EQUATIONS AND INTERVAL COMPUTATIONS FOR SOME FRACTALS

Lincong Fang, Dominique Michelucci, Sebti Foufou

EQUATIONS AND INTERVAL COMPUTATIONS FOR SOME FRACTALS

LINCONG FANG,[*,§] DOMINIQUE MICHELUCCI[†] and SEBTI FOUFOU[†,‡]
[*]*School of Information*
Zhejiang University of Finance & Economics
Hangzhou 310018, P. R. China
[†]*LE2I UMR6306, CNRS*
Arts et Métiers, Université Bourgogne Franche-Comté
F-21000 Dijon, France
[‡]*New York University of Abu Dhabi*
Computer Science, P. O. Box 129188, Abu Dhabi, UAE
[§]*lincongfang@gmail.com*

Received May 11, 2017
Accepted April 16, 2018
Published August 13, 2018

Abstract

Very few characteristic functions, or equations, are reported so far for fractals. Such functions, called Rvachev functions in function-based modeling, are zero on the boundary, negative for inside points and positive for outside points. This paper proposes Rvachev functions for some classical fractals. These functions are convergent series, which are bounded with interval arithmetic and interval analysis in finite time. This permits to extend the Recursive Space Subdivision (RSS) method, which is classical in Computer Graphics (CG) and Interval Analysis, to fractal geometric sets. The newly proposed fractal functions can also be composed with classical Rvachev functions today routinely used in Constructive Solid Geometry (CSG) trees of CG or function-based modeling.

Keywords: Function-Based Modeling; Function Representation; Fractal; Interval Arithmetic.

[§]Corresponding author.

1. INTRODUCTION

Function-based modeling[1–4] is well studied in Computer-Aided Design (CAD) and Computer Graphics (CG). It is a generalization of the Constructive Solid Geometry (CSG) representation in which not only primitives, but also composed objects (e.g. with set operations) are represented with characteristic functions. A characteristic function has value zero for points on the boundary, and takes values of opposite signs for inside and outside points. In this paper, we use the convention that the function is negative inside and positive outside the geometric set.

Algebraic objects such as spheres, quadrics, and tori, are represented with polynomial functions (e.g. $f(x,y,z) = (x-c_x)^2 + (y-c_y)^2 + (z-c_z)^2 - r^2 \leq 0$ for a ball of known radius r centered at a given point (c_x, c_y, c_z)) or with signed distance functions[5] (e.g. for the polygon). The generalization to other smooth functions such as trigonometric functions or exponential is also possible.

An arbitrary complex solid is defined using a single continuous real-valued function, the zero set of which is the boundary of the object (it is called a surface). For example, let $\mathbf{A}$ and $\mathbf{B}$ be two objects in $\mathbb{R}^d$, $d > 0$, represented with functions $f_{\mathbf{A}}$ and $f_{\mathbf{B}}$. The simplest approach to formalize set operation between $\mathbf{A}$ and $\mathbf{B}$ is to represent $\mathbf{A} \cap \mathbf{B}$ with the function $f_{\mathbf{A}\cap\mathbf{B}}(p) = \max(f_{\mathbf{A}}(p), f_{\mathbf{B}}(p))$, $p \in \mathbb{R}^d$, and $\mathbf{A} \cup \mathbf{B}$ with the function $f_{\mathbf{A}\cup\mathbf{B}}(p) = \min(f_{\mathbf{A}}(p), f_{\mathbf{B}}(p))$. The complement of the object $\mathbf{A}$ is represented with the function $f_{\neg\mathbf{A}}(p) = -f_{\mathbf{A}}(p)$. For affine invertible transforms T (translations, rotations, scalings), the function $f_{T(\mathbf{A})}$ for $T(\mathbf{A})$ is $f_{T(\mathbf{A})}(p) = f_{\mathbf{A}}(T^{-1}(p))$. Rvachev proposed other functions, now known as R-functions, or Rvachev functions, to realize set operations between solids.[1,6]

Functions representing objects must be computable, i.e. they can be approximated within a prescribed accuracy.[7] Only continuous functions are computable,[7] in the sense of "computable numbers" or computable analysis. A real number is computable if by definition it is possible to produce a set of nested intervals converging to it. A function f is computable if for all computable numbers x, $f(x)$ is computable. Therefore, only continuous functions are computable. The equality of two computable numbers is nondecidable. The nonequality of two computable numbers is decidable (compute two thin enough intervals for each, eventually disjoint). For example, the sign function, -1 for negative, 0 for 0, 1 for positive is noncomputable, more precisely the computation does not halt for 0.

An advantage of the function-based modeling is that the boundary is extracted from an approximation or a cover, computed with a prescribed accuracy threshold. The topology of this approximation may not be equal to the topology of the exact object, which is anyway noncomputable (since, for example, the number of connected components is not a continuous function; moreover for fractals, it may be infinite).

Interval arithmetic and analysis provide methods to compute a guaranteed and sharp enclosure of the values of smooth (i.e. differentiable) functions over an interval, or a box (a box is a vector of intervals).[2,8,9] In Sec. 2.2, this feature is used by the classical Recursive Space Subdivision (RSS) method.

Fractals are widely used in geometric modeling to represent natural shapes[10,11] or artifacts, e.g. the antenna of some cellular phones is a two-dimensional (2D) Menger sponge. Finding equations for fractals could bridge geometric modeling and fractals and therefore ease fractals' integration in geometric modelers. Moreover, it might provide a feasible way for three-dimensional (3D) printing of fractal objects. However, for fractal objects, these equations or Rvachev functions are not differentiable, and sometimes even not continuous.

It is possible to extend interval computations to nonsmooth functions, i.e. functions which are nowhere or almost nowhere differentiable, and to noncontinuous functions. We call them fractal functions, because they appear in the function-based shape modeling of fractal objects.

In this paper, we propose functions, actually convergent series, of some classical fractal objects.[10–12] These functions are differentiable nowhere or almost nowhere, though they are usually continuous in some sense (Lipschitz,[13] Hölder,[14] etc.). These series can be enclosed within intervals,[2,7,8] sometimes tightly, in finite time. These proposed fractal functions will give another way to account for fractals into the function-based modeling framework, and extend the scope of the RSS method, which may also generate new interesting and challenging geometric problems.

We summarize the contributions of this paper as follows: it gives equations (i.e. a characteristic function which is negative for inside points and positive for outside points) for some classical fractals; it shows that exactly one contractant mapping is sufficient in the Iterated Function System (IFS). These functions can be bounded with intervals computations. They can be composed with other classical Rvachev functions.[1,15]

The rest of the paper is organized as follows: we first introduce the background in Sec. 2. Section 3 gives some examples of fractal functions, which define fractal shapes, and can be enclosed with interval arithmetics. The fractal functions permit to cover the fractal object or its boundary with the RSS method. Arising questions are discussed in Sec. 4.

2. BACKGROUND

This section presents methods for computing a cover of a fractal inside a box. We recall the classical representations of fractals in Sec. 2.1. Implicit equations of classical fractals are rarely reported, though the parametric equation of some fractal curves is known.[16] Methods for tracing strange attractors like Julia sets or Hénon attractor were proposed.[17–20] Methods were presented for ray-casting surfaces defined by fractal combinations of procedural noise functions.[21–23]

Section 2.2 presents the classical RSS method, which can be used when a Rvachev function is available and is computable with interval analysis. This method also applies for objects defined with CSG trees.

In Sec. 2.3, the fractal is defined by an IFS and no Rvachev function is available. This section presents a branch and bound method for computing the distance of a given point to a fractal defined by an IFS: the latter can be used to compute a cover of the fractal.

In Sec. 2.4, we consider the case when no Rvachev function is available and the fractal is the attractor of orbits which does not diverge to infinity, for some given computable map f. Examples are Julia sets, and the Hénon strange attractor.[18] This section presents a graph-based method for computing a cover of the fractal.

The conclusion to be learned from this work is that it is simpler and easier to compute a cover of a fractal object when a Rvachev function is available.

Table 1 Evaluation Cost of Approximations of R-Functions[24] Relatively to k, the Recursion Depth.

Fractal	Complexity
Sierpiński's napkin	$O(k)$
Sierpiński's carpet	$O(k)$
Levy fractal	$O(2^k)$
Koch curve	$O(4^k)$
Pythagoras fractal	$O(5^k)$

2.1. Classical Representations of Fractals

Fractals can be approximated with a union of simple geometric primitives (triangles in the example of the Sierpiński triangle), or with a CSG tree, but with a small recursion depth.[24] Traversing this CSG tree straightforwardly produces a Rvachev function. But, the evaluation cost of this naive Rvachev function increases, sometimes exponentially, with the recursion depth, which limits the depth to a small constant. Such Rvachev functions for some classical fractals are proposed in Ref. 24, their evaluation cost is given in Table 1. Though this approach is sufficient for visualization purposes, it does not solve the challenging problem tackled here. In this paper, we will define Rvachev functions with constant size but with infinite recursive depth, i.e. convergent series, and we will use interval analysis to bound them in finite time.

Some fractals are defined as the attractor of an IFS. Rice[25] proposes a method to compute a bounding box of this kind of fractals. Barnsley[10] proposes the chaos game to sample them. We are not aware of any method to compute a Rvachev function of a fractal defined by an IFS.

Some fractals are defined as attractors of nondivergent orbits, for a given function, like the Hénon strange attractor or Julia sets (or the related Fatou sets). Michelucci *et al.*[19] and Paiva *et al.*[20] proposed the graph-based methods to compute a cover of such fractals. This method is presented in Sec. 2.4.

Some fractals have a combinatorial definition, with a finite automata (for instance, the von Koch curve or the dragon curve), or with exotic numeration systems.[16,26,27]

A Rvachev function is advantageous because it avoids to design special algorithms to deal with fractals: the RSS algorithm can manage them, and they can be combined easily with other Rvachev shapes

(a shape is Rvachev if its Rvachev function is available). Examples of Rvachev functions for some classical fractals are given in the following sections.

2.2. The RSS Method and Rvachev Function

This section recalls the classical method[28] used to compute a cover of a geometric set $\mathbf{A}$, inside a given box $\mathbf{B}$, when a Rvachev function $f_{\mathbf{A}}$ is known, and is computable with interval analysis. In 2D, to cover the object inside a box $\mathbf{B} = (X, Y)$, where X and Y are two intervals, the method computes an interval $Z = [u, v]$ enclosing $f(X, Y)$, with some interval computation.[28] If v is negative, then the box (X, Y) is completely inside the object: it is added to the cover. If u is positive, then the box (X, Y) is completely outside. Otherwise, $u \leq 0 \leq v$, and nothing can be said because the interval $[u, v]$ may overestimate the exact range. If the box is small enough, so that no more subdivision can be done, the conservative choice is to insert the box in the cover. Otherwise, the box is subdivided, and the sub-boxes are recursively evaluated. This recursive subdivision of 2D space is a classical method used to tessellate objects, perform volume computations, etc. within the framework of function-based modeling. For instance, the marching cube method computes the boundary of an object at the voxel level. Fryazinov *et al.*[3] and Martin *et al.*[28] compared several interval computations for this algorithm. They considered only differentiable polynomial functions.

2.3. Branch and Bound Method and IFS

In this section, the fractal $\mathbf{F}$ is defined as the attractor of an IFS and no Rvachev function is available.

The branch and bound method makes it possible to use the RSS method in this case. The branch and bound method computes intervals enclosing the unsigned (nonnegative) distance from a given point p to $\mathbf{F}$. Typically, the interval width decreases at each iteration of the algorithm, which stops when a tight enough interval is reached. This method can be used to compute the (unsigned) distance from a box $\mathbf{B}$ with center p to $\mathbf{F}$. Let $r = [r^-, r^+]$ be the radius of the smallest ball $\mathcal{B}(p, r)$ enclosing the box $\mathbf{B}$. The value r is represented with an interval to account for inaccuracy (for example when $r = \sqrt{2}$). Let $d = [d^-, d^+]$ be an interval enclosing the unsigned distance from p to $\mathbf{F}$: d is computed with the branch and bound algorithm. Then an interval enclosing the distance from $\mathbf{B}$ to $\mathbf{F}$ is

$$d(\mathbf{B}, \mathbf{F}) \in [\max(0, d^- - r^+), d^+ + r^+]. \tag{1}$$

We now explain the branch and bound method for computing an interval of the distance from a given point p to the fractal $\mathbf{F}$. $\mathbf{F}$ is the attractor of a given IFS $\mathcal{F} = \{f_1, \ldots, f_n\}$. The maps $f_1, \ldots, f_n$, are contractant, usually affine, transformations. Let $\rho \in (0, 1)$ be the contracting factor of $\mathcal{F}$, thus

$$\forall\, i \in [1, n], \forall\, x, \forall\, y, \quad \|f_i(x) - f_i(y)\| \leq \rho \|x - y\|.$$

The fractal $\mathbf{F}$ is nonempty: It contains at least the fixed points of the transformations f_i.

We recall the definition of Hutchinson's operator:

$$H : \mathbf{X} \to H(\mathbf{X}) = \bigcup_{i=1}^{n} f_i(\mathbf{X}),$$

where $\mathbf{X}$ is any compact set. The fixed point of H is the attractor of the IFS $\mathcal{F}$, i.e. the fractal $\mathbf{F}$. Computationally, convenient sets X are unions of balls, possibly overlapping: the image of a ball $\mathcal{B}(c, r)$ by a map $f_i \in \mathcal{F}$ is included in the ball $\mathcal{B}(f_i(c), r \times \rho)$, the center of which is $f_i(c)$, and the radius of which is $r \times \rho$. We assume a ball $\mathcal{B}(C, R)$ enclosing $\mathbf{F}$ is known: C is its center, and R its radius.[25,29]

For convenience, we redefine H as follows: for a ball $\mathcal{B}(c, r)$,

$$H(\mathcal{B}(c, r)) := \bigcup_{i=1}^{n} \mathcal{B}(f_i(c), r \times \rho),$$

and for a union of balls, the image by H is the union of the images of balls by H. The underlying approximation in this redefinition is conservative. Define $U^0 := \{\mathcal{B}(C, R)\}$ and $U^{k+1} := H(U^k)$. Each U^k is a union of n^k balls, all with radii $R \times \rho^k$. Each U^k is a cover of the fractal $\mathbf{F}$. Moreover, this cover is fair: $U^k \cap \mathbf{F}$ is never empty. As point sets, $U^{k+1} \subset U^k$, and $\mathbf{F} = \lim_{k \to \infty} U^k$.

A virtual tree of balls can be associated in a natural way to the set of balls in U^k, $k \in \mathbb{N}$: the root is the bounding ball $\mathcal{B}(C, R)$ in U^0, and each ball $\mathcal{B}(c, r)$ has n sons $\mathcal{B}(f_i(c), \rho \times r)$. U^k is the union of balls at depth k in the tree.

An ϵ-cover of $\mathbf{F}$ is a cover by a set of balls the radii of which are all smaller or equal to ϵ. The Hausdorff distance between $\mathbf{F}$ and an ϵ-cover of $\mathbf{F}$

is at most ϵ. Let k be the integer:

$$R \times \rho^k \leq \epsilon \Rightarrow k = \left\lceil \frac{\log(\epsilon/R)}{\log \rho} \right\rceil, \qquad (2)$$

so U^k is an ϵ-cover of $\mathbf{F}$. The distance to $\mathbf{F}$ of a given point p is (up to ϵ) the distance of p to the union of balls U^k.

This gives a method to compute a fair ϵ-cover U^k.

This also suggests a first, naive and brute force method to compute (an interval of) the distance of p to $\mathbf{F}$: generate all n^k balls of U^k and compute which one is the closest to p. It is clearly exponential time $O(n^k)$. The branch and bound principle[8,17] speeds up this first naive method. Suppose some greedy method finds a ball in the ϵ-cover, thus with radius smaller than ϵ, and the distance of this ball to p is at most η. Then, it is useless to compute $H(\mathcal{B}(c,r)), H(H(\mathcal{B}(c,r)))$, etc., i.e. the subtree of $\mathcal{B}(c,r)$, as soon as the distance from p to $\mathcal{B}(c,r)$ is greater than η: The ball $\mathcal{B}(c,r)$ is too far, and its sons in the tree are even further away.

Regarding the greedy method, for a given ball $\mathcal{B}$, it only considers the closest ball to p amongst $f_1(\mathcal{B}), \ldots, f_n(\mathcal{B})$. It may give a nonoptimal result, but it quickly provides an upper bound of the distance.

In practice, the branch and bound method[8,17] is much faster than the naive method computing U^k with k defined in Eq. (2). However, it remains slower than the interval analysis method presented in Sec. 2.2. But, to apply the latter, an equation of the fractal is needed.

2.4. Attractors of Nondivergent Orbits

In this section, no IFS and no Rvachev function is available. Nonetheless, interval analysis permits to compute in a reliable way covers of fractals like the Julia sets or the Hénon attractor[18–20] which are attractors of orbits $\mathcal{O}(p) = \{p, f(p), f(f(p)), \ldots, \}$ that are not diverging. For example, for Hénon attractor, the function f is defined as

$$f : (x,y) \in \mathbb{R}^2 \rightarrow f(x,y) = (1 + y - ax^2, bx) \in \mathbb{R}^2,$$

where a and b are parameters, $a = 1.4$ and $b = 0.3$ for the famous Hénon strange attractor. For Julia sets, $f(z) = z^2 + c$, where $c \in \mathbb{C}$ is a parameter.

The method operates as follows: assume, for simplicity, that a bounding box of the attractor is known. This bounding box is partitioned into a set $\mathcal{C}$ of square or rectangular cells $\mathbf{C}_i$, with $i = 1, \ldots, n^2$. Each cell $\mathbf{C}$ in $\mathcal{C}$ is associated to a vertex c in a graph G, and each time there is a point $p \in \mathbf{C}$ such that $f(p) \in \mathbf{C'}$, i.e. interval analysis detects that $f(\mathbf{C}) \cap \mathbf{C'}$ is not empty, an arc $c \rightarrow c'$ is added to the graph G. Then, the strongly connected components of the graph G are computed, with Tarjan's method.[30] So, each vertex c is associated to its strongly connected component $\mathrm{scc}(c)$. A vertex c in G, corresponding to a cell $\mathbf{C} \in \mathcal{C}$, is transient if there is no arc $c \rightarrow c$ in the graph G, and c belongs to a strongly connected component which contains only the vertex c itself, i.e. $\mathrm{scc}(c) = \{c\}$. In other words, if c is transient, no orbit for f starting in the cell $\mathbf{C}$ can return to $\mathbf{C}$. Thus, transient cells contain no point of the attractor. Nontransient cells cover the attractor. In comparison, the orbit method often gives wrong results[18–20] typically near repulsive points of the fractal, which cause "leaks".

An interesting feature of this method is that it can be used iteratively: Let $\mathbf{C}$ be the first set of cells, and let $\mathbf{C}_1$ be the set of nontransient cells. Then, cells in $\mathbf{C}_1$ can be subdivided again, and the same algorithm is reexecuted, to obtain a thinner cover $\mathbf{C}_2$, etc. Some (2, 3, or 4) iterations yield an exact cover much thinner than the pictures produced with the orbit method. This method can be modified to compute periodic points in the attractor.[20] It should also be used with IFS: an arc links c to c' if for some function f_k in the IFS, there is a point $p \in \mathbf{C}$ such that $f_k(p) \in \mathbf{C'}$, i.e. $\mathbf{C} \cap f_k^{-1}(\mathbf{C'})$ is nonempty, or equivalently $f_k(\mathbf{C}) \cap \mathbf{C'}$ is nonempty.

Once a cover of the fractal is known, it may seem easy to estimate the distance of a given point p to the fractal, using the cover instead, represented with some quadtree or octree data structure. However, the cover is exact (no part of the attractor is outside the cover, which is usually tight), but it may happen that some cells in the cover set are actually empty, i.e. that they contain no point of the fractal (remark that this problem cannot occur with the branch and bound method (Sec. 2.3)). A possibility is to make the cover fair: for each cell, compute a periodic point inside it,[20] or prove that it is empty.

3. COMPUTING ENCLOSURES OF FRACTAL FUNCTIONS

This section presents functions for some fractals and the methods to compute their enclosures. To define fractal functions, the most basic function is

the distance to the integers. Let $x \in \mathbb{R}$, the distance function is defined as

$$d_{\mathbb{Z}}(x) = \min(x - \lfloor x \rfloor, \lceil x \rceil - x), \tag{3}$$

where $\lfloor x \rfloor$ is the floor of x, and $\lceil x \rceil$ is the ceiling of x. The function $d_{\mathbb{Z}}$ has period 1; it is piecewise linear, continuous, bounded, i.e. $d_{\mathbb{Z}}(\mathbb{R}) = [0, 1/2]$. The interval enclosing $d_{\mathbb{Z}}(X)$ for a given interval $X = [x_0, x_1]$ is easily computed. Furthermore, the odd function $d_{\mathbb{Z}_1}$ and the even function $d_{\mathbb{Z}_0}$ can be defined as follows:

$$d_{\mathbb{Z}_1}(x) = 2d_{\mathbb{Z}}\left(\frac{x}{2} + \frac{1}{4}\right) - \frac{1}{2}, \tag{4}$$

$$d_{\mathbb{Z}_0}(x) = -d_{\mathbb{Z}_1}(x). \tag{5}$$

It is obvious that $d_{\mathbb{Z}_1}(\mathbb{R}) \in [-\frac{1}{2}, \frac{1}{2}]$ and $d_{\mathbb{Z}_0}(\mathbb{R}) \in [-\frac{1}{2}, \frac{1}{2}]$. Figure 1 shows the graph of the functions $d_{\mathbb{Z}}$, $d_{\mathbb{Z}_1}$, and $d_{\mathbb{Z}_0}$.

3.1. The Takagi Function

Let

$$T_k(x) = \sum_{i=0}^{k} \frac{d_{\mathbb{Z}}(2^i x)}{2^i}, \tag{6}$$

then $T_\infty(x)$ is the Takagi function (as usual, $T_\infty(x) = \lim_{k\to\infty} T_k(x)$). Since $d_{\mathbb{Z}}$ is bounded, the series converges. The Takagi function has period 1; it is continuous but not differentiable. It is possible to compute an interval enclosing $T_\infty(X)$ for a given interval $X = [x_0, x_1]$ as follows:

$$\begin{aligned} T_\infty([x_0, x_1]) &= \sum_{i=0}^{n} \frac{d_{\mathbb{Z}}(2^i[x_0, x_1])}{2^i} \\ &\quad + \sum_{i=n+1}^{\infty} \frac{d_{\mathbb{Z}}(2^i[x_0, x_1])}{2^i} \\ &\in \sum_{i=0}^{n} \frac{d_{\mathbb{Z}}(2^i[x_0, x_1])}{2^i} \\ &\quad + \frac{[0, \frac{1}{2}]}{2^{n+1}} \left(1 + \frac{1}{2} + \cdots\right) \\ &\in \sum_{i=0}^{n} \frac{d_{\mathbb{Z}}(2^i[x_0, x_1])}{2^i} + \left[0, \frac{1}{2^{n+1}}\right]. \end{aligned}$$

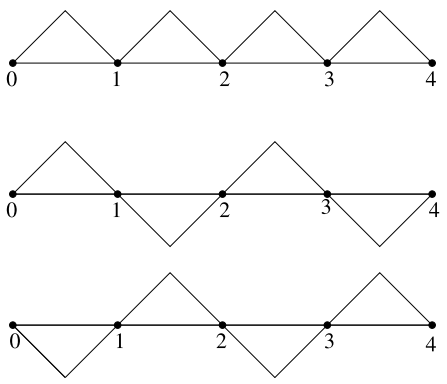

Fig. 1 From top to bottom: the $d_{\mathbb{Z}}$ function is the distance to integers, the odd function $d_{\mathbb{Z}_1}$ (x is inside if $\lfloor x \rfloor$ is odd), the even function $d_{\mathbb{Z}_0}$.

The first part of the right-hand side can be computed with intervals, or exactly with rational arithmetic when x_0, x_1 are rational.

Figure 2 shows the fractal $\{(x, y) \,|\, y = T_\infty(x)\}$, for $x \in [0, 1], y \in [0, 1]$, the recursive subdivision process of the outward rounded interval arithmetic is shown on the left.

Notably, the Takagi function is Hölder continuous, that is

$$|T(x_2) - T(x_1)| \leq \frac{2^{1-\alpha}}{1 - 2^{\alpha-1}} |x_2 - x_1|^\alpha.$$

We computed the value of the function T at the center of the considered interval X, and then get the enclosure of $T(X)$. This method gives the same subdivision as the naive interval arithmetic in our implementation.

3.2. The Devil's Staircase

The devil's staircase can be defined[31] as $E_\infty(x)$, where

$$E_k(x) = \sum_{i=1}^{k} \frac{\lfloor ix \rfloor}{2^i}. \tag{7}$$

The floor function is not continuous, but it can be computed with interval arithmetic (it is possible to enclose the image of an interval by the floor function) easily, because it is increasing, an enclosure of $E_\infty([x_0, x_1])$ is $[E_\infty(x_0), E_\infty(x_1)]$. Obviously, the series converges.

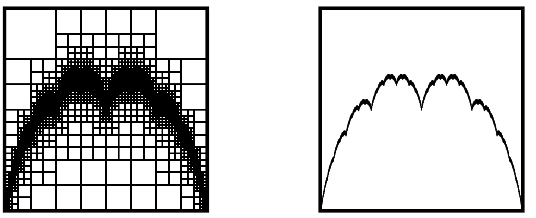

Fig. 2 The Takagi curve, also called the Blancmange curve. Left: the recursive subdivision process. Right: the cover obtained with 12 levels of recursion.

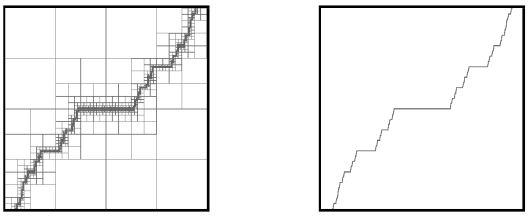

Fig. 3 The fractal object $\{(x, y) \mid y = E_\infty(x)\}$, where E_∞ is the devil's staircase function. Left: the recursive subdivision process using interval evaluation of the function. Right: the resulting cover.

The devil's staircase function is an example of a function that is continuous, but not absolutely continuous. It is also called Cantor's function: it is constant for values outside the Cantor set (or Cantor dust). It is not Lipschitz continuous. See Fig. 3 for a graph of the function, traced with the recursive subdivision method and outward rounded interval arithmetic.

3.3. Distance to Cantor Dust

This section presents a function to get the signed distance to the covers of the one-dimensional (1D) Cantor dust (or Cantor set). The signed distance is positive for a point outside and negative for a point inside the cover.

The Cantor set is created by repeatedly removing the open middle third of a set of line segments. Let $\mathbf{C}_k$ be the obtained set of the kth step in this process, then the cover of $\mathbf{C}_0$ is the segment, or interval $[0, 1]$. The cover $\mathbf{C}_k$ is the union of $\frac{1}{3}\mathbf{C}_{k-1}$ and of $\frac{2}{3} + \frac{1}{3}\mathbf{C}_{k-1}$. The limit of $\mathbf{C}_k$ when $k \to \infty$ is the famous Cantor set.

To define the signed distance function to the Cantor set, we pose $f(x) = d_{\mathbb{Z}}(x) - \frac{1}{3}$ for convenience. f is bounded, i.e. $f(x) \in [-\frac{1}{6}, \frac{1}{3}]$. Then the signed distance to the cover $\mathbf{C}_k$ is defined as

$$C_k(x) = \max_{i=0}^{k} \frac{f(3^i x)}{3^i},$$

which is indeed convergent when $k \to \infty$, and $C_\infty(x)$ is the signed distance function to the Cantor set.

The RSS method is applied to the set defined by the equation $y - C_k(x) = 0$. It is essential that f is bounded, so the series converges, and it is possible to lower and upper bound the error at each step. Figure 4 shows the graph of these functions for $C_k(x), k = 0, \ldots, 5$ for $x \in [0, 1]$, $y \in [-0.5, 0.5]$. Figure 5 shows the graph of the function $C_\infty(x)$.

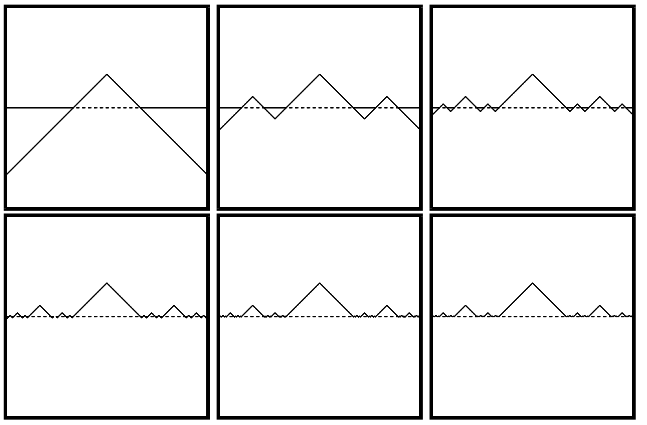

Fig. 4 The graph of the cover of the curve $\{(x, y) \mid y - C_k(x) = 0\}, k = 0, \ldots, 5, x \in [0, 1]$, and $y \in [-0.5, 0.5]$.

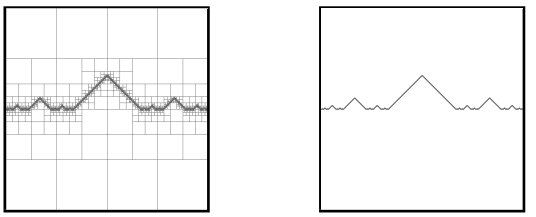

Fig. 5 The graph of the signed distance to the Cantor dust $\mathbf{C}_\infty$. Left: the recursive subdivision algorithm. Right: the resulting cover.

3.4. Sierpiński's Gasket

Sierpiński's gasket is another icon of fractals, see Fig. 6. It has various relatives,[32] and its 3D variant is also called Menger's sponge. Define a series:

$$S_k(x, y) = \min_{i=0}^{k} \frac{f(3^i x, 3^i y)}{3^i}, \tag{8}$$

where $f(x, y) = \frac{1}{3} - \min(d_{\mathbb{Z}}(x), d_{\mathbb{Z}}(y))$. The signed distance to the union of holes in Sierpiński's gasket is $S_\infty(x, y)$. It is the generalized Rvachev function of the Sierpiński gasket. Since f is bounded, the series clearly converges. It is then possible to compute with intervals an enclosure of the signed distance to the Sierpiński's gasket for any point and any box.

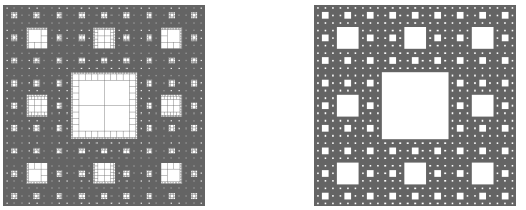

Fig. 6 Sierpiński's gasket (sometimes called Menger's sponge). Left: the recursive subdivision method with outward rounded interval arithmetic. Right: the corresponding cover.

Figure 6 shows the result of the RSS method with outward-rounded interval computations.

3.5. Sierpiński's Triangle

Figure 7 shows the Sierpiński's triangle. Let $d_m(x) = x \mod 1$ and $f(x, y)$ be the signed distance to the tiling of triangles $\{(x, y) \,|\, d_m(y) \leq d_m(x)\}$. The function $f(x, y) = d_m(y) - d_m(x)$ can be used, and it was used, to produce our first figures, e.g. see Fig. 7, although it uses the noncontinuous function, i.e. modulo 1. Once a function f is available, the function for the signed distance to the holes in the Sierpiński's triangle is $S_\infty(x, y)$, where

$$S_k(x, y) = \max_{i=0}^{k} \frac{f(2^i x, 2^i y)}{2^i}. \tag{9}$$

A definition for the function f which does not depend on a discontinuous function (like the modulo 1 function) can be found. As shown in Fig. 8, first define functions for stripes, for example, the odd function (4) is zero for integers x, negative in the open intervals $(1, 2), (3, 4), \ldots, (2n + 1, 2n + 2)$, and positive in the open intervals $(2, 3)$, $(4, 5), \ldots, (2n, 2n+1)$. In other words, $d_{\mathbb{Z}_1}(x)$ is the signed distance functions to stripes $2\mathbb{Z} + [1, 2]$. The function for the complement is the even function (5). Then, define vertical stripes $\mathbf{A}$ and $\mathbf{A}'$, horizontal stripes $\mathbf{B}$ and $\mathbf{B}'$, diagonal stripes $\mathbf{C}$ and $\mathbf{C}'$ (e.g. considering $d_{\mathbb{Z}_0}(x + y)$, $d_{\mathbb{Z}_1}(x + y)$). Then, the tiling of triangles can be obtained as $(\mathbf{A} \cap \mathbf{B} \cap \mathbf{C}') \cup (\mathbf{A} \cap \mathbf{B}' \cap \mathbf{C}) \cup (\mathbf{A}' \cap \mathbf{B} \cap \mathbf{C}) \cup (\mathbf{A}' \cap \mathbf{B}' \cap \mathbf{C}')$, see Fig. 8.

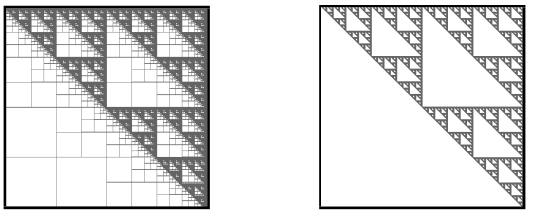

Fig. 7 Sierpiński's triangle. Left: the recursive subdivision method with outward-rounded interval arithmetic. Right: the corresponding cover.

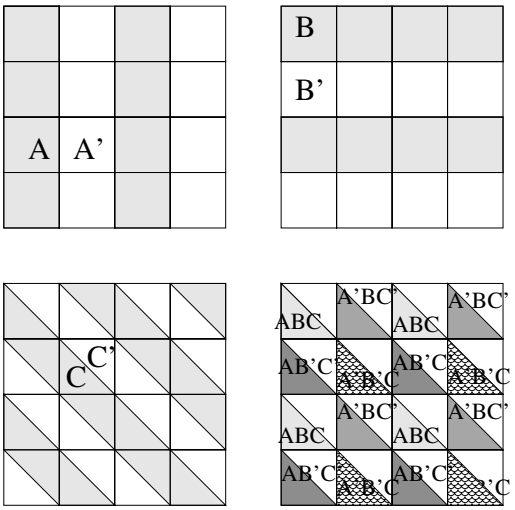

Fig. 8 Partition of the plane with vertical stripes $\mathbf{A}$ (even) and $\mathbf{A}'$ (odd), horizontal stripes $\mathbf{B}$ (even) and $\mathbf{B}'$ (odd), diagonal stripes $\mathbf{C}$ (even) and $\mathbf{C}'$ (odd). Then, the union of these strips gives a tiling with triangles. An advantage of this definition is that it uses only the basic, continuous function $d_{\mathbb{Z}}$.

3.6. Koch Flake

A parametric equation was proposed by Allouche *et al.*,[33] but there is no characteristic function up to now for the best of our knowledge. This section provides an equation for the von Koch snowflake. We use $d_m(x) = x \mod 1$, defined in previous subsection to define the tiling of triangles shown on the left side of Fig. 9, where $\{(x, y) \,|\, d_m(x - \frac{y}{\sqrt{3}}) + d_m(\frac{2y}{\sqrt{3}}) \leq 1\}$ are the white triangles, and $\{(x, y) \,|\, d_m(x - \frac{y}{\sqrt{3}}) + d_m(\frac{2y}{\sqrt{3}}) \geq 1\}$ are the gray triangles. The vertices of the lattice are $\mathbb{Z}(1, 0) + \mathbb{Z}(\frac{1}{2}, \frac{\sqrt{3}}{2}) = (i + \frac{1}{2}j, \frac{\sqrt{3}}{2}j), i, j \in \mathbb{Z}$.

Let

$$d_\Delta(x, y) = \min_i \min_j \sqrt{(x - x_{i,j})^2 + (y - y_{i,j})^2} - \sqrt{3}/6,$$

where (x_i, y_i) are the vertices of the lattice, then $d_\Delta(x, y) \leq 0$ defines the disks which are centered at these vertices, as shown on the right side of Fig. 9. Since $d_\Delta(x, y)$ is bounded by the length of the edges of the tiling, the series clearly converges. We give two more methods to define $d_\Delta(x, y)$ in Secs. 3.6.1 and 3.6.2.

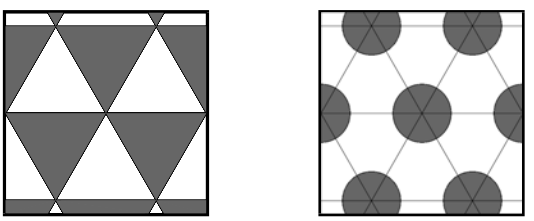

Fig. 9 Tiling of triangles and the Koch flake with $k = 0$ for $x \in [-1, 1]$, $y \in [-1, 1]$.

Now, we define a function

$$F_k(x,y) = \min_{i=0}^{k} \frac{d_\Delta(3^i x, 3^i y)}{3^i}. \tag{10}$$

The limit of $F_k(x,y)$ when $k \to +\infty$ is the expression of the signed distance to the Koch flake.

Furthermore, let

$$T(x,y) = \sqrt{3} \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix},$$

then

$$\widehat{F_k}(x,y) = \begin{cases} F_k(x,y) & \text{if } F_k(x,y) \le 0 \\ & \text{and } F_k(x,y) \\ & \le F_k(T(x,y)), \\ -F_k(T(x,y)) & \text{if } F_k(T(x,y)) \le 0 \\ & \text{and } F_k(T(x,y)) \\ & \le F_k(x,y), \\ 0 & \text{otherwise} \end{cases}$$

gives a definition of a double Koch flake, which is also a signed distance function to the boundary of the double Koch flake. An example of the recursively subdivision process is shown in Fig. 10. Furthermore, we also show a shaded single Koch flake and a shaded double Koch flake in Fig. 11.

It suffices to use a specific tuned procedure to evaluate $d_m(x)$ for intervals. These fractal functions should be inserted as a Directed Acyclic Graph (DAG) node, so when it is evaluated, the specific

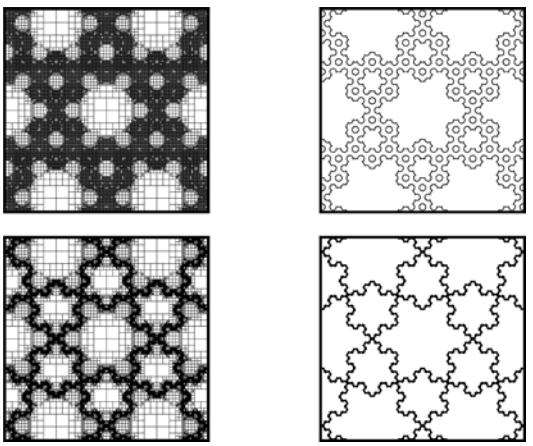

Fig. 10 Single Koch flake (upper row) and double Koch flake (lower row) for $x \in [-1,1]$, $y \in [-1,1]$. Left: the recursive subdivision with outward rounded interval arithmetic. Right: the corresponding cover.

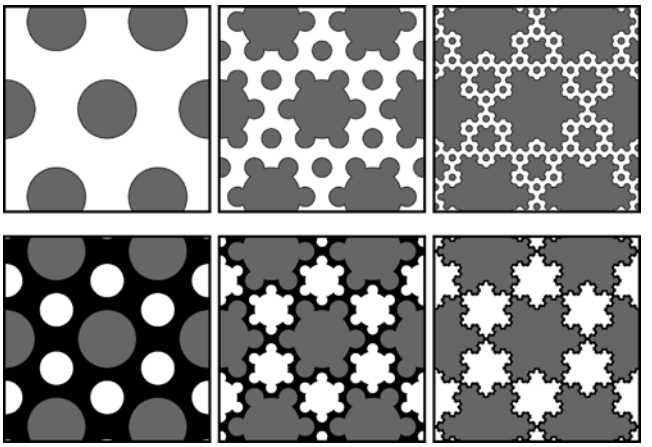

Fig. 11 The Koch flakes with $k = 0, 1, 2$ for $x \in [-1,1]$, $y \in [-1,1]$. Upper row: single Koch flake. Lower row: double Koch flake.

(tuned) procedure is called. Though the method proposed in the previous subsection can be used to get continuous functions, here, we propose methods only using $d_{\mathbb{Z}}(x)$. Following these methods, the Koch flake can be computed using interval arithmetic. In our implementation, we use (1) to compute the functions in a given box.

3.6.1. *2D affine map*

Let

$$M(x,y) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2\sqrt{3}} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix},$$

and denote $(x',y')^t = M(x,y)$, $(x'',y'')^t = M(x + \frac{1}{2}, y + \frac{\sqrt{3}}{2})$, let

$$f(x,y) = \sqrt{4d_{\mathbb{Z}}^2(x) + 12d_{\mathbb{Z}}^2(y)} - \sqrt{3}/6,$$

then

$$d_\Delta(x,y) = \min\{f(x',y'), f(x'',y'')\} \le 0$$

defines the expected disks on the right side of Fig. 9.

3.6.2. *3D affine map*

Here, we give another method to get the disks by projecting a 3D lattice to the Oxy plane. We search the isometry which transforms the equilateral lattice in the Oxy plane to the plane $\mathbf{P}$ with equation $x + y + z = 1$. The integer vertices in $\mathbf{P}$ form an equilateral 2D lattice, see Fig. 12.

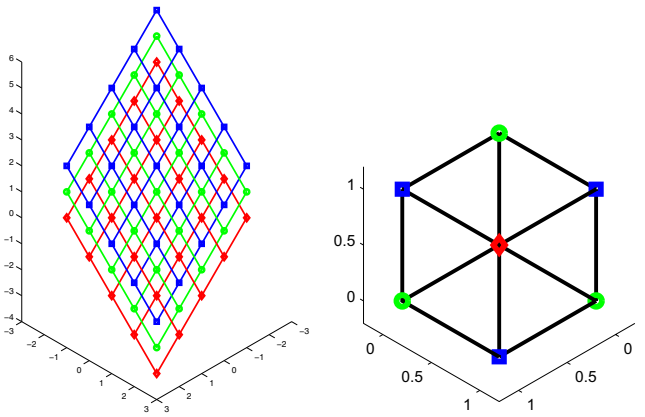

Fig. 12 The lattice on the plane $x+y+z=1$. Right: the diamond, circle, and square points are the projections of the integer vertices of the planes $x+y+z=0$, $x+y+z=1$, and $x+y+z=2$, respectively. Left: part of the lattice is shown.

In the Oxy plane, the lattice has length $\alpha = \frac{\sqrt{6}}{3}$, and is the set of vertices $\mathbb{Z}(\alpha, 0) + \mathbb{Z}(\frac{1}{2}\alpha, \frac{\sqrt{3}}{2}\alpha)$, see Fig. 12. Implicitly, $z = 0$ for points in the Oxy plane.

Now, the point $(0,0,0)$ of the Oxy plane is mapped to the point $(1,0,0)$ in the plane $\mathbf{P}$. The vector $(1,0,0)$ of the Oxy plane is mapped to the vector $(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)$ in the plane $\mathbf{P}$. The vector $(0,1,0)$ of the Oxy plane is mapped to the vector $(-\frac{1}{\sqrt{6}}, -\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}})$. The vector $(0,0,1)$ of the Oxy plane is mapped to the vector $(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$. Let M be the 4×4 matrix (we use homogeneous coordinates as usual) of the isometry mapping the Oxy plane to $\mathbf{P}$. We get

$$\begin{pmatrix} \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{6}} & \frac{1}{\sqrt{3}} & 1 \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{6}} & \frac{1}{\sqrt{3}} & 0 \\ 0 & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = M \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = M.$$

Thus, the matrix M is known. The matrix M permits to compute the distance to vertices of the equilateral lattice for the point (x,y): let $(x',y',z',1)^t = M(x,y,0,1)^t$, then the distance of (x,y) to the equilateral lattice $d_\Delta(x,y)$ (with side length $\alpha = \frac{\sqrt{6}}{3}$) can be computed as $d_\mathbb{Z}(x') + d_\mathbb{Z}(y') + d_\mathbb{Z}(z')$ for the Manhattan distance, or as $\sqrt{d_\mathbb{Z}^2(x') + d_\mathbb{Z}^2(y') + d_\mathbb{Z}^2(z')}$ for the Euclidean distance, or as $\max(d_\mathbb{Z}(x'), d_\mathbb{Z}(y'), d_\mathbb{Z}(z'))$ for L^∞.

Remark. We can consider irrational planes: the "cut-and-project" method intersects a slice around a nonrational (hyper) plane and a periodic tiling ($\mathbb{Z}^d$ for instance), and projects it on the (hyper)plane to obtain an aperiodic tiling.[34,35] Our method will likely generate an aperiodic fractal tiling.

3.6.3. *Signed distance to a single Koch flake*

In this section, we give another way to define the Koch flake. The vertices defined previously are the centers of the Koch flake. Without the loss of generality, we propose our method for the Koch flake which is centered at the origin point.

Let $d(x,y) = \sqrt{x^2+y^2} - r \leq 0$ be a signed distance function to a disk with radius $r = \sqrt{3}/6$, which is the biggest disk inside the Koch flake. Let (ρ, θ) be the polar coordinate of a point (x,y), and $(x',y') = M(x,y)$ is a transformation of a point in the plane, which is

$$M(x,y) = 3\left(\rho\cos\left(\frac{\pi}{3}d_{\mathbb{Z}_0}\left(\frac{3\theta}{\pi}\right)\right) - \frac{1}{3}, \rho\sin\left(\frac{\pi}{3}d_{\mathbb{Z}_0}\left(\frac{3\theta}{\pi}\right)\right)\right).$$

Then, we give the expression of the signed distance to the Koch flake as follows:

$$F(x,y) = \min_{k=0}^{\infty} \frac{d(M^k(x,y))}{3^k}. \qquad (11)$$

Thus, outward-rounded interval arithmetic can be used for the Koch flake. In Fig. 13, a Koch flake is generated using this method. We compute the function using (1), the recursive subdivision procedure is also shown.

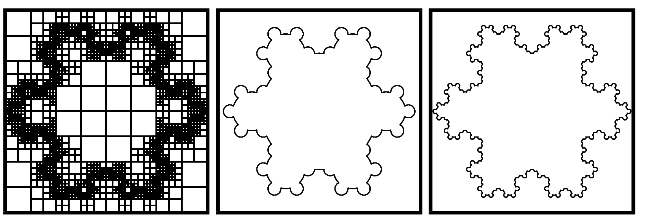

Fig. 13 The graph of the signed distance to the Koch flake. Left: the recursive subdivision algorithm. Middle: the von Koch curve with $k = 2$. Right: the resulting von Koch curve.

4. DISCUSSION AND CONCLUSION

This section formalizes our definition of fractals. First, we rely on some lattice $\mathbf{L}$; the simplest are $\mathbb{Z}^2$ and $\mathbb{Z}^3$; we also use the 2D equilateral lattice $\mathbb{Z}(1,0)+\mathbb{Z}(\frac{1}{2},\frac{\sqrt{3}}{2})$ for von Koch. We then define the distance function of point p to $\mathbf{L}$, i.e. to the vertices of $\mathbf{L}$, called $d(p,\mathbf{L})$. We then define a seed set $\mathbf{S}$, which is a periodic pattern, copied at each vertex of $\mathbf{L}$. In the simplest case, the pattern is a disk, so the characteristic function of $\mathbf{S}$ is $f_{\mathbf{S}}(p) = d(p,\mathbf{L}) - r$, where r is the radius of the disk. Finally, we use exactly one contractant similitude T; let σ be its eigenvalue: $|\sigma| < 1$. We define $\mathbf{S}_k = \cup_{i=0}^{k} T^i(\mathbf{S})$. The limit of $\mathbf{S}_k$ when $k \to \infty$ is the fractal, actually an infinite and periodic tiling of this fractal pattern. Note that it is an IFS with only one contractant transform. To define the characteristic function of $\mathbf{S}_k$, we use the classical property:

$$d(p, T^k(\mathbf{S})) = |\sigma|^{-k} d(T^{-k}(p), \mathbf{S})$$

(reusing the classical idea routinely used to raytrace CSG scenes). Thus, the characteristic function of $\mathbf{S}_k$ is

$$f_{\mathbf{S}_k}(p) = \min_{i=0}^{k} |\sigma|^{-i} d(T^{-i}(p), \mathbf{S}). \tag{12}$$

Because the function $d(p \in \mathbb{R}^2, \mathbf{S})$ is bounded below and above, the terms of (12) converge in a controllable way, so it is possible to compute quickly and accurately the characteristic function of the fractal. Finally, note that instead of T^k, we can use $T'_k = f(k) \circ T^k \circ g(k)$, which composes T^k with some translation and rotation $f(k)$ and $g(k)$ depending on k, as in (11).

In this paper, fractal functions have the form $\sum_{i=0}^{\infty} \frac{K(\alpha^i x)}{\beta^i}$, where the sum $\sum$ can be replaced with min or max, and where the seed or kernel function K is bounded, which ensures convergence and computability ("enclosability"). In passing, note the analogy with the noise/turbulence Perlin's function proposed by Ken Perlin in 1985 to generate procedural 3D textures[36]:

$$\text{NOISE}(x) = \sum_{i=0}^{N-1} \frac{\text{Noise}(b^i x)}{a^i},$$

where $\text{Noise}(x)$ is the Perlin's noise[36] or Mandelbrot's F-noise function,[11] N is typically between 6 and 10, b is some positive number greater than 1, most commonly it will be powers of 2. The parameter a controls how rough the final $\text{NOISE}(x)$ function will be. Small values of a, e.g. 1, give very rough functions, larger values give smoother functions.

We conclude with some remarks or arising issues. In function-based modeling, expressions are often represented with DAG, roughly trees with shareable nodes. A leave is a symbol x, y, z, a number or an interval, and nodes are binary operators $(+, -, \times,$ etc) or special functions $(\cos, \sin,$ etc.). To account for fractal functions, it is convenient to introduce new kind of nodes, i.e. $d_{\mathbb{Z}}$ node, or a $d(M\mathbb{Z}^d+T, x \in \mathbb{R}^d)$ node, where $M\mathbb{Z}^d+T$ is the image of the lattice $\mathbb{Z}^d$ after some affine map $x \to Mx+T$, and some new DAG to store $\min_{i=i_0}^{\infty} K(i,x)$, $\max_{i=i_0}^{\infty} K(i,x)$, $\sum_{i=i_0}^{\infty} K(i,x)$, where the expression for K is also represented with some DAG.

We found our first Rvachev functions by trials and errors. Is it possible to find by algorithm the Rvachev functions (for instance, a signed distance function) for other classical fractals, like the Julia set,[19,21,37] or defined by an IFS[10] or a Controlled IFS (remember that in CIFS, some patterns are forbidden, like T_1^2 if T_1 is one of the IFS transforms), or finite automata? It is an open question, linked to Number Theory, Lattice Theory, Automata Theory and Harmonic Analysis.

Is it possible to reconstruct a signed distance function from a picture? The analogy with harmonic analysis, Fourier transform, may give some insight. Note also the analogy of this question with procedural 3D textures: the latter are synthetized with Fourier analysis from 3D pictures e.g. by Ghazanfarpour and Dischler.[38]

How to contain the wrapping effect (e.g. $d_m(x) = x \mod 1$) for these interval computations?

For fractal functions f which are not Lipschitz continuous, but Hölder continuous, it seems possible to generalize the centered evaluation form of interval analysis with an Hölder evaluation form. Remember that a function is (k,h) Hölder when $|f(b) - f(a)| \leq k(b-a)^h$. Thus, the issue of computing (k,h) values for nondifferentiable functions defined by DAG arises.

This paper shows that nondifferentiable functions, and even noncontinuous functions like the modulo 1 function, are computable (enclosable) in polynomial time with interval arithmetic. These functions are Rvachev functions of geometric objects with fractal geometry. This paper provides Rvachev functions for some classical fractals, which were found by trials and errors. These

Rvachev functions permit to extend the classical RSS method to fractals. We wonder if it is algorithmically possible to get these functions from a given IFS, or a finite automaton.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for Ref. 24, and for their valuable comments and suggestions that helped to improve the paper. The work is supported, in part, by Zhejiang Provincial Natural Science Foundation of China (LY18F020023).

REFERENCES

1. V. Shapiro, Real functions for representation of rigid solids, *Comput.-Aided Geom. Des.* **11**(2) (1994) 153–175.
2. A. Gomes, I. Voiculescu, J. Jorge, B. Wyvill and C. Galbraith, *Implicit Curves and Surfaces*: *Mathematics*, *Data Structures and Algorithms* (Springer, London, 2009).
3. O. Fryazinov, A. Pasko and P. Comninos, Fast reliable interrogation of procedurally defined implicit surfaces using extended revised affine arithmetic, *Comput. Graph.* **34**(6) (2010) 708–718.
4. A. Pasko, V. Adzhiev, A. Sourin and V. Savchenko, Function representation in geometric modeling: Concepts, implementation and applications, *Visual Comput.* **11**(8) (1995) 429–446.
5. D. Storti, C. Finley and M. Ganter, Interval extensions of signed distance functions: iSDF-reps and reliable membership classification, *J. Comput. Inf. Sci. Eng.* **10**(2) (2010) 021012.
6. Y. D. Fougerolle, A. Gribok, S. Foufou and F. Truchetet, Boolean operations with implicit and parametric representation of primitives using r-functions, *IEEE Trans. Vis. Comput. Graphics* **11**(5) (2005) 529–539.
7. K. Weihrauch, *Computability* (Springer, Berlin, London, 1987).
8. L. Jaulin, M. Kieffer, O. Didrit and E. Walter, *Applied Interval Analysis with Examples in Parameter and State Estimation*, *Robust Control and Robotics* (Springer-Verlag, London, 2001).
9. C. Funfzig, D. Michelucci and S. Foufou, Polytope-based computation of polynomial ranges, *Comput.-Aided Geom. Des.* **29**(1) (2012) 18–29.
10. M. Barnsley, *Fractals Everywhere* (Academic Press Professional, USA, 1988).
11. B. B. Mandelbrot, *The Fractal Geometry of Nature* (W. H. Freedman and Co., New York, 1983).
12. J. L. Ramírez, G. N. Rubiano and B. J. Zlobec, Generating fractal patterns by using p-circle inversion, *Fractals* **23**(4) (2015) 1550047.
13. A. H. Barr, Global and local deformations of solid primitives, *ACM SIGGRAPH Comput. Graph.* **18**(3) (1984) 21–30.
14. L. Evans, *Partial Differential Equations*, 2nd edn. (Wadsworth & Brooks/Cole Mathematics) (American Mathematical Society, Providence, RI, 2010), pp. 211–223.
15. V. Shapiro, Semi-analytic geometry with R-functions, *Acta Numer.* **16** (2007) 239–303.
16. J.-P. Allouche and J. O. Shallit, *Automatic Sequences — Theory*, *Applications*, *Generalizations* (Cambridge University Press, Cambridge, 2003).
17. D. Hepting, P. Prusinkiewicz and D. Saupe, Rendering methods for iterated function systems, *Fractals Fund. Appl. Sci.* **11** (1991) 183–224.
18. D. Michelucci, Reliable representations of strange attractors, in *Scientific Computing*, *Validated Numerics*, *Interval Methods* (Springer, Boston, 2001), pp. 379–390.
19. D. Michelucci and S. Foufou, Interval-based tracing of strange attractors, *Int. J. Comput. Geom. Appl.* **16**(1) (2006) 27–39.
20. A. Paiva, L. H. de Figueiredo and J. Stolfi, Robust visualization of strange attractors using affine arithmetic, *Comput. Graph.* **30**(6) (2006) 1020–1026.
21. J. C. Hart, D. J. Sandin and L. H. Kauffman, Ray tracing deterministic 3D fractals, *ACM SIGGRAPH Comput. Graph.* **23** (1989) 289–296.
22. M. N. Gamito and S. C. Maddock, Ray casting implicit fractal surfaces with reduced affine arithmetic, *Visual Comput.* **23**(3) (2007) 155–165.
23. O. Fryazinov and A. Pasko, Interactive ray shading of FRep objects, in *WSCG*'2008, *Communications Papers Proc.* (Václav Skala - Union Agency, Plzen, Czech Republic, 2008), pp. 145–152.
24. K. V. Maksymenko-Sheyko and T. I. Sheyko, Mathematical modeling of geometric fractals using R-functions, *Cybern. Syst. Anal.* **48**(4) (2012) 614–620.
25. J. Rice, Spatial bounding of self-affine iterated function system attractor sets, in *Proc. Conf. Graphics Interface* '96, *GI* '96, Canadian Information Processing Society, Toronto, Canada (1996), pp. 107–115.
26. V. Berthé and A. Siegel, Tiling associated with beta numeration and substitution, *Integers* **5**(3) (2005) A02.
27. V. Berthé and M. Rigo, *Combinatorics*, *Automata and Number Theory*, Vol. 135 (Cambridge University Press, Cambridge, 2010).
28. R. Martin, H. Shou, I. Voiculescu, A. Bowyer and G. Wang, Comparison of interval methods

for plotting algebraic curves, *Comput.-Aided Geom. Des.* **19**(7) (2002) 553–587.
29. A. Mishkinis, Extension of algorithmic geometry to fractal structures, Phd thesis, Université de Bourgogne (2013).
30. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, 3rd edn. (MIT Press, Cambridge, 2009).
31. D. H. Bailey and R. E. Crandall, Random generators and normal numbers, *Exp. Math.* **11**(4) (2002) 527–546.
32. T. D. Taylor, Using epsilon hulls to characterize and classify totally disconnected Sierpiński relatives, *Fractals* **23**(2) (2015) 216–481.
33. J.-P. Allouche and G. Skordev, Von Koch and Thue–Morse revisited, *Fractals* **15**(4) (2006) 405–409.
34. N. De Bruijn, Algebraic theory of Penrose's non-periodic tilings of the plane. I, in *Indagationes Mathematicae* (*Proc.*), Vol. 84 (Elsevier, Eindhoven, The Netherlands, 1981), pp. 39–52.
35. T. T. Le, Local rules for quasiperiodic tilings, *NATO ASI Ser. C Math. Phys. Sci.* **489** (1997) 331–366.
36. K. Perlin, An image synthesizer, *ACM SIGGRAPH Comput. Graph.* **19** (1985) 287–296.
37. J. C. Hart, L. H. Kauffman and D. J. Sandin, Interactive visualization of quaternion julia sets, in *Proc.* 1*st Conf. Visualization*' 90 (IEEE Computer Society Press, New York, 1990), pp. 209–218.
38. D. Ghazanfarpour and J. M. Dischler, Spectral analysis for automatic 3D texture generation, *Comput. Graph.* **19**(3) (1995) 413–422.