iterative Modeling, Subdivision

1 INTRODUCTION

Saving energy by designing lighter objects while maintaining high physical properties is a crucial issue for the industry. The global structure must respond to several constraints, like mechanical resistance, energy absorption, heat transfer, and soundproofing, using a minimum quantity of material.

Porous metallic material is a pertinent solution for producing lighter objects. One can obtain metal foams from different techniques: mixtures of gas bubbles and a molten alloy or with the sintering and dissolution process [JZSL05] for instance. Because of the complexity and the randomness of the manufacturing process, it is challenging to guarantee expected properties [Ban06, JWZ07]. Then, the definition, the characterization, and the measure of the porosity arise in different domains to study their relations with physical properties [Esp12, AC15, JWZ07].

Thanks to additive manufacturing, we can produce lighter structures by designing controlled and structured geometry like lattice [TMV+16]. These structures generally fill a given volume with the minimum of matter. The most straightforward approach generates a strict periodic lattice as if one directly cuts the volume in a lattice block. The lattice comes from an initial periodic primary cell (thickened edges of a cube, 3D cross, gyroid) duplicated along the three 3D axes. This approach induces discontinuities at the boundaries of the filled volume. A parameter (thickness of beams, for instance) can also control the geometry of the initial cell to modulate its density and generate lattices with variable density to respond to specific physical properties [LJP+19]. Numerous surveys on this topic are available; see [PHL20, CLLZ21, FFL+18]. Softwares are available like [AKA21] to generate such lattices using optimal surfaces like gyroid as primary cells with variable density. An improvement of this method is to consider the geometry of the volume to adapt the lattice to its boundary to avoid discontinuities. In [KT10], Kou et al. introduce randomness to obtain irregular porous structures to produce variable density. More recently, McNulty et al. [MBZ+20] and Levo et al. [LVSZ21] propose bio-inspired approaches introducing multi-scale structures. Topological optimization also uses the multi-scale aspect. A second optimization step introduces a micro-lattice structure in

https://www.doi.org/10.24132/JWSCG.2024.1

under-constrained parts of the macro lattice, resulting from the first optimization step (see left Figure 1). However, the hierarchy level is generally shallow (one level for topological optimization and generally two levels for bio-inspired approaches). Furthermore, they are provided from ad-hoc implementation without formalizing the topological subdivision process and could be challenging to adapt to any situation.

We propose a new approach based on multi-scale lacunar structures resulting from fractal geometry, as shown in Figure 1 (middle and right). The motivations are: fractals appear as an optimal solution [LC19, AALS17] or are used as a model of bio-inspired structure for optimization [BRBH19]; fractal models can generate a great variety of topologies with different kinds of lacunarities and consequently interesting multi-physics properties such as heat and mass transfer [Pen10, Zha11, ZZ13], energy absorption [MGM$^+$18] or acoustical properties [AJS18, SHR97]; fractal modeling encompasses standard NURBS and subdivision surfaces, standard lattice structures, and multi-scale lattices [GGS21].

Producing fractal structures is easy by programming recursive algorithms. However, controlling the resulting shapes is more complicated. Some models like L-system or Boundary Controlled Iterated Function System, associated with topological constraints, allow accurate geometry control [TGM$^+$09, GGS21] and enable us to design varieties of lacunar structures with specific fractal topology.

Generating a lightweight structure by an assembly of fractal structures is a challenge. Difficulties come from the different topologies and geometries arising from fractal modeling while ensuring continuity to produce well-defined printable objects. Controlling the boundary topologies of cells makes the assembly process straightforward. For our purpose, we use the BC-IFS model. This model has the advantage of coding the fractal topology with adjacency and incidence constraints, controlling the geometry and the topology independently, and defining free-form shapes according to a set of control points.

As said before, this formalism provides powerful tools for designing various complicated multi-scale lacunar structures. However, currently, designers have to describe all the topological constraints by hand, which could be tedious. Nevertheless, once the topology is defined, the model guarantees the topological coherence, regardless of the geometric realization, which we adjust by control points and the iteration level.

The contribution of this paper is to provide parametrized algorithms that automatically generate a BC-IFS model of various multi-scale 2D structures with different lacunar topological complexities. Section 2 introduces the needed definitions and properties for the BC-IFS model. Section 3 shows a method to automatically define the subdivision rule of a 2D cell (a face) depending on the fractal topology of its boundary. We complete the lacunarity control with delay subdivisions in Section 4. In Section 5, we propose some measures to characterize the lacunarity and the complexity of fractal structures. Section 6 concludes this article and suggests future works.

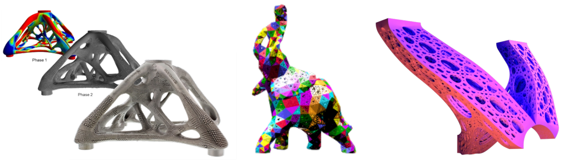


Figure 1: Left: Example of a topological optimization with two levels of lattice structures (©Altair Engineering). Middle: Elephant structure (from CGAL Computing Library at `https://github.com/CGAL/cgal`) built with multi-scale lacunas. Right: part of a mechanical system redesigned with multi-scale lacunar components shown in Figure 2.

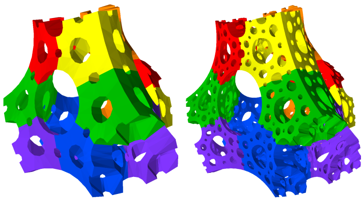


Figure 2: Subdivision of a truncated tetrahedron into some truncated tetrahedrons

2 BACKGROUND: BC-IFS MODEL

The Boundary Controlled Iterated Function System (BC-IFS) [SGGM15] model is based on Iterated Function System [Hut81, Bar90]. A set of contractive transformations defines an object by describing its self-similarity property, i.e., how this set of contractions subdivides the object into smaller copies of itself. By iterating on this set of transformations from an initial object (named the primitive), the obtained sequence converges to a unique shape called the attractor. In practice, the number of iterations is finite, giving an approximation of the attractor.

Recurrent IFS [BEH89] or C-IFS [ZT96] control the subdivision process using an automaton or a directed graph specifying the subdivision of an attractor into copies of itself or other attractors. Each state of the automaton defines a sub-attractor, and the outgoing transitions define the associated subdivision rules. By defining attractors in a barycentric space, C-IFS provide free-form fractal shapes.

Finally, Boundary C-IFS (BC-IFS) [SGGM15, GGS21] uses additional concepts to encode the cellular decomposition with incidence constraints: fractal volumes bounded by fractal surfaces bounded by fractal edges bounded by vertices. With a set of adjacency constraints between subdivided cells, we define the fractal topology of the attractor (see Figure 3). Incidence and adjacency constraints induce constraints on the BC-IFS transformations. They determine the topology, while the remaining degrees of freedom control the geometry.

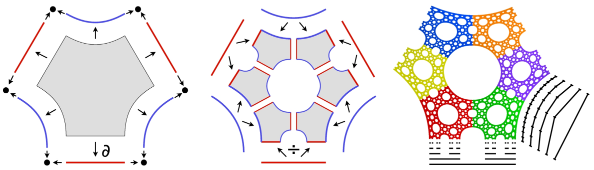


Figure 3: Example of definition 2D fractal topology. Left: the cellular decomposition. Middle: the topology subdivision. Right: the resulting attractor.

3 AUTOMATIC SUBDIVISIONS OF 2D CELLS

To build structures composed of lacunar cells with the BC-IFS model (see middle of Figure 1), one has to define a global shape that assembles different cells. The assembly uses adjacency constraints to ensure a consistent fractal topology at the junction of each cell with another. Then, one has to define the subdivision rule of each cell adapted to the wanted lacunarity (see Figures 3 and 4 for an example in 2D).

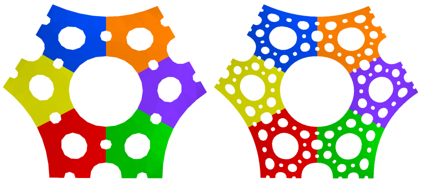


Figure 4: Subdivision of a hexahedron into six hexahedrons

Defining the subdivision process of each cell (including incidence and adjacency constraints) could be challenging for many cells. We want to automatize the definition of the subdivision process. To facilitate the assembly of fractal structure, we change the design paradigm by deducing face subdivision from imposed edges. We automatically define the subdivision of a face from its edges with a lacunarity control and deduce all incidence and adjacency constraints.

3.1 Notation

We classify two types of edge topology: B_n edges and C_n edges. The B_n edges subdivide into n connected sub-edges of type B_n (see Figure 5 for an example). The C_n edges subdivide into n disconnected sub-edges of type B_n (see Figure 6 for an example). In practice, B_n edges are Bézier edges, and C_n edges are Cantor edges.

We denote a fractal face by $\mathcal{F}(\partial, E, \mathcal{A})$ with ∂ its boundary, E a tuple of three edge topologies characterizing subdivision process, and $\mathcal{A}$ its subdivision algorithm.

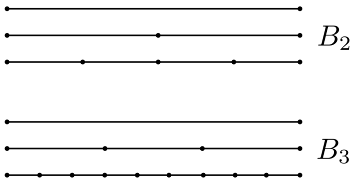


Figure 5: Example of subdivisions of B_2 and B_3 edges.

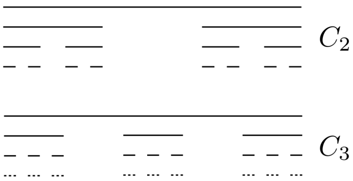


Figure 6: Example of subdivisions of C_2 and C_3 edges.

The parameter ∂ is a tuple of the face's edges in the counter-clockwise direction. We use the following notation $\partial = \lbag e_0, e_1, \dots e_N \rbag, e_i$ being the i-th edge. The order of the edges in ∂ is significant, up to a circular permutation. We use this specific notation to facilitate the comparison between boundaries. For instance if $\partial_1 = \lbag C_3, B_2, C_4, B_2, C_2, B_2 \rbag$ and $\partial_2 = \lbag B_2, C_2, B_2, C_3, B_2, C_4 \rbag$, they represent the same boundary because $\partial_1 = \partial_2$ with a circular shift of three elements.

The parameter E contains a tuple of three types of edges (E_a, E_l, E_c). The subdivision algorithm uses these edges to define the boundary of the sub-faces. Figure 7 shows where these edges appear after a subdivision. The choice of these edge types directly impacts the lacunarity.

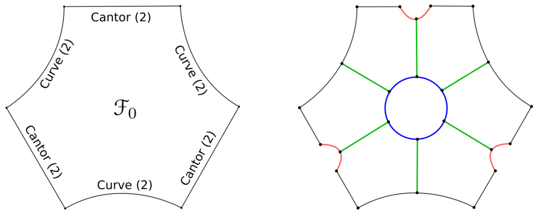


Figure 7: Example of a subdivision process. The adjacencies of the sub-faces are made along the edges E_a in green. The edges that form the central lacuna are the edges E_l in blue. The edges E_c in red are added between two C_n sub-edges.

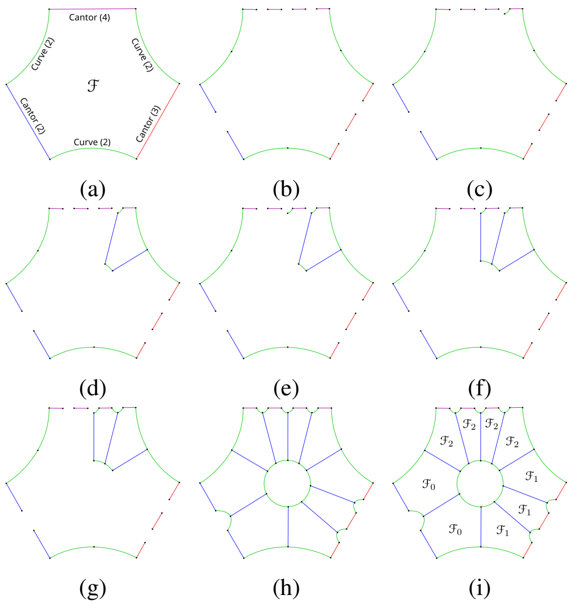
In this paper, we present several algorithms to define the subdivision of a face. The parameter $\mathcal{A}$ defines the algorithm used.

3.2 Algorithm

We present the algorithm $\mathcal{A}_1$ to automatize the definition of the subdivision rules of a fractal face from its edges. We present other algorithms in Section 4.3. Notice that the algorithm does not operate the subdivision process. It defines the rules of subdivision and all the adjacency and incidence constraints of the BC-IFS model. Figure 8 describes all the steps of the algorithm $\mathcal{A}_1$. It creates one sub-face for each pair of sub-edges in the corner of the face and one for each sub-edge that is not in a corner. The algorithm defines the incidence constraints for all sub-faces with the first (or the two firsts) edge(s) in the boundary. It also defines adjacency constraints between all adjacent subcells on their E_a edges. The sub-faces have the same parameters as the face, but their boundary differs. The algorithm works as follows:

1. Subdivide the boundary (all the edges) of the fractal face (a).
2. Selection of the sub-edges. The selection step takes, in the sub-edges of the face, the next sub-edge not in a corner, or the next two sub-edges in a corner. Each selection leads to a sub-face creation. The boundary of the sub-face starts with the selection. Define incidence constraints between the sub-edge of the face and the edges in selection.
3. Add to the boundary the E_c edge if the last edge in the current sub-face boundary is a C_n edge (c).
4. Add to the boundary the edges E_a, E_l and E_a in this order (d).
5. Add the E_c edge if the first edge in the sub-face boundary is a C_n edge.
6. Repeat from step 2 until all the sub-faces are created (no more selection).
7. Identify all sub-faces by their edges (i).
8. Define adjacency constraints between all sub-faces on their E_a edges.

We execute the algorithm on an example with the fractal face $\mathcal{F}(\partial, E, \mathcal{A}_1)$, having a boundary $\partial = \lbag C_4, B_2, C_2, B_2, C_3, B_2 \rbag$, and having $E = (C_2, B_2, B_2)$. Figure 8 details step-by-step the algorithm for this example. The first step subdivides the boundary. The second step is selection. The first selection is, for instance, the top right corner of the cell: the two sub-edges are B_2 and C_4. Hence, the boundary of the sub-face starts with $\partial_1 = \lbag B_2, C_4 \rbag$. The third step adds an E_c edge since the last edge was a C_n edge: $\partial_1 = \lbag B_2, C_4, B_2 \rbag$. The fourth step fills the boundary of the sub-face with the edges E_a, E_l and E_a: $\partial_1 = \lbag B_2, C_4, B_2, C_2, B_2, C_2 \rbag$. The fifth step adds a E_c edge if the first edge in the boundary is a C_n edge (not the case for ∂_1). We continue to the next sub-face. The next selection is a sub-edge not in a corner: a C_4 sub-edge. Hence, the boundary of the sub-face starts with $\partial_2 = \lbag C_4 \rbag$. Since the last edge in the boundary is a C_n edge, the third step adds the E_c edge: $\partial_2 = \lbag C_4, B_2 \rbag$. The fourth step adds the edges E_a, E_l and E_a to the boundary: $\partial_2 = \lbag C_4, B_2, C_2, B_2, C_2 \rbag$. The fifth step adds the E_c edge since the first edge is a C_n edge. Finally, $\partial_2 = \lbag C_4, B_2, C_2, B_2, C_2, B_2 \rbag$, and we continue so on for all the sub-faces.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 8: Algorithm $\mathcal{A}_1$ to define the subdivision rule of a face from its edges.

Figure 9 details five steps to intuitively understand the $\mathcal{A}_1$ algorithm's logic. An intuitive idea allows us to quickly create other algorithms based on the same principle. We introduce some algorithms in Subsection 4.3. The idea of the algorithm is the following:

1. Subdivide all edges (b).
2. Add two E_c edges between the sub-edges of the C_n edges (c).
3. Add E_l edges in the center of the face to create the lacuna (d).
4. Add the E_a edges between the external boundary and the central lacuna (e).
5. Identify all sub-faces by their edges (f).

3.3 Convergence

The algorithm defines the subdivision rule of a face into sub-faces. It can induce sub-faces with unspecified subdivision rules. Applying the algorithm recursively provides the subdivision rules for all sub-faces that appear during the process. For any initial face, a finite number of faces can appear over the recursion. All sub-faces contain only edges from the base face or the parameter E of the face (E_a, E_l, and E_c). Each sub-face contains a limited number of edges. The maximum is seven: a pair (C_n, C_n) of sub-edges in the corner, two edges E_c, and the three interior edges (two E_a and one E_l). The minimum is four: one B_n sub-edge (not in a corner) and

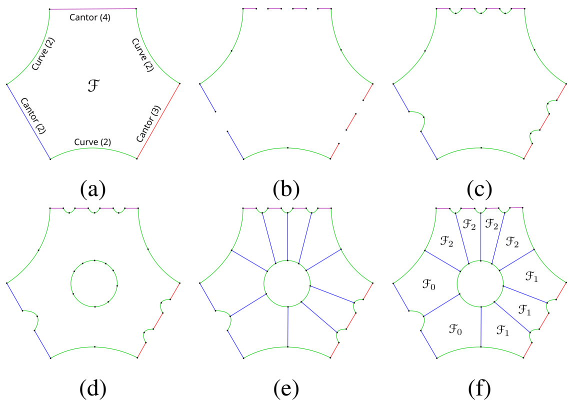


Figure 9: Intuitive algorithm to define the subdivision process of a face from its edges.

the three interior edges. Since there are a finite number of edge types and each sub-face has a finite number of edges, there are a finite number of different sub-faces. Hence, the recursion is guaranteed to terminate.

3.4 Lacunarity control

One can control the lacunarity by choosing the types of edges E_a, E_l, and E_c. Depending on that choice, one face often appears that subdivides into copies of itself. For instance the face $\mathcal{F}_0(\partial_0, E, \mathcal{A}_1)$ with $\partial_0 = \lbag C_2, B_2, C_2, B_2, C_2, B_2 \rbag$ and $E = (C_2, B_2, B_2)$ always appears in the sub-faces of all faces $\mathcal{F}(\partial, E, \mathcal{A}_1)$ after some iterations, regardless the boundary ∂. Figure 10 shows the face $\mathcal{F}_0$ subdivision.

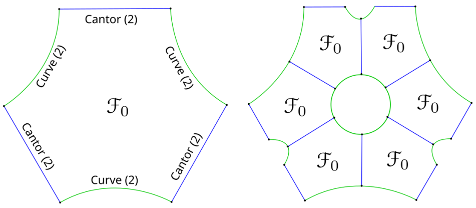


Figure 10: Subdivision of the face $\mathcal{F}_0(\partial_0, E, \mathcal{A}_1)$ with $\partial_0 = \lbag C_2, B_2, C_2, B_2, C_2, B_2 \rbag$ and $E = (C_2, B_2, B_2)$. It subdivides into six faces $\mathcal{F}_0$.

In the other cases, two or more faces appear, each subdividing into copies of themselves and copies of the other faces. Let be the faces $\mathcal{F}_1(\partial_1, E, \mathcal{A}_1)$ and $\mathcal{F}_2(\partial_2, E, \mathcal{A}_1)$ with $\partial_1 = \lbag C_2, B_2, C_2, B_2, C_2, C_2 \rbag$, $\partial_2 = \lbag C_2, C_2, C_2, C_2, B_2, C_2, C_2 \rbag$, and $E = (C_2, B_2, C_2)$. It is an example where $\mathcal{F}_1$ subdivides into four sub-faces $\mathcal{F}_1$ and two sub-faces $\mathcal{F}_2$, whereas $\mathcal{F}_2$ subdivides into five sub-faces $\mathcal{F}_2$ and two sub-faces $\mathcal{F}_1$. There is no face subdividing into several copies of itself, but there is a cycle between two faces. These faces appear in the sub-faces of all faces $\mathcal{F}(\partial, E, \mathcal{A}_1)$ after some iterations, regardless the boundary ∂.

4 DELAYED SUBDIVISIONS

We introduce a delay in subdivisions to have structures with faces with different local iteration levels. Without delay (left of Figure 11), the size of the lacunas might be tiny on the small face compared to the ones of the bigger faces. The delay allows us to have a uniform lacunarity when the size of the cells is not uniform (right of Figure 11). On the other hand, when faces have the same size, the delay induces a different lacunarity (see Figure 14). Hence, we propose delay subdivisions and some improvements in the presented algorithm to handle them.

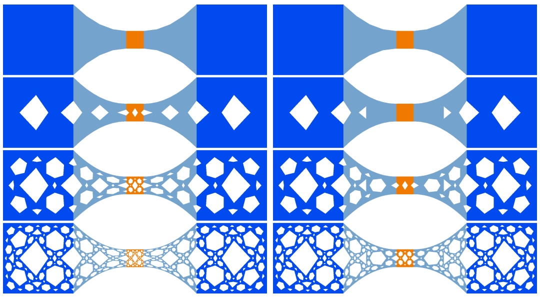


Figure 11: A fractal structure made of five faces without delay (left) and with delay (right). The central face is smaller than the others, but with a delay in its subdivision, its lacunas are not very small compared to the other lacunas.

4.1 Definition

A face with a delay in its subdivision is a face that induces no lacuna after its subdivision but has only its edges subdivided to preserve the consistency of the structure if the face shares an edge with another face. We denote the delay by an exponent $d \geq 2$. The face $\mathcal{F}^d(\partial, E, \mathcal{A})$ subdivides into only one face $\mathcal{F}^{d-1}(\partial', E, \mathcal{A})$ with the boundary ∂' that contains the subdivided edges of ∂. The sub-edges of C_n edges are unconnected, so we add two E_c edges between each sub-edges. The Figure 12 explains the delay subdivision of the face $\mathcal{F}^2$ with a boundary $\partial = \lbag C_4, B_2, C_2, B_2, C_3, B_2 \rbag$. The sub-face boundary $\partial' = \lbag C_4, B_2, B_2, C_4, B_2, B_2, C_4, B_2, B_2, C_4, B_2, B_2, C_2, B_2, B_2, C_2, B_2, B_2, C_3, B_2, B_2, C_3, B_2, B_2, C_3, B_2, B_2 \rbag$.

We also introduce the notion of delay in the subdivision process of edges, also denoted by an exponent $d \geq 2$. The edge subdivides into only one sub-edge of type X_n^{d-1}. Hence, the geometry of the edge does not change over iterations while the edge has a delayed subdivision (i.e., while $d \geq 2$).

4.2 Example

One can combine a delayed subdivision on a face with a delayed subdivision on all its edges to create a global delay in the face subdivision process. It creates a subface that does not change the face's shape since it adds

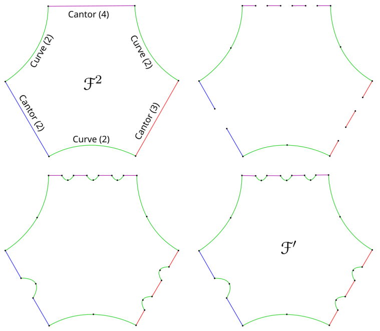


Figure 12: Algorithm to find the subdivision of a delayed face depending on its edges.

no lacuna inside the face, and the boundary does not change because of the delay on each edge. One can also mix delayed and non-delayed edges for one face to make a junction between a face having a global delay and a face having no delay at all.

For instance, we build a structure with three faces $\mathcal{F}_0^2(\partial_0',E,\mathcal{A}_1)$, $\mathcal{F}_1(\partial_1,E,\mathcal{A}_1)$ and $\mathcal{F}_0(\partial_0,E,\mathcal{A}_1)$ with $\partial_0 = \wr C_2,B_2,C_2,B_2,C_2,B_2 \int$, ∂_0' the same as ∂_0 but with a delay for each edge, $\partial_1 = \wr C_2,B_2,C_2^2,B_2,C_2^2,B_2,C_2^2,B_2 \int$, and $E = (C_2,B_2,B_2)$. Figure 13 shows a scheme of this structure. The faces $\mathcal{F}_0^2$ and $\mathcal{F}_0$ behave as the same face, but $\mathcal{F}_0^2$ has a delay of one iteration level because there is a delay in the face subdivision and also in all its edges subdivisions. The middle face $\mathcal{F}_1$ connects the faces $\mathcal{F}_0^2$ and $\mathcal{F}_0$ with its shared edge C_2 with the face $\mathcal{F}_0$ and its shared edge C_2^2 with the face $\mathcal{F}_0^2$.

At the first iteration level of the structure, the face $\mathcal{F}_0$ subdivides into six sub-faces $\mathcal{F}_0$ (see Figure 10). However, the face $\mathcal{F}_0^2$ subdivides into one sub-face $\mathcal{F}_0$; a delay appears here. Figure 14 shows the structure's third, fourth, and fifth iteration levels. We use adjacency constraints to make the junctions between the faces. Each face has a consistent subdivision thanks to the definition of all adjacency and incidence constraints (by the algorithm). Hence, the structure is consistent at all iteration levels.

4.3 Other subdivision algorithms

The presented algorithm $\mathcal{A}_1$ creates sub-faces with a shared edge (E_a) between two adjacent sub-faces. The adjacency edges start from the lacuna to the junction of two B_n edges (see Figure 9), but there are other solutions. For instance, Figure 15 shows several ways to subdivide the face $\mathcal{F}_0$. The sub-faces depend on how the algorithm places the adjacency edges from the central lacuna on the boundary. The algorithm $\mathcal{A}_1$ places the adjacency edges at the junction between two B_n edges (middle of Figure 15). The algorithm $\mathcal{A}_2$ places the same adjacency edges as $\mathcal{A}_1$ but also on all corners (right of Figure 15). When some edges of a face have a delayed subdivision, there are also several ways to subdivide the face, as shown in Figure 16. The algorithm $\mathcal{A}_1$ places the adjacency edges at the junction between two B_n edges (top right of Figure 16). The algorithm $\mathcal{A}_2$ places the same adjacency edges as $\mathcal{A}_1$ but also on all corners (bottom left of Figure 16). The algorithm $\mathcal{A}_3$ places the same adjacency edges as $\mathcal{A}_1$ but also on corners surrounding an edge with a delay subdivision (bottom right of Figure 16). When a face has no delay edges, $\mathcal{A}_3$ and $\mathcal{A}_1$ generate the same subdivision rule.

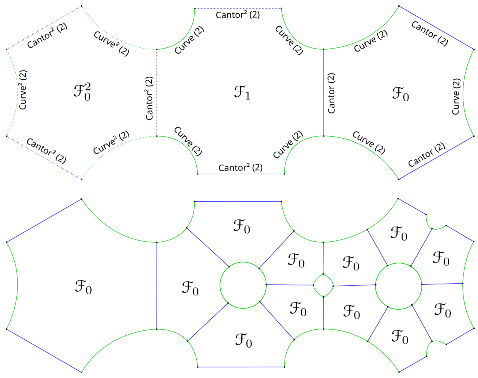


Figure 13: (Above) Scheme of a structure with three faces. The left one is the same as the right one but with a delay in its subdivision. The middle face makes the links between them. (Below) The structure's subdivision.

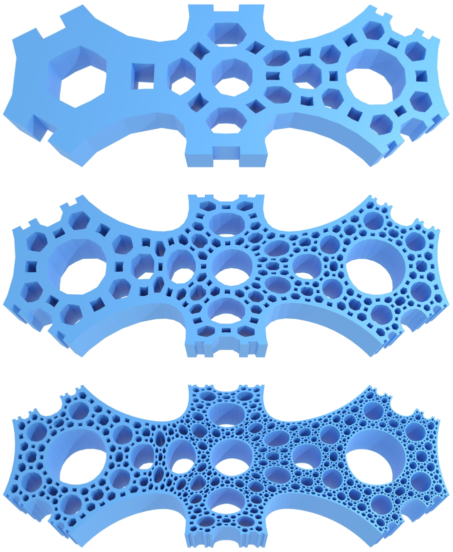


Figure 14: From top to bottom, the second, third, and fourth iteration levels of the extruded structure presented in Figure 13.

Choosing the suitable algorithm would depend on the specific application. However, to limit the number of face types, one has to stick to the same algorithm across all iterations rather than changing it at each level. Alternatively, one can use one algorithm when dealing with faces containing edges that have a delay in their subdivision and another algorithm for faces without such

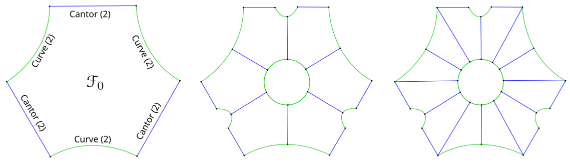


Figure 15: Different subdivisions of the face $\mathcal{F}_0$ with a boundary $\partial_0 = \lbag C_2, B_2, C_2, B_2, C_2, B_2 \rbag$.

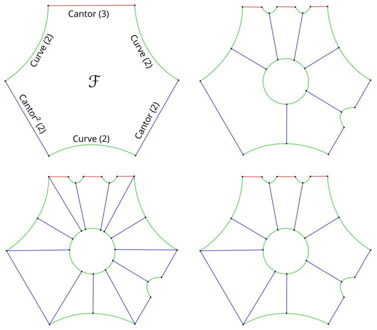


Figure 16: Different subdivisions algorithm of a face $\mathcal{F}$ with a boundary $\partial = \lbag C_3, B_2, C_2^2, B_2, C_2, B_2 \rbag$.

edges. There are a lot of possibilities to define the subdivision algorithm. The key is to ensure that when the algorithm is applied repeatedly, it leads to already-known faces (no creation of new face topologies). Note that for a structure containing several faces, one can use a different algorithm for each face in order to vary the subdivision. Figure 17 shows an example of two faces having the same boundary but a different algorithm.

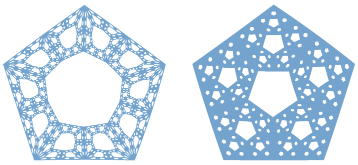


Figure 17: Two faces $\mathcal{F}_1(\partial, E, \mathcal{A}_1)$ and $\mathcal{F}_2(\partial, E, \mathcal{A}_2)$ with the same boundary $\partial = \lbag B_2, B_2, B_2, B_2, B_2 \rbag$ and the same $E = (B_2, B_2, B_2)$. The algorithm $\mathcal{A}_2$ (left) creates more lacunas than $\mathcal{A}_1$ (right).

5 MEASURES ON FRACTAL STRUCTURES

To evaluate the variability of the lacunarity, one can use several measures like relative density [JWZ07], area, or length of boundary (perimeter). However, these measures strongly depend on the iteration level. To characterize fractal structures, evaluating the evolution of the area and the perimeter is convenient from one iteration to the following. The fractal dimension is especially recommended for such a structure. We propose an additional measure that characterizes the topological complexity of such structures. For the rest of the article, $\mathcal{F}_0$ is the face $\mathcal{F}_0(\partial_0, E, \mathcal{A}_1)$ with $\partial_0 = \lbag C_2, B_2, C_2, B_2, C_2, B_2 \rbag$ and $E = (C_2, B_2, B_2)$. Its geometric realization is in Figure 18.

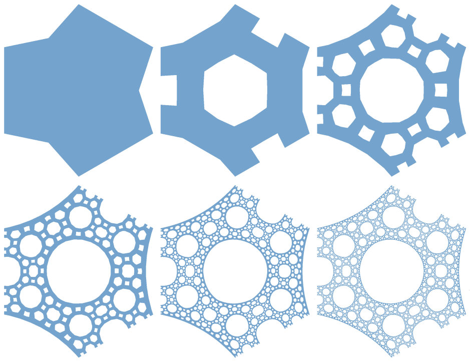


Figure 18: First five iteration levels of the face $\mathcal{F}_0$ we use for all geometry based measures. Top-left image is the primitive.

5.1 Area and perimeter

For lacunar fractal structures, the area A_i decreases, and the perimeter P_i increases with each iteration. We look at how these values are changing through iteration levels. For instance, with the face $\mathcal{F}_0$, we compute A_i and P_i for all iterations. To normalize the results and see the factor F_A (respectively F_P) multiplying the area (respectively the perimeter) at each iteration, we compute A_i/A_0 and P_i/P_0. Table 1 contains the values for the area and the perimeter of the face $\mathcal{F}_0$ for the iteration levels 0 to 5. Figure 19 (respectively 20) shows the variation of the area (respectively the perimeter) through iteration levels. We have $F_A = 0.76$ and $F_P = 1.97$, meaning the area decreases by 24% while the perimeter increases by 97% for each iteration level.

i	A_i	P_i	$\ln(A_i/A_0)$	$\ln(P_i/P_0)$
0	234.90	61.61	0	0
1	174.49	97.96	-0.30	0.46
2	132.50	185.63	-0.57	1.10
3	100.92	381.78	-0.84	1.82
4	76.92	807.93	-1.12	2.57
5	58.64	1725.87	-1.39	3.33

Table 1: Area and perimeter of the face $\mathcal{F}_0$ for each iteration level i.

5.2 Relative density

We compute the relative density [JWZ07] of a fractal structure at different scale levels by using an image of that structure and a mask with variable size. The idea is to have a local evaluation of the matter quantity. We drag the mask over the image for each pixel of the structure and count the number of pixels belonging to the structure after the mask application. Then, we divide the value by the surface of the mask. Finally, we color the image's pixel at the mask's center depending on the relative density found (between 0 and 1) using the colormap in Figure 21. Figure 22 contains the resulting image for the third iteration level of the face $\mathcal{F}_0$ for dif-

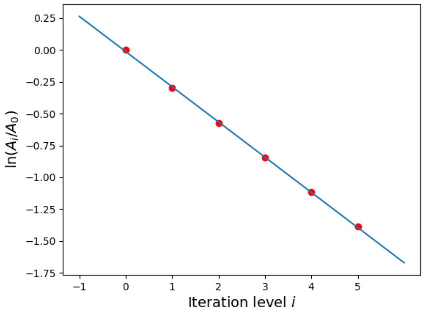


Figure 19: For each iteration level, the logarithm of the area over the area of the primitive for the face $\mathcal{F}_0$. The area decreases by 24% for each iteration level.

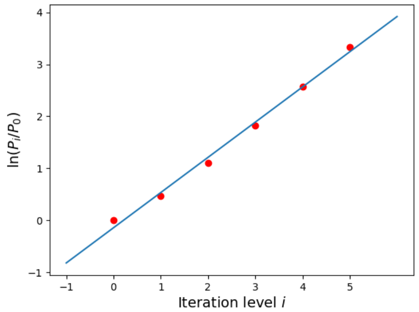


Figure 20: For each iteration level, the logarithm of the perimeter over the perimeter of the primitive for the face $\mathcal{F}_0$. The perimeter increases by 97% for each iteration level.

ferent mask sizes. Figure 23 contains the relative density for the structure in Figure 13. The structure's left side has a higher density than the right side because of the delay in subdivision. The additional iteration adds more lacunas on the right side, reducing the relative density.

Figure 21: The colormap we use for the relative density. A value of 0 is mapped on the blue and a value of 1 is mapped on the red. Any other value between 0 and 1 is mapped linearly between blue and red.

5.3 Fractal dimension

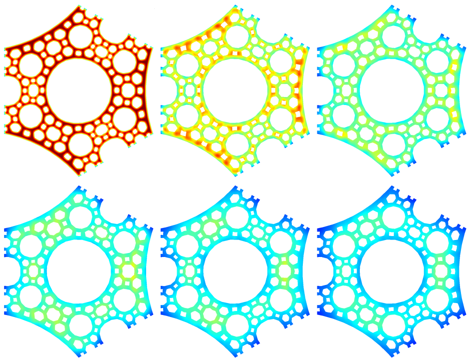


Figure 22: Relative density on the third iteration level of the face $\mathcal{F}_0$ for different mask sizes. In the reading direction, the mask sizes are 21, 51, 101, 151, 201 and 251 pixels. The image has a size of 825 × 953 pixels.

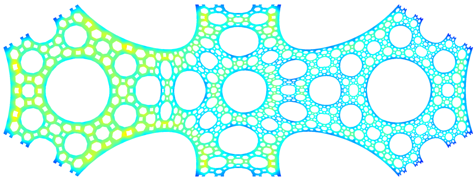


Figure 23: Relative density on the fractal structure in Figure 13 for a mask size of 101 pixels. The image has a size of 2607 × 953 pixels.

Using the box-counting algorithm, we compute the fractal dimension and note it D_{box} [Man85]. The fractal dimension is the slope of the line when we plot the values of $\ln(N_\varepsilon)$ on the Y-axis against the value of $\log(1/\varepsilon)$ on the X-axis. We note ε the size of the boxes and N_ε the number of boxes of size ε that cover the geometry. For a given geometry, we set it in a squared grid of size one by one. We start with a box size of $1/16$ since bigger box sizes are insignificant. We use the geometry of the face $\mathcal{F}_0$ (in Figure 18) to compute all geometry-based measures. With the fifth iteration level, we compute that the face has a fractal dimension of $D_{box} = 1.65$. Table 2 contains the values for each size of boxes, and Figure 24 represents the corresponding line.

ε	$\ln(1/\varepsilon)$	$\ln(N_\varepsilon)$
$1/16$	2.77	4.96
$1/32$	3.47	6.09
$1/64$	4.16	7.28
$1/128$	4.85	8.46
$1/256$	5.55	9.62
$1/512$	6.24	10.71
$1/1024$	6.93	11.80

Table 2: Box-counting method on the fifth iteration of the face $\mathcal{F}_0$.

5.4 Topological measure of fractals

One cannot use the Euler-Poincaré characteristic to characterize the topology of a fractal structure because of the infinite increase of the number of lacunas. To address this question, we suggest computing the ratio limit between the number of lacunas and the sum of the number of lacunas and sub-faces. This measure does not depend on the geometric realization. For a given iteration level, we consider the space as a partition composed of sub-faces and lacunas. We quantify the ratio of the space lacunas occupy. Then, the topological measure of a fractal is the limit of this ratio when the iteration level tends to infinity. This ratio gives information on the number of created lacunas

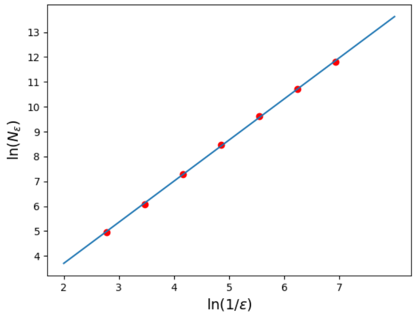


Figure 24: For the fifth iteration of the face $\mathcal{F}_0$, the slope of the curve gives $D_{box} = 1.65$.

as a function of created sub-faces along the iterative process.

For a given iteration level i, we note the number of lacunas (respectively number of sub-faces) N_i^L (respectively N_i^C). We compute them recursively. For the primitive (when the iteration level is zero), there is no lacuna ($N_0^L = 0$). For the first iteration level, a no-delayed face has a central lacuna that counts as one lacuna. It can also have lacunas in its boundary due to C_n edges. We consider that each C_n edge creates $n-1$ half lacunas. When the iteration level is more than zero, N_i^L is the face's number of lacunas at the first iteration level plus the number of lacunas of each sub-face at the previous iteration level.

About the number of sub-faces, when the iteration level is zero, there is one face. For any other iteration level, the number of sub-faces is the sum of the number of sub-faces for each sub-face at the previous iteration level. When the subdivision of a face $\mathcal{F}$ gives n sub-faces $\mathcal{F}$, we compute the number of sub-faces with the equation 1 and the number of lacunas with the equation 2.

$$N_i^C = n^i \tag{1}$$

$$N_i^L = N_1^L + n \times N_{i-1}^L \tag{2}$$

For instance, we compute the values of N_i^C and N_i^L for the face $\mathcal{F}_0$. Figure 10 describes this face subdivision. When there is no iteration, $N_0^L = 0$ and $N_0^C = 1$. The face has three C_2 edges, each of which induces one-half lacuna at the next iteration level. Hence at the first iteration level, N_1^L of $\mathcal{F}_0 = 1 + 3 \times 0.5 = 2.5$. The subdivision of $\mathcal{F}_0$ gives six sub-faces $\mathcal{F}_0$, hence N_1^C of $\mathcal{F}_0 = 6$. For the second iteration level, we have N_2^L of $\mathcal{F}_0 = 2.5 + 6 \times 2.5 = 17.5$ and N_2^C of $\mathcal{F}_0 = 6 \times 6 = 36$. Since the subdivision of $\mathcal{F}_0$ gives six sub-faces $\mathcal{F}_0$, we could compute N_2^C as 6^2 that is also 36.

Let the ratio R_i be:

$$R_i = \frac{N_i^L}{N_i^C + N_i^L}$$

Table 3 indicates the topology-based measures of $\mathcal{F}_0$ for the first five iteration levels. The curve of the ratio R_i of $\mathcal{F}_0$ for the first ten iteration levels is in Figure 25. Notice that the ratio has a finite limit. We found that the limit depends only on the parameter E of the face. The algorithm creates more and more characteristic faces that depend on these parameters. Hence, the limit R of R_i when i tends to the infinity of any face depends on the limit R of the specific faces that appear due to the subdivision algorithm. For a characteristic face that subdivides into itself and from equations 1 and 2, we have the following formula:

$$R = \lim_{i \to +\infty} R_i = \frac{N_1^L}{N_1^L + N_1^C - 1}$$

For instance, with the face $\mathcal{F}_0$, the limit is:

$$R = \frac{2.5}{2.5 + 6 - 1} = \frac{1}{3}$$

i	N_i^L	N_i^C	R_i
0	0	1	0
1	2.5	6	0.294
2	17.5	36	0.327
3	107.5	216	0.332
4	647.5	1296	0.333
5	3887.5	7776	0.333

Table 3: Topology-based measures N_i^L, N_i^C and R_i of the face $\mathcal{F}_0$ for the iteration levels i from 0 to 5.

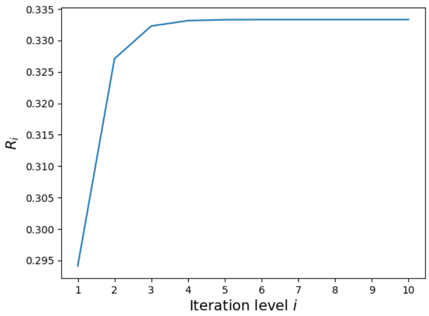


Figure 25: The ratio R_i of the face $\mathcal{F}_0$ for the iteration levels from 0 to 10.

5.5 Discussion

We created some faces that subdivide in themselves to study the impact of the algorithms and their parameter E in the lacunarity. The faces are the following:

- $\mathcal{F}_0$.
- $\mathcal{F}_1(\partial_1, E_1, \mathcal{A}_1)$ with $\partial_1 = \wr C_2, C_2, C_2, C_2, C_2, C_2, C_2 \wr$ and $E_1 = (C_2, C_2, C_2)$.
- $\mathcal{F}_2(\partial_2, E_2, \mathcal{A}_1)$ with $\partial_2 = \wr B_2, B_2, B_2, B_2, B_2 \wr$ and $E_2 = (B_2, B_2, B_2)$.

- $\mathcal{F}_3(\partial_3, E_1, \mathcal{A}_2)$ with $\partial_3 = \lbag C_2, C_2, C_2, C_2, C_2 \rbag$.
- $\mathcal{F}_4(\partial_4, E_2, \mathcal{A}_2)$ with $\partial_4 = \lbag B_2, B_2, B_2, B_2 \rbag$.
- $\mathcal{F}_5(\partial_5, E_3, \mathcal{A}_2)$ with $\partial_5 = \lbag B_3, B_3, B_3, B_3 \rbag$ and $E_3 = (B_3, B_3, B_3)$.

Figure 26 contains the faces at the fourth iteration level. Table 4 lists those faces with the value of N_1^L, N_1^C, $\lim_{i \to +\infty} R_i$, the factors F_A and F_P, and the fractal dimension at the fourth iteration level of the cells.

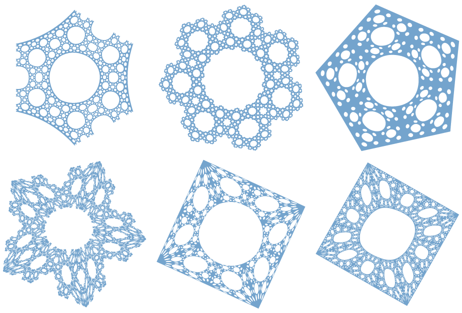


Figure 26: Some characteristic faces that subdivide into themselves. (Top left) The face $\mathcal{F}_0$. (Top middle) The face $\mathcal{F}_1$. (Top right) The face $\mathcal{F}_2$. (Bottom left) The face $\mathcal{F}_3$. (Bottom middle) The face $\mathcal{F}_4$. (Bottom right) The face $\mathcal{F}_5$.

Face	N_1^L	N_1^C	R	F_A	F_P	D_{box}
$\mathcal{F}_0$	2.5	6	0.33	0.76	1.97	1.72
$\mathcal{F}_1$	4.5	7	0.43	0.76	2.09	1.74
$\mathcal{F}_2$	1	5	0.20	0.85	1.69	1.82
$\mathcal{F}_3$	3.5	10	0.28	0.79	2.58	1.77
$\mathcal{F}_4$	1	8	0.12	0.75	1.92	1.72
$\mathcal{F}_5$	1	12	0.08	0.81	2.40	1.77

Table 4: List of measures for each face in Figure 26. There are the number of lacuna and the number of subfaces at the first iteration, the R value that is the limit of R_i when i tends to infinity, the factors F_A and F_P, and the fractal dimension.

Globally, the set of parameters we used to provide the different fractal faces impacts our topological measure R. The values depend on the subdivision algorithm and the parameter E. They reflect the specificity of the face lacunarity.

On the other hand, geometry measures are more delicate to interpret because they depend on geometric realization. However, we can observe that for two faces having an identical value of F_A, the value of F_P is higher when the parameter E contains the most C_n edges. Intuitively, C_n edges induce lacunas while B_n edges induce no lacunas.

Generally, the higher the fractal dimension, the higher the F_A and the lower the F_P, even if the fractal dimension is relatively homogeneous for these examples. However, for an identical value of F_A, the fractal dimension increases with the value of F_P.

6 CONCLUSION AND FUTURE WORK

Automatic construction of fractal structures while controlling the lacunarity is challenging because of the variety and complexity of possible topologies.

We propose an automatic method defining the topology of fractal faces's subdivision, depending on the topology of their boundary and additional parameters controlling the lacunarity. Our method presents three main advantages. First, a few intuitive sets of parameters define the type of lacunarity. Second, we base our construction on the BC-IFS model, which codes the topology of the fractal structure. It ensures the consistency of the limit fractal topology and over iterations. The resulting model can be exported in STL format and directly printed in additive manufacturing (see Figure 27). This topological approach allows us to adapt the lacunar geometry to intended applications. Third, as we define a lacunar face from its boundary, we can directly create complicated assemblies without additional constraints, with each face being defined from the edges of its neighborhood faces.

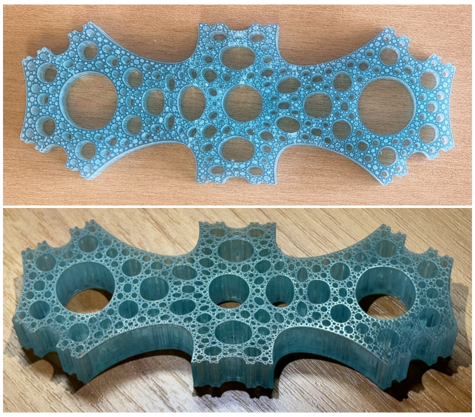


Figure 27: 3D print of the fractal structure presented in Figure 13 and 14.

We complete this approach by introducing a delay in the subdivision process, allowing the cohabitation in an assembly of different iteration levels of an identical lacunar fractal. We use this extension to produce a homogeneous lacunarity over faces with different sizes or to manage variation lacunarity. We can extend this approach by proposing additional subdivision algorithms and introducing more than one main lacunar, for example.

We analyze the lacunarity of the resulting structures by evaluating and proposing different measures for geometric and topologic characteristics.

Future works focus on an automatic process to subdivide volumes. It is more complicated than faces since the adjacency of two volumes is on faces, with more topology possibilities than edges. As for faces, we can subdivide each volume edge (faces) and create subvolumes for each sub-face. However, there could be several adjacency faces and lacuna faces for each subvolume. The complexity induced by the topological cellular decomposition of 3D fractal structures needs deep analysis to identify, parametrize, and automatize the subdivision process while guaranteeing topological consistency.

7 REFERENCES

[AALS17] Niels Aage, Erik Andreassen, Boyan S Lazarov, and Ole Sigmund. Giga-voxel computational morphogenesis for structural design. *Nature*, 550(7674):84–86, 2017.

[AC15] Lawrence M. Anovitz and David R. Cole. Characterization and Analysis of Porosity and Pore Structures. *Reviews in Mineralogy and Geochemistry*, 80(1):61–164, 2015.

[AJS18] Diab W. Abueidda, Iwona Jasiuk, and Nahil A. Sobh. Acoustic band gaps and elastic stiffness of pmma cellular solids based on triply periodic minimal surfaces. *Materials & Design*, 145:20–27, 2018.

[AKA21] Oraib Al-Ketan and Rashid K. Abu Al-Rub. MSLattice: A free software for generating uniform and graded lattices based on triply periodic minimal surfaces. *Material Design and Processing Communications*, 3(6):1–10, 2021.

[Ban06] J. Banhart. Metal Foams: Production and Stability. *Advanced Engineering Materials*, 8(9):781–794, September 2006.

[Bar90] Anthony Barcellos. Fractals everywhere. by michael barnsley. *The American Mathematical Monthly*, 97(3):266–268, 1990.

[BEH89] Michael F. Barnsley, John H. Elton, and Douglas P. Hardin. Recurrent Iterated Function Systems. *Constructive Approximation*, pages 3–31, 1989.

[BRBH19] Brent R Bielefeldt, Gregory W Reich, Philip S Beran, and Darren J Hartl. Development and validation of a genetic l-system programming framework for topology optimization of multifunctional structures. *Computers & Structures*, 218:152–169, 2019.

[CLLZ21] Liang-Yu Chen, Shun-Xing Liang, Yujing Liu, and Lai-Chang Zhang. Additive manufacturing of metallic lattice structures: Unconstrained design, accurate fabrication, fascinated performances, and challenges. *Materials Science and Engineering: R: Reports*, 146:100648, October 2021.

[Esp12] Laura Espinal. Porosity and Its Measurement. In *Characterization of Materials*, pages 1–10. John Wiley & Sons, Ltd, 2012. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/0471266965

[FFL+18] Jiawei Feng, Jianzhong Fu, Zhiwei Lin, Ce Shang, and Bin Li. A review of the design methods of complex topology structures for 3d printing. *Visual Computing for Industry, Biomedicine, and Art*, 1(1):1–16, 2018.

[GGS21] Christian Gentil, Gilles Gouaty, and Dmitry Sokolov. *Geometric Modeling of Fractal Forms for CAD*. John Wiley & Sons, 2021.

[Hut81] John E Hutchinson. Fractals and self similarity. *Indiana University Mathematics Journal*, 30(5):713–747, 1981.

[JWZ07] Bin Jiang, Zejun Wang, and Naiqin Zhao. Effect of pore size and relative density on the mechanical properties of open cell aluminum foams. *Scripta Materialia*, 56(2):169–172, January 2007.

[JZSL05] B Jiang, N Zhao, C Shi, and J Li. Processing of open cell aluminum foams with tailored porous morphology. *Scripta Materialia*, 53(6):781–785, September 2005.

[KT10] X.Y. Kou and S.T. Tan. A simple and effective geometric representation for irregular porous structure modeling. *Computer-Aided Design*, 42(10):930–941, October 2010.

[LC19] Yuan Liang and Gengdong Cheng. Topology optimization via sequential integer programming and canonical relaxation algorithm. *Computer Methods in Applied Mechanics and Engineering*, 348:64–96, 2019.

[LJP+19] Y. Li, H. Jahr, P. Pavanram, F.S.L. Bobbert, U. Puggi, X.-Y. Zhang, B. Pouran, M.A. Leeflang, H. Weinans, J. Zhou, and A.A. Zadpoor. Additively manufactured functionally graded biodegradable porous iron. *Acta Biomaterialia*, 96:646–661, September 2019.

[LVSZ21] Nikita Letov, Pavan Velivela, Siyuan Sun, and Yaoyao Zhao. Challenges and Opportunities in Geometric Modelling of Complex Bio-Inspired 3D Objects Designed for Additive Manufacturing. *Amer-*

ican Society of Mechanical Engineers, page 10, 2021.

[Man85] Benoit B Mandelbrot. Self-affine fractals and fractal dimension. *Physica scripta*, 32(4):257, 1985.

[MBZ+20] T. McNulty, D. Bhate, A. Zhang, M. A. Kiser, L. Ferry, A. Suder, S. Bhattacharya, and P. Boradkar. A framework for the design of biomimetic cellular materials for additive manufacturing. *Solid Freeform Fabrication 2017: Proceedings of the 28th Annual International Solid Freeform Fabrication Symposium - An Additive Manufacturing Conference, SFF 2017*, (2):2188–2200, 2020.

[MGM+18] Mehrdad Mohsenizadeh, Federico Gasbarri, Michael Munther, Ali Beheshti, and Keivan Davami. Additively-manufactured lightweight metamaterials for energy absorption. *Materials & Design*, 139:521–530, 2018.

[Pen10] Deborah Pence. The simplicity of fractal-like flow networks for effective heat and mass transport. *Experimental Thermal and Fluid Science*, 34(4):474–486, 2010.

[PHL20] Chen Pan, Yafeng Han, and Jiping Lu. Design and optimization of lattice structures: A review. *Applied Sciences*, 10(18), 2020.

[SGGM15] Dmitry Sokolov, Gilles Gouaty, Christian Gentil, and Anton Mishkinis. Boundary controlled iterated function systems. In *Curves and Surfaces: 8th International Conference, Paris, France, June 12-18, 2014, Revised Selected Papers 8*, pages 414–432. Springer, 2015.

[SHR97] Bernard Sapoval, Olivier Haeberlé, and Stephanie Russ. Acoustical properties of irregular and fractal cavities. *The Journal of the Acoustical Society of America*, 102(4):2014–2019, 1997.

[TGM+09] Olivier Terraz, Guillaume Guimberteau, Stéphane Mérillou, Dimitri Plemenos, and Djamchid Ghazanfarpour. 3Gmap L-systems: An application to the modelling of wood. *Visual Computer*, 25(2):165–180, 2009.

[TMV+16] Mary Kathryn Thompson, Giovanni Moroni, Tom Vaneker, Georges Fadel, R Ian Campbell, Ian Gibson, Alain Bernard, Joachim Schulz, Patricia Graf, Bhrigu Ahuja, et al. Design for additive manufacturing: Trends, opportunities, considerations, and constraints. *CIRP annals*, 65(2):737–760, 2016.

[Zha11] Li-Zhi Zhang. Heat and mass transfer in a randomly packed hollow fiber membrane module: a fractal model approach. *International Journal of heat and mass transfer*, 54(13-14):2921–2931, 2011.

[ZT96] Chems Eddine Zair and Eric Tosan. Fractal modeling using free form techniques. *Computer Graphics Forum*, 15(3):269–278, 1996.

[ZZ13] Fuzong Zhou and Xinrong Zhang. Assessment of heat transfer in an aquifer utilizing fractal theory. *Applied thermal engineering*, 59(1-2):445–453, 2013.