Theoretical Computer Science 145 (1995) 241–270

A polynomial-time computable curve whose interior has a nonrecursive measure☆

Ker-I Ko*

Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY 11794, USA

Received August 1993; Revised April 1994
Communicated by P. Young

Abstract

A polynomial-time computable simple curve is constructed such that its measure in the two-dimensional plane is positive. This construction is applied to prove the following two results: (1) there exists a polynomial-time computable simple closed curve in the two-dimensional plane such that the measure of its interior region is a nonrecursive real number; (2) there exists a polynomial-time computable simple curve in the two-dimensional plane such that its length is finite but is a nonrecursive real number.

1. Introduction

We investigate the problem of measuring the area of the interior of a given simple closed curve in $\mathbb{R}^2$ and the problem of measuring the length of a given simple curve in $\mathbb{R}^2$. For the first problem, we recall that the integral of a polynomial-time computable real function $f:[0,1]\to\mathbb{R}$ is computable in polynomial time by an oracle machine using a function $\phi\in\#P$ as an oracle [2]. It implies that for any polynomial-time computable, simple, closed, rectifiable curve Γ,[1] the measure of the interior of Γ is also computable in polynomial time relative to $\#P$ (see [1] for more details). In this paper, we construct a polynomial-time computable function $f:[0,1]\to\mathbb{R}^2$ whose image is a simple, closed, nonrectifiable curve Γ_f such that its interior has a nonrecursive measure. In other words, the polynomial-time computability of the boundary for a region does not even imply the computability of the area of the region.

☆ The main results of this paper has been announced in the abstract "Some Complexity Issues on the Simply Connected Regions of the Two-Dimensional Plane" in Proc. 25th ACM Symp. on Theory of Computing, 1993.

* E-mail: keriko@sbcs.sunysb.edu Research supported in part by NSF grant CCR 9121472.

[1] A curve Γ is called rectifiable if it has a finite length.

SSDI 0304-3975(94)00154-5

The second problem, the problem of computing the length of a given simple curve, is called the *coastline problem in* [6]. It is recognized as a difficult problem since the length of the given curve could actually be infinite. In this paper, we construct a polynomial-time computable function $f:[0,1]\to\mathbb{R}^2$ whose image is a simple, rectifiable curve having a nonrecursive length. This result justifies the difficulty of the coastline problem: even if the length of the curve is finite, it is not necessarily computable.

The computational model used in this study is that of Ko and Friedman [4] that is based on the model of recursive analysis (see also [3] for a complete treatment). Ref. [1] contains the formal definitions for the notion of polynomial-time computability of two-dimensional regions.

The main ingredient of the proofs of the above two results is a modification of the Peano space-filling curve. That is, we construct a simple curve Γ in polynomial time that has a positive measure in $\mathbb{R}^2$. This curve is not exactly the same as the classical Peano space-filling curve since it is defined by a one-to-one function. Rather, it is very similar to the curves constructed by Lebesgue [5] and Osgood [7] which have recently found interesting applications in the construction of fractal sets [6]. Such a curve is called a Lebesgue–Osgood monster in [6]. Both of the main nonrecursiveness results are based on the construction of the Lebesgue–Osgood monster curves and their variations. Although the idea of the construction is similar to Lebesgue's and Osgood's original construction, we choose to present a precise definition of this monster curve, since we need to show that such a construction can be done in polynomial time. Furthermore, we need formal proofs for other properties of the curve to obtain our main results.

We give a brief review of the computational model for real functions in Section 2. The complete definition of our version of the Lebesgue–Osgood monster curve is presented in Section 3, and the various properties of this curve are proved in Section 4. Sections 5 and 6 contain the main results. The construction used for our first main result has some other interesting application to the complexity theory of two-dimensional regions developed in [1]. We prove this result in Section 7.

2. Polynomial-time computable real functions

In this paper, we will write $\mathbf{z}$ or $\langle x,y\rangle$ to denote a point in $\mathbb{R}^2$. A directed line segment L from $\mathbf{z}_1$ to $\mathbf{z}_2$ is denoted as line $(\mathbf{z}_1,\mathbf{z}_2)$. The length of an interval $I=[a,b]$ is denoted as leng(I) and the length of a line segment L is denoted as leng(L).

The concept of polynomial-time computable real functions used in this paper is that of Ko and Friedman [4] (see [3] for a complete treatment). The basic representation system for real numbers is the set $\mathbf{D}$ of dyadic rational numbers; that is, the set of all rationals which have finite binary representations. A real number $x\in\mathbb{R}$ is represented by a *Cauchy function* $\phi:\mathbf{N}\to\mathbf{D}$ that binary converges to x in the sense that $|\phi(n)-x|\leqslant 2^{-n}$. A real number x is *recursive* if there exists a computable Cauchy function ϕ that binary converges to x. It is *polynomial-time computable* if there exist a Turing machine M and a polynomial p such that for all inputs $n > 0$, $M(n)$ outputs a dyadic rational d in time $p(n)$ such that $|d - x| \leqslant 2^{-n}$. A sequence $\{x_n\}$ of real numbers is *polynomial-time computable* if there exist a Turing machine M and a polynomial p such that for all inputs $n, k > 0$, $M(n, k)$ outputs a dyadic rational d in time $p(n + k)$ such that $|d - x_k| \leqslant 2^{-n}$.

Definition 2.1. A function $f: [0, 1] \to \mathbb{R}^2$ is *polynomial-time computable* if there exist an oracle Turing-machine M and a polynomial p such that for any oracle function ϕ that binary converges to a real number $x \in [0, 1]$, and any integer n as the input, the machine M outputs, in time $p(n)$, two dyadic rationals d_1 and d_2 such that $|\langle d_1, d_2\rangle - f(x)| \leqslant 2^{-n}$. In other words, the oracle machine computes the operator that maps a Cauchy function for x to two Cauchy functions for $f(x)$.

An equivalent definition for polynomial-time computable real functions f will be used in this paper. We say a function $f: [0, 1] \to \mathbb{R}^2$ has a *polynomial modulus* if there exists a polynomial p such that $|x - y| \leqslant 2^{-p(n)}$ implies $|f(x) - f(y)| \leqslant 2^{-n}$.

Proposition 2.2. *A function $f: [0, 1] \to \mathbb{R}^2$ is polynomial-time computable if and only if*
(i) *f has a polynomial modulus and*
(ii) *there exist a Turing machine M and a polynomial p such that for any dyadic rational d of length $\leqslant n$,*[2] *and any integer n, $M(d, n)$ outputs, in time $p(n)$, two dyadic rationals e_1 and e_2 such that $|\langle e_1, e_2\rangle - f(d)| \leqslant 2^{-n}$.*

A function $f: [0, 1] \to \mathbb{R}^2$ could also be defined as the limit of a polynomial-time computable sequence of real functions.

Definition 2.3. A sequence $\{f_n\}$ of functions from $[0, 1]$ to $\mathbb{R}^2$ is *polynomial-time computable* if there exist a Turing machine M and a polynomial p such that for any oracle function ϕ that binary converges to a real number $x \in [0, 1]$ and for any input (n, k), M outputs, in time $p(n + k)$, two dyadic rationals e_1, e_2 such that $|\langle e_1, e_2\rangle - f_k(x)| \leqslant 2^{-n}$.

Proposition 2.4. *A function $f: [0, 1] \to \mathbb{R}^2$ is polynomial-time computable iff there exists a polynomial-time computable sequence $\{f_n\}$ of functions from $[0, 1]$ to $\mathbb{R}^2$ such that for some polynomial p, $|f_{p(n)}(x) - f(x)| \leqslant 2^{-n}$ for all $x \in [0, 1]$.*

3. The Lebesgue–Osgood monster

We are going to define a sequence $\{g_n\}$ of one-to-one functions from $[0, 1]$ to $\mathbb{R}^2$. The functions g_n are defined in such a way that its limit g exists and is also one-to-one.

[2] The length of a dyadic rational d is the length of its binary representation without trailing zeros.

Since the construction is very complicated, we first give a brief overview of the construction.

Basically, each function g_n divides $[0,1]$ onto 6^n subintervals and maps each subinterval into a short line segment in $\mathbb{R}^2$. These short line segments are uniformly distributed in the square $[-2,2]\times[-2,2]$ so that they almost "cover" the whole square. The function g_{n+1} is then obtained from g_n by expanding each line segment L of g_n into up to 6 more shorter line segments surrounding the original line segment L. Thereby, they "cover" the square $[-2,2]\times[-2,2]$ even more densely. Furthermore, the expansion from g_n to g_{n+1} leaves the two functions sufficiently close so that they eventually converge to a limit function g.

The above is the basic idea of the construction of a space-filling curve. To make the functions g_n and the limit function g to be one-to-one, however, we must modify the construction to allow extra space between any two nonadjacent line segments of g_n so that the expansion of a line segment of g_n to more shorter line segments of g_{n+k}, for any $k > 0$, does not meet other parts of the expanded curves of g_{n+k}. This requirement makes the construction much more complicated, and also makes the resulting limit function g not able to cover the whole unit square. We will, however, show that it is possible to guarantee that g still covers a large portion of the square.

We now begin the formal construction. We first define three sequences of rational numbers, which will be used to define the length of line segments of functions g_n.

Definition 3.1. (a) For each $n \geqslant 0$, $\alpha_{2n+1} = 5^{-n} + 4^{-1}\cdot 5^{-2n}$, and $\alpha_{2n} = 2\alpha_{2n+1} = 2\cdot 5^{-n} + 2^{-1}\cdot 5^{-2n}$.

(b) For each $n \geqslant 0$, $\beta_{2n+1} = \alpha_{2n+1} = 5^{-n} + 4^{-1}\cdot 5^{-2n}$, and $\beta_{2n+2} = 2^{-1}\cdot\beta_{2n+1} = 2^{-1}\cdot 5^{-n} + 8^{-1}\cdot 5^{-2n}$.

(c) For each $n \geqslant 0$, $\delta_{2n+1} = (24)^{-1}\cdot 5^{-(2n-1)}$, and $\delta_{2n+2} = 2^{-1}\cdot\delta_{2n+1}$.

We observe that these numbers satisfy the following properties.

Lemma 3.2. (a) *For each* $n \geqslant 0$, $\alpha_{2n+1} = 2\sum_{j=n+1}^{\infty}\alpha_{2j} + \delta_{2n+1}$.

(b) *For each* $n \geqslant 0$, $\alpha_n > \sum_{j=n+1}^{\infty}\alpha_j$.

(c) *For each* $n \geqslant 1$, $\beta_{2n} = 2\sum_{j=n+1}^{\infty}\beta_{2j+1} + \delta_{2n}$.

(d) *For each* $n \geqslant 0$, $\beta_n > \sum_{j=n+1}^{\infty}\beta_j$.

(e) *For each* $n \geqslant 1$, $\alpha_n \leqslant \beta_n \leqslant 3\cdot 5^{-n/2}$ *and* $\alpha_n\cdot\beta_n > 5^{-(n-1)}$.

Proof. (a)

$$2\cdot\sum_{j=n+1}^{\infty}\alpha_{2j} = 4\cdot\sum_{j=n+1}^{\infty}\alpha_{2j+1} = 4\cdot\sum_{j=n+1}^{\infty}(5^{-j} + 4^{-1}\cdot 5^{-2j})$$

$$= 4\cdot\frac{5^{-(n+1)}}{1-5^{-1}} + \frac{5^{-2(n+1)}}{1-5^{-2}} = 5^{-n} + (24)^{-1}\cdot 5^{-2n} = \alpha_{2n+1} - \delta_{2n+1}.$$

(b) It is clear that $\alpha_n \geqslant 2\alpha_{n+1}$. If n is odd then, from (a),

$$\alpha_n > \sum_{j=(n+1)/2}^{\infty} (2\alpha_{2j}) \geqslant \sum_{j=(n+1)/2}^{\infty} (\alpha_{2j} + \alpha_{2j+1}) = \sum_{j=n+1}^{\infty} \alpha_j.$$

If n is even then, by above,

$$\alpha_n \geqslant 2\alpha_{n+1} > \alpha_{n+1} + \sum_{j=n+2}^{\infty} \alpha_j.$$

(c) $\beta_{2n} = 2^{-1}\alpha_{2n-1}$, and by (a) above, is equal to

$$\sum_{j=n}^{\infty} \alpha_{2j} + 2^{-1}\delta_{2n-1} = \sum_{j=n}^{\infty} 2\alpha_{2j+1} + \delta_{2n} = 2\sum_{j=n}^{\infty} \beta_{2j+1} + \delta_{2n}.$$

(d) Similar to (b).

(e) It follows from simple calculations. □

Next, we define an infinite, ordered tree T that is to be used to define the relation between line segments of g_n and line segments of g_{n+1}.

Definition 3.3 (*Tree T*). There are three type of nodes in T: type A, type B and type C. Each node in T has one of 8 labels: ru, rd, lu, ld, ur, ul, dr, dl. The root is a type-A node with label ru. Each type-A node with label xy has 6 children:

(1) the first and sixth children are of type A and have label yx,
(2) the third and fourth children are of type A and have label $\bar{\text{y}}$x,
(3) the second child of type B and has label x$\bar{\text{y}}$, and
(4) the fifth child is of type B and has label xy,

where $\bar{\text{y}}$ is the complement of y in the sense that {y, $\bar{\text{y}}$} = {u, d} or {y, $\bar{\text{y}}$} = {l, r}. Each type-B node with label xy has 3 children:

(1) the first child is of type A and has label x$\bar{\text{y}}$,
(2) the second child is of type C and has label xy, and
(3) the third child is of type A and has label xy.

Each type-C node with label xy has 3 children:

(1) the first child is of type A and has label $\bar{\text{y}}$x,
(2) the second child is of type B and has label xy, and
(3) the third child is of type A and has label yx.

The top of the three T is shown in Fig. 1.

The label of a node v in the tree T is denoted by label(v) and its first character is denoted by label$_1$(v). The character label$_1$(v) is used to denote the direction of a directed line segment L_v associated with v (to be defined later). The label r means that the line segment L_v is horizontal and pointing right; the label l means that the line segment L_v is horizontal and pointing left; the label u means that the line segment L_v is vertical and pointing upward; the label d means that the line segment L_v is vertical and

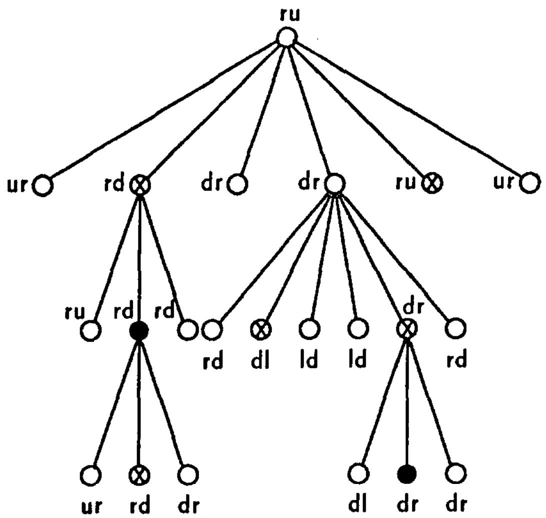

Fig. 1. The top part of tree T. In the above, a ∘ denotes a type-A node, a ⊗ denotes a type-B node and a • denotes a type-C node.

pointing downward. We say a node v in T is a *horizontal node* if $\text{label}_1(v) \in \{\mathsf{r}, \mathsf{l}\}$, and it is a *vertical node* if $\text{label}_1(v) \in \{\mathsf{u}, \mathsf{d}\}$. (The second character of label(v) often, but not always, denotes the direction of the line segment of the right neighbor of v.) The following lemma shows the pattern of the directions of L_v based on the depth of the node v.

Lemma 3.4. *Let v be a node of depth n.*

(a) *If v is of type* A *or type* C, *then v is a horizontal node if and only if n is even.*

(b) *If v is of type* B, *then v is a horizontal node if and only if n is odd.*

Proof. We prove the lemma by induction on n. First, the root is of depth 0 and of type A and is horizontal.

For the inductive step, let v be a type-A node of depth $n > 0$. If its parent u is of type A or type C, then from the definition of the tree T we know that v is a horizontal node if and only if u is a vertical node. Thus, from the inductive hypothesis, v is horizontal if and only if n is even. If its parent u is of type B then, from the definition of the three T, v is horizontal if and only if u is horizontal. Again, by the inductive hypothesis, v is horizontal if and only if n is even.

Next, assume that v is of type B. Then its present u must be either of type A or of type C. In either case, u and v must be both horizontal or both vertical. It follows from the inductive hypothesis that v is horizontal if and only if n is odd.

Finally, assume that v is of type C. Then its parent u must be of type B and u and v must be both horizontal or both vertical. Again, by the inductive hypothesis, we conclude that v is horizontal if and only if n is even. □

For each node v of type B or of type C, let anc(v) denote the lowest ancestor u of v such that all nodes between node u and node v, including node u, are not of type A. (Thus, anc(v) are always of type B.) We let k_v denote the depth of anc(v). For instance, assume that w_0 is of depth n and of type A, and that for each $i \geqslant 0$, w_{i+1} is the second child of w_i, then all nodes w_i, $i \geqslant 1$, have anc$(w_i) = w_1$ and $k_{w_i} = n + 1$. Note that if v is of type B or C and its parent u is also of type B or C, then $k_v = k_u$.

Now we will define functions g_n. For each $n \geqslant 0$, the function g_n is defined in terms of the nodes of T of depth n. For each node v of the three T of depth n, we are going to define an interval $I_v = [a_v, b_v]$ and a directed line segment L_v. The function g_n is the function that maps each interval I_v linearly to the line segment L_v. These intervals I_v and line segments L_v of any node v of depth $n \geqslant 0$ will satisfy the following properties.

(1) The direction of the line segment L_v is determined by label$_1(v)$ as described above.
(2) For any type-A node v, leng$(I_v) = 6^{-n}$, with $a_v = i \cdot 6^{-n}$ for some $i \geqslant 0$.
(3) For any type-A node v, if v is a horizontal node then leng$(L_v) = \alpha_n$ and if v is a vertical node then leng$(L_v) = \beta_n$.
(4) For any type-B or type-C node v, leng$(I_v) = 6^{-k_v} - 2\sum_{j=k_v+1}^{n} 6^{-j}$ and $a_v = i \cdot 6^{-n}$ for some $i \geqslant 0$
(5) For any type-B node v, if v is horizontal then leng$(L_v) = \alpha_n + \delta_{k_v} - \delta_n$; and if v is vertical then leng$(L_v) = \beta_n + \delta_{k_v} - \delta_n$.
(6) For any type-C node v, if v is horizontal then leng$(L_v) = \alpha_{n+1} + \delta_{k_v} - \delta_{n+1}$; and if v is vertical then leng$(L_v) = \beta_{n+1} + \delta_{k_v} - \delta_{n+1}$.
(7) For each $n \geqslant 0$, the intervals I_v of all nodes v of depth n form a partition of the interval $[0, 1]$ in the sense that if v_1 and v_2 are two neighboring nodes at depth n, then the end point of I_{v_1} is identical to the starting point of I_{v_2}. Furthermore, the line segments L_v also have this property; that is, if v_1 and v_2 are two neighboring nodes at depth n, then the end point of L_{v_1} is identical to the starting point of L_{v_2}.

Definition 3.5. We define I_v and L_v recursively as follows: First, for $n = 0$, $I_v = [0, 1]$ and L_v is the line segment define from $\langle -\alpha_1, 0\rangle$ to $\langle \alpha_1, 0\rangle$ (i.e., $L_v = $ line$(\langle -\alpha_1, 0\rangle, \langle \alpha_1, 0\rangle)$). Next, for any depth-$n$ node v, we assume that I_v and L_v have been defined and satisfy the above properties. We define I_w and L_w for each child w of v as follows:

Case 1: v is of type A. Then, we have leng$(I_v) = 6^{-n}$ and $a_v = j \cdot 6^{-n}$ for some $j \geqslant 0$. Let w_i, $1 \leqslant i \leqslant 6$, be the 6 children of node v. Divide I_v into 6 subintervals each of length $6^{-(n+1)}$, and let I_{w_i} be the ith subinterval, $1 \leqslant i \leqslant 6$; that is, $I_{w_i} = [(6j + i - 1)6^{-(n+1)}, (6j + i)6^{-(n+1)}]$.

We describe the six line segments L_{w_i}, $1 \leqslant i \leqslant 6$, by their starting points, their directions and their lengths:

(i) The line segment L_{w_1} begins at the starting point of L_v, and for $i = 2, \ldots, 6$, the line segment L_{w_i} begins at the end point of $L_{w_{i-1}}$.
(ii) For each $i = 1, \ldots, 6$, the line segment L_{w_i} goes toward the direction of label$_1(w_i)$ and has length equal to α_{n+1} if w_i is horizontal and equal to β_{n+1} if w_i is vertical.

For instance, if label(v) = ru, and $L_v = \text{line}(\langle x, y\rangle, \langle x+\alpha_n, y\rangle)$ for some $x, y \in \mathbb{R}$, then we have

$$L_{w_1} = \text{line}(\langle x, y\rangle, \langle x, y+\beta_{n+1}\rangle),$$

$$L_{w_2} = \text{line}(\langle x, y+\beta_{n+1}\rangle, \langle x+\alpha_{n+1}, y+\beta_{n+1}\rangle),$$

$$L_{w_3} = \text{line}(\langle x+\alpha_{n+1}, y+\beta_{n+1}\rangle, \langle x+\alpha_{n+1}, y\rangle),$$

$$L_{w_4} = \text{line}(\langle x, +\alpha_{n+1}, y\rangle, \langle x+\alpha_{n+1}, y-\beta_{n+1}\rangle),$$

$$L_{w_5} = \text{line}(\langle x+\alpha_{n+1}, y-\beta_{n+1}\rangle, \langle x+2\alpha_{n+1}, y-\beta_{n+1}\rangle),$$

$$L_{w_6} = \text{line}(\langle x+2\alpha_{n+1}, y-\beta_{n+1}\rangle, \langle x+2\alpha_{n+1}, y\rangle).$$

Fig. 2(a) shows the relation between functions g_0 and g_1, which is similar to the relation between L_v and L_{w_i}'s in the above example.

Case 2: v is of type B or type C. Let the three children of v be w_1, w_2 and w_3. Assume that $I_v = [a_v, b_v]$. Then, we let $I_{w_1} = [a_v, a_v + 6^{-(n+1)}]$, $I_{w_2} = [a_v + 6^{-(n+1)}, b_v - 6^{-(n+1)}]$ and $I_{w_3} = [b_v - 6^{-(n+1)}, b_v]$.

The three line segments L_{w_i}, $i = 1, 2, 3$, are defined as follows:

(i) L_{w_1} begins at the starting point of L_v, and L_{w_i}, $i = 2, 3$, begins at the end point of $L_{w_{i-1}}$;

(ii) L_{w_i}, $i = 1, 2, 3$, goes toward the direction of $\text{label}_1(w_i)$;

(iii) $\text{leng}(L_{w_1}) = \text{leng}(L_{w_3})$ and it is equal to α_{n+1} if w_1 and w_3 are horizontal, and equal to β_{n+1} if they are vertical; and

(iv) If v is of type B, then $\text{leng}(L_{w_2}) = \text{leng}(L_v) - 2\cdot\text{leng}(L_{w_1})$, and if v is of type C, then $\text{leng}(L_{w_2}) = \text{leng}(L_v)$.

For instance, Assume that label(v) = ru, and $L_v = \text{line}(\langle x_1, y\rangle, \langle x_2, y\rangle)$, for some $x_1, x_2, y \in \mathbb{R}$, $x_1 < x_2$. If v is of type B, then

$$L_{w_1} = \text{line}(\langle x_1, y\rangle, \langle x_1+\alpha_{n+1}, y\rangle),$$

$$L_{w_2} = \text{line}(\langle x_1+\alpha_{n+1}, y\rangle, \langle x_2-\alpha_{n+1}, y\rangle),$$

$$L_{w_3} = \text{line}(\langle x_2-\alpha_{n+1}, y\rangle, \langle x_2, y\rangle).$$

If v is of type C, then

$$L_{w_1} = \text{line}(\langle x_1, y\rangle, \langle x_1, y-\beta_{n+1}\rangle),$$

$$L_{w_2} = \text{line}(\langle x_1, y-\beta_{n+1}\rangle, \langle x_2, y-\beta_{n+1}\rangle),$$

$$L_{w_3} = \text{line}(\langle x_2, y-\beta_{n+1}\rangle, \langle x_2, y\rangle).$$

The above completes the definition of I_v and L_v. In the following, we verify the above 7 properties of the intervals I_v and line segments L_v. Since we defined them recursively, we only need to verify that if I_v and L_v satisfy these properties, then I_{w_i} and L_{w_i} of all its children w_i also satisfy these properties.

(1) We define the direction of L_v based on $\text{label}_1(v)$.

(2) and (3) Clear from the definition.

(4) We prove this by induction on the depth n. Assume that v is of depth $n > 0$ and is of type B or C. Let u be its parent node.

Case 1: u is of type A. Then, $\text{leng}(I_u) = 6^{-(n-1)}$, and $\text{leng}(I_v) = 6^{-n}$. Note that $k_v = n$ and so the property (4) holds.

Case 2: u is of type B or C. Then $\text{leng}(I_u) = 6^{-k} - 2\sum_{j=k+1}^{n-1} 6^{-j}$, where $k = k_u = k_v$. Thus, $\text{leng}(I_v) = \text{leng}(I_u) - 2\cdot 6^{-n} = 6^{-k} - 2\sum_{j=k+1}^{n} 6^{-j}$, where $k = k_u = k_v$.

(5) and (6) We prove these properties together by induction on the depth n. Assume that v is of depth n and is of type B. Let u be the parent node of v.

Case 1: n is even. Then, by Lemma 3.4, v is a vertical node.

Subcase 1.1: u is of type A. Then, we know that u is vertical, $\text{leng}(L_u) = \beta_{n-1}$ and $k_v = n$. It follows from the definition that $\text{leng}(L_v) = \beta_n = \beta_n + \delta_{k_v} - \delta_n$.

Subcase 1.2: u is of type C. Then, by the inductive hypothesis, u is vertical and $\text{leng}(L_u) = \beta_n + \delta_{k_u} - \delta_u$. From the definition, $\text{leng}(L_v) = \text{leng}(L_u) = \beta_n + \delta_{k_u} - \delta_n$ because $k_v = k_u$.

Case 2: n is odd. The proof is similar to Case 1, except that v is a horizontal node.

Next, assume that v is of depth n and is of type C. Then its parent u must be of type B. Further assume that n is odd, then both u and v are vertical. From the inductive hypothesis, $\text{leng}(L_u) = \beta_{n-1} + \delta_{k_u} - \delta_{n-1}$. By the definition, $\text{leng}(L_v) = \text{leng}(L_u) - 2\beta_n$ and, by Lemma 3.2(c), it is equal to

$$\beta_{n-1} - 2\beta_n - \delta_{n-1} + \delta_{k_u} = 2\sum_{j=1}^{\infty} \beta_{n+2j} + \delta_{k_u} = \beta_{n+1} + \delta_{k_v} - \delta_{n+1},$$

since $k_v = k_u$. The case when n is even can be verified in a similar way (by Lemma 3.2(a)).

(7) It suffices to show that for each node v, the end point of L_{w_i} of the last child w_i of v coincides with the end point of L_v. This is just a routine check with the definition. For instance, assume that v is of type A and is horizontal and $L_v = \text{line}(\langle x, y\rangle, \langle x + \alpha_n, y\rangle)$. Then, it is easy to check that the end point of L_{w_6} is $\langle x + 2\alpha_{n+1}, y\rangle$. From Lemma 3.4, we know that the depth n of v is even, and so $2\alpha_{n+1} = \alpha_n$. Thus, the end points of L_{w_6} and L_v are identical. We omit the checking of other cases.

Definition 3.6 (*Functions* g_n). For each node v of depth n, we define g_n to be the function that maps I_v linearly to L_v. From property (7) above, we can combine all these g_n together to form a continuous, piecewise linear function g_n from $[0, 1]$ to $\mathbb{R}^2$. We let Γ_n be the curve defined by g_n on $[0, 1]$; that is, $\Gamma_n = g_n([0, 1])$.

We show the functions g_1, g_2 and the first half of g_3 in Fig. 2. From this figure, it is easy to see the pattern of the functions g_n.

Next, we prove that the limit of g_n exists.

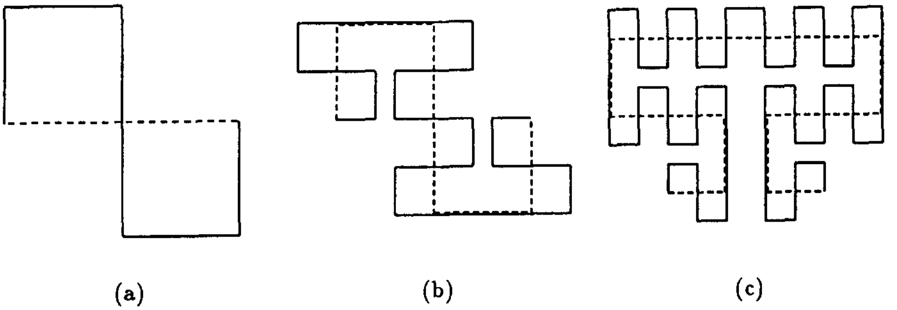

Fig. 2. The functions g_n, $n = 1, 2, 3$, where the dot lines indicate the function g_{n-1}. (a) g_1. (b) g_2. (c) The first half of g_3.

Lemma 3.7. *Let v be a node of depth $n \geqslant 1$ and of type B or type C. Let $\gamma_v = 6^{-(n+1)} \cdot \text{leng}(I_v)^{-1} \cdot \text{leng}(L_v)$. Then, $\gamma_v \leqslant \alpha_{n+1}$ if v is horizontal, and $\gamma_v \leqslant \beta_{n+1}$ if v is vertical.*

Proof. Assume that v is of type B and $k_v = n$. Further assume that v is horizontal. Then from properties (4) and (5), $\text{leng}(I_v) = 6^{-n}$ and $\text{leng}(L_v) = \alpha_n$. It follows that $6^{-(n+1)} \cdot \text{leng}(I_v)^{-1} \cdot \text{leng}(L_v) = 6^{-1} \cdot \alpha_n < \alpha_{n+1}$. Similarly, if $k_v = n$ and v is vertical, then $6^{-(n+1)} \cdot \text{leng}(I_v)^{-1} \cdot \text{leng}(L_v) = 6^{-1} \cdot \beta_n < \beta_{n+1}$.

Now we consider the nodes v with $k_v \leqslant n - 1$. In the following, we let k denote k_v, and observe that

$$\text{leng}(I_v) = 6^{-k} - 2 \cdot \sum_{j=k+1}^{n} 6^{-j} > \tfrac{1}{2} \cdot 6^{-k}.$$

Now we further assume that v is of type B and is horizontal. Then, we have, from Property (5), that $\text{leng}(L_v) < \alpha_n + \delta_k \leqslant \alpha_n + 3 \cdot 5^{-k}$. Thus,

$$\gamma_v = 6^{-(n+1)} \cdot \text{leng}(I_v)^{-1} \cdot \text{leng}(L_v) < 6^{-(n+1)} \cdot 2 \cdot 6^k \cdot (\alpha_n + 3 \cdot 5^{-k})$$

$$\leqslant (18)^{-1} \cdot \alpha_n + 6^{-n} \cdot (\tfrac{6}{5})^k < (18)^{-1} \cdot \alpha_n + 5^{-n}.$$

(In the above, we have used the fact that $k \leqslant n - 1$.) It can be verified by a simple calculation that $(18)^{-1} \cdot \alpha_n + 5^{-n} < \alpha_{n+1}$, and so the lemma follows.

The other cases when v is of type B and is vertical, or when v is of type C, can be proved in a similar way. (Note that if v is of type C then k_v must be less than n.) We omit the details. □

To see that $\{g_n\}$ converges to a continuous function g, we first observe that $\{g_n(t)\}$ converges for all endpoints t of I_v of any node v of depth n.

Lemma 3.8. *If v is a node of depth $n \geqslant 0$, and t is an endpoint of L_v, then $g_n(t) = g_{n+k}(t)$ for all $k > 0$.*

Proof. Immediate from property (7). □

Now we consider the general cases.

Lemma 3.9. *For any $n \geqslant 0$ and any $t \in [0, 1]$, $|g_n(t) - g_{n+1}(t)| \leqslant \alpha_{n+1} + \beta_{n+1}$.*

Proof. From Lemma 3.8, we may assume that $t \in [a_v, b_v)$ for some node v of depth n. We consider three cases.

Case 1: v is of type A. Then, from the definition of g_{n+1}, it is clear that $|g_n(t) - g_{n+1}(t)| \leqslant \alpha_{n+1} + \beta_{n+1}$.

Case 2: v is of type B. Let the three children of v be w_1, w_2 and w_3. Since g_n is linear on $[a_v, b_v]$ and g_{n+1} is linear on $[a_{w_i}, b_{w_i}]$, $i = 1, 2, 3$, and since $g_n(x) = g_{n+1}(x)$ for $x = a_v$ and $x = b_v$, we only need to show that $|g_n(t) - g_{n+1}(t)| \leqslant \alpha_{n+1} + \beta_{n+1}$ for $t = b_{w_1}$ and $t = a_{w_3}$. By the symmetricity of the function g_{n+1} on I_v, we actually only need to verify this for $t = b_{w_1}$.

First assume that v is horizontal. We observe that L_{w_1} is just the initial segment of L_v of length α_{n+1}. Thus, if $g_n(a_v) = g_{n+1}(a_v) = \langle x, y \rangle$, then $g_{n+1}(b_{w_1}) = \langle x + \alpha_{n+1}, y \rangle$. Also, by the linearity of g_n on I_v, we know that $g_n(b_{w_1}) = \langle x + \gamma_v, y \rangle$, where $\gamma_v = \text{leng}(I_{w_1}) \cdot \text{leng}(I_v)^{-1} \cdot \text{leng}(L_v)$. By Lemma 3.7, $\gamma_v < \alpha_{n+1}$, and so $|g_n(b_{w_1}) - g_{n+1}(b_{w_1})| = |\alpha_{n+1} - \gamma_v| \leqslant \alpha_{n+1}$.

The case that v is vertical can be proved in a similar way.

Case 3: v is of type C. Let the three children of v be w_1, w_2 and w_3. Again, like Case 2, we only need to show that $|g_n(t) - g_{n+1}(t)| \leqslant \alpha_{n+1} + \beta_{n+1}$ for $t = b_{w_1}$.

Assume that v is horizontal, and hence n is even. Also assume that $g_n(a_v) = g_{n+1}(a_v) = \langle x, y \rangle$. Then $g_{n+1}(b_{w_1}) = \langle x, y \pm \beta_{n+1} \rangle$, and $g_n(b_{w_1}) = \langle x + \gamma_v, y \rangle$. Therefore, by Lemma 3.7, $|g_n(b_{w_1}) - g_{n+1}(b_{w_1})| \leqslant \gamma_v + \beta_{n+1} < \alpha_{n+1} + \beta_{n+1}$.

The case of the vertical nodes v can be proved in a similar way. □

Theorem 3.10. *The sequence $\{g_n\}$ converges to a continuous function g such that for all $t \in [0, 1]$, $|g(t) - g_n(t)| \leqslant \alpha_n + \beta_n$.*

Proof. This follows immediately from Lemma 3.9:

$$|g(t) - g_n(t)| \leqslant \sum_{j=n}^{\infty} |g_j(t) - g_{j+1}(t)| \leqslant \sum_{j=n}^{\infty} (\alpha_{j+1} + \beta_{j+1}) < \alpha_n + \beta_n.$$

(In the above, the last inequality follows from Lemma 3.2(b) and (d).) □

We let Γ be the curve defined by g on $[0, 1]$; that is, $\Gamma = g([0, 1])$.

4. Properties of the monster curve

4.1. Polynomial-time computability

We first show that $\{g_n\}$ is a polynomial-time computable sequence of functions, in the sense of Definition 2.3. It follows then from Lemma 3.9 that g is also polynomial-time computable.

Lemma 4.1. (a) *If* $a, b \in [0, 1]$ *satisfy* $|a - b| \leqslant 6^{-n} \cdot 2^{-k}$ *for some* $n, k > 0$, *then* $|g_n(a) - g_n(b)| \leqslant 3\beta_n \cdot 2^{-k}$.

(b) *If* $a, b \in [0, 1]$ *satisfy* $|a - b| \leqslant 6^{-n}$ *for some* $n > 0$, *then* $|g(a) - g(b)| \leqslant 7\beta_n$.

Proof. (a) Since g_n is linear on the interval $[i \cdot 6^{-n}, (i + 1)6^{-n}]$, for each $0 \leqslant i \leqslant 6^n - 1$, we only need to show that if $a = i \cdot 6^{-n}$ and $b = (i + 1)6^{-n}$ for some $0 \leqslant i \leqslant 6^n - 1$, then $|g_n(a) - g_n(b)| \leqslant 3\beta_n$. We consider three cases.

Case 1: $[a, b] = I_v$ for some type-A node v of depth n. Then, it is clear from the definition that $|g_n(a) - g_n(b)| \leqslant \max\{\alpha_n, \beta_n\} = \beta_n$.

Case 2: $[a, b] \subseteq I_v$ for some node v that is of depth n and is of type B or type C. Then, $|g_n(a) - g_n(b)| = 6^{-n} \cdot \text{leng}(I_v)^{-1} \cdot \text{leng}(L_v)$. By Lemma 3.7, we see that $|g_n(a) - g_n(b)| \leqslant 6 \cdot \max\{\alpha_{n+1}, \beta_{n+1}\} \leqslant 3\beta_n$.

(b) $|g(a) - g(b)| \leqslant |g(a) - g_n(a)| + |g(b) - g_n(b)| + |g_n(a) - g_n(b)| \leqslant 4\beta_n + 3\beta_n = 7\beta_n$. □

Theorem 4.2. (a) $\{g_n\}$ *is a polynomial-time computable sequence.*

(b) g *is polynomial-time computable.*

Proof. Part (b) follows from part (a) and Theorem 3.10. For part (a), we note that the above lemma showed that $\{g_n\}$ has a uniform polynomial modulus. From Proposition 2.2, and from the linearity of g_n, we only need to show that the discrete function ϕ that maps any pair (n, i), $0 \leqslant i \leqslant 6^n - 1$, to $g_n(i \cdot 6^{-n})$ is computable in time $p(n)$ for some polynomial p. (Note that $\{i \cdot 6^{-n}: 0 \leqslant i \leqslant 6^n - 1\}$ includes all endpoints of I_v for all nodes v of depth n.) But this follows directly from the definitions of I_v and L_v. Namely, to find $g_n(r)$ where $r = i \cdot 6^{-n}$ for some i, we recursively determine the node v_k of depth k with $r \in I_k$ and compute I_{v_k} and L_{v_k}, according to $I_{v_{k-1}}$, $L_{v_{k-1}}$ and label(v_{k-1}). Apparently, this is a polynomial-time procedure. □

4.2. One-to-oneness

To establish the one-to-oneness of functions g_n and function g, we need a more precise estimation of the distance between two line segments L_{v_1} and L_{v_2}. For any horizontal line segment L with the endpoints $\langle a_1, b\rangle$ and $\langle a_2, b\rangle$, $a_1 < a_2$, let

$$\text{box}(L, \sigma, \tau) = \{\langle x, y\rangle: a_1 - \sigma \leqslant x \leqslant a_2 + \sigma, |y - b| \leqslant \tau\}$$

and for any vertical line segment L with the endpoints $\langle a, b_1\rangle$ and $\langle a, b_2\rangle$, $b_1 < b_2$, let

$$\mathrm{box}(L, \sigma, \tau) = \{\langle x, y\rangle : |x - a| \leqslant \sigma, b_1 - \tau \leqslant y \leqslant b_2 + \tau\}.$$

Lemma 4.3. *Let v be a node of depth n.*

(a) *If n is odd and v is horizontal, or if n is even and v is vertical (and so v is of type B), then $g_{n+1}[I_v] \subseteq \mathrm{box}(L_v, 0, 0)$.*

(b) *If n is even and v is horizontal, then $g_{n+1}[I_v] \subseteq \mathrm{box}(L_v, 0, \beta_{n+1})$.*

(c) *If n is odd and v is vertical, then $g_{n+1}[I_v] \subseteq \mathrm{box}(L_v, \alpha_{n+1}, 0)$.*

Proof. Immediate from Lemma 3.4 and the definition of g_{n+1}. □

Fof each integer $n > 0$, define

$$\sigma_n = \sum_{2j \geqslant n+1} \alpha_{2j}, \qquad \tau_n = \sum_{2j+1 \geqslant n+1} \beta_{2j+1},$$

and let $\mathrm{Box}(v) = \mathrm{box}(L_v, \sigma_n, \tau_n)$, if v is of depth n.

Lemma 4.4. *If $n > 0$ is even, then $2\sigma_n = \alpha_{n+1} - \delta_{n+1}$ and $2\tau_n = \beta_n - \delta_n$. If $n > 0$ is odd, then $2\sigma_n = \alpha_n - \delta_n$ and $2\tau_n = \beta_{n+1} - \delta_{n+1}$.*

Proof. It follows immediately from Lemma 3.2. □

Lemma 4.5. *Let v be a node of depth n. Then, $g[I_v] \subseteq \mathrm{Box}(v)$.*

Proof. Let $\sigma_{n,k} = \sum_{n+1 \leqslant 2j \leqslant n+k} \alpha_{2j}$ and $\tau_{n,k} = \sum_{n+1 \leqslant 2j+1 \leqslant n+k} \beta_{2j+1}$. We prove by induction that for all $k \geqslant 1$, $g_{n+k}[I_v] \subseteq \mathrm{box}(L_v, \sigma_{n,k}, \tau_{n,k})$. First observe that the case $k = 1$ has been proved in Lemma 4.3.

Next, assume that $g_{n+k}[I_v] \subseteq \mathrm{box}(L_v, \sigma_{n,k}, \tau_{n,k})$, and consider $g_{n+k+1}[I_w]$ for all depth-$(n+k)$ descendants w of v. We consider 3 cases.

(1) If $n + k$ is odd and w is horizontal, or if $n + k$ is even and w is vertical then, from Lemma 4.3(a), $g_{n+k+1}[I_w] \subseteq \mathrm{box}(L_w, 0, 0) = L_w = g_{n+k}[I_w] \subseteq \mathrm{box}(L_v, \sigma_{n,k}, \tau_{n,k}) \subseteq \mathrm{box}\ (L_v, \sigma_{n,k+1}, \tau_{n,k+1})$.

(2) If $n + k$ is even and w is horizontal, then, from Lemma 4.3(b), $g_{n+k+1}[I_w] \subseteq \mathrm{box}(L_w, 0, \beta_{n+k+1})$ and hence it is contained in $\mathrm{box}(L_v, \sigma_{n,k}, \tau_{n,k} + \beta_{n+k+1}) = \mathrm{box}(L_v, \sigma_{n,k+1}, \tau_{n,k+1})$.

(3) If $n + k$ is odd and w is vertical, then, from Lemma 4.3(c), $g_{n+k+1}[I_w] \subseteq \mathrm{box}(L_w, \alpha_{n+k+1}, 0) \subseteq \mathrm{box}(L_v, \sigma_{n,k} + \alpha_{n+k+1}, \tau_{n,k}) = (L_v, \sigma_{n,k+1}, \tau_{n,k+1})$. It follows that $g[I_v] \subseteq \mathrm{Box}(v)$. □

Lemma 4.6. *If w is a child of v, then $\mathrm{Box}(w) \subseteq \mathrm{Box}(v)$.*

Proof. Again, we check this by 3 cases. Assume that v is of depth n.

(1) If n is odd and v is horizontal, or if n is even and v is vertical, then $L_w \subseteq \mathrm{box}(L_v, 0, 0) = L_v$. Furthermore, it is clear that $\sigma_{n+1} \leqslant \sigma_n$ and $\tau_{n+1} \leqslant \tau_n$, and it follows that $\mathrm{Box}(w) \subseteq \mathrm{Box}(v)$.

(2) If n is even and v is horizontal, then $L_w \subseteq \mathrm{box}(L_v, 0, \beta_{n+1})$. Furthermore, we can check that $\sigma_{n+1} = \sigma_n$ and $\tau_{n+1} = \tau_n - \beta_{n+1}$, and it follows that $\mathrm{Box}(w) \subseteq \mathrm{Box}(v)$.

(3) If n is odd and v is vertical, then $L_w \subseteq \mathrm{box}(L_v, \alpha_{n+1}, 0)$. Furthermore, we can check that $\sigma_{n+1} = \sigma_n - \alpha_{n+1}$ and $\tau_{n+1} = \tau_n$, and it follows that $\mathrm{Box}(w) \subseteq \mathrm{Box}(v)$ $\square$

Lemma 4.7. *Assume that* v_1 and v_2 *are two nodes of depth n.*
(a) *If* v_1 *and* v_2 *are not neighbors, then* $\mathrm{Box}(v_1) \cap \mathrm{Box}(v_2) = \emptyset$.
(b) *If* v_1 *and* v_2 *are neighbors, then* $\mathrm{Box}(v_1) \cap \mathrm{Box}(v_2)$ *has measure* $4\sigma_n\tau_n$.

Proof. (a) We prove this by induction on n. First, if $n = 0$, then there is only one node. Assume that $n > 1$ and the parents of v_1 and v_2 are u_1 and u_2, respectively.

Case 1: u_1 and u_2 are not neighbors. Then, by the inductive hypothesis, $\mathrm{Box}(u_1) \cap \mathrm{Box}(u_2) = \emptyset$. Since, by Lemma 4.6, $\mathrm{Box}(v_i) \subseteq \mathrm{Box}(u_i)$, for $i = 1, 2$, it follows that $\mathrm{Box}(v_1) \cap \mathrm{Box}(v_2) = \emptyset$.

Case 2. $u_1 = u_2$. We consider three more subcases.

Subcase 2.1: u_1 is of type A. Then, it is clear that L_{v_1} and L_{v_2} have either a horizontal distance $\geqslant \alpha_n$ or a vertical distance $\geqslant \beta_n$. We observed from Lemma 4.4 that $\alpha_n > 2\sigma_n$ and $\beta_n > 2\tau_n$, and so the distance between the two line segments L_{v_1} and L_{v_2} are big enough to prevent $\mathrm{Box}(v_1)$ from meeting $\mathrm{Box}(v_2)$.

Subcase 2.2: u_1 is of type B. First assume that u_1 is horizontal, and hence n is even. Then, v_1 and v_2 must be both of type A and L_{v_1} and L_{v_2} must be subsegments of L_{u_1} and have distance $d = \alpha_{n+1} + \delta_k - \delta_{n+1} = 2\sigma_n + \delta_k$ (Lemma 4.4), where $k = k_{u_1} \leqslant n - 1$. Therefore, $\mathrm{Box}(v_1) \cap \mathrm{Box}(v_2) = \emptyset$.

The same argument works for the case of the vertical u_1. More precisely, the distance between L_{v_1} and l_{v_2} is $d = \beta_{n+1} + \delta_k - \delta_{n+1} = 2\tau_n + \delta_k$.

Subcase 2.3: u_1 is of type C. This case is similar to Subcase 2.2. We omit the details of the verification.

Case 3: u_1 and u_2 are neighbors. Then, we claim that the distance between L_{v_1} and L_{v_2} is, like Subcase 2.1, at least α_n in the horizontal direction or β_n in the vertical direction. From this claim, the lemma can be proved as in Subcase 2.1.

The claim can be verified by inspection in the following subcases:

Subcase 3.1: Both u_1 and u_2 are of type A. Then, we can check, from the definition of the tree T, that u_1 and u_2 must have the same label, and hence must lie in the same horizontal or vertical line. Therefore, the relation between L_{v_1} and L_{v_2} is like that in Subcase 2.1.

Subcase 3.2: One of u_1 or u_2 is of type B or C. Then, u_1 and u_2 must have a common parent node w. By expanding the subtree of T rooted at w, we can check that the claim holds for L_{v_1} and L_{v_2}.

(b) The line segments L_{v_1} and L_{v_2} meet at one endpoint $\langle x_0, y_0 \rangle$ and are either both on the same line or are perpendicular to each other. In either case, the intersection of $\mathrm{Box}(v_1)$ and $\mathrm{Box}(v_2)$ is the rectangle $R = \{\langle x, y \rangle \colon |x - x_0| \leqslant \sigma_n, |y - y_0| \leqslant \tau_n\}$, whose measure is $4\sigma_n\tau_n$. $\square$

Theorem 4.8. *Functions g_n and function g are one-to-one functions.*

Proof. First consider g_n, $n \geqslant 0$. Let $t_1 \neq t_2$ be two points in [0, 1]. Let v_1 and v_2 be the two depth-n nodes such that $t_1 \in I_{v_1}$ and $t_2 \in I_{v_2}$.

Case 1: $v_1 = v_2$. Then g_n is linear on I_{v_1} so $g_n(t_1) \neq g_n(t_2)$.

Case 2: $v_1 \neq v_2$ but they are neighbors. Then, g_n maps $I_{v_1} \cup I_{v_2}$ to two consecutive line segments L_{v_1} and L_{v_2}. Note from the definition of tree T, $\text{label}_1(v_1)$ and $\text{label}_1(v_2)$ are never complementary if v_1 and v_2 are neighbors. Thus, L_{v_1} and L_{v_2} meet only at one endpoint. So, $g_n(t_1) \neq g_n(t_2)$.

Case 3: $v_1 \neq v_2$ and they are not neighbors. Then, by Lemma 4.7, $\text{Box}(v_1) \cap \text{Box}(v_2) = \emptyset$, and so $g_n(t_1) \neq g_n(t_2)$.

Next, we consider function g. Let $t_1 \neq t_2$ be two points in [0, 1]. For each $n \geqslant 0$, let $v_{n,1}$ and $v_{n,2}$ be the depth-n nodes such that $t_1 \in I_{v_{n,1}}$ and $t_2 \in I_{v_{n,2}}$.

Case 1: There exists an integer n such that $v_{n,1}$ and $v_{n,2}$ are distinct and are not neighbors. Then, by Lemma 4.7, $\text{Box}(v_{n,1}) \cap \text{Box}(v_{n,2}) = \emptyset$, and so $g(t_1) \neq g(t_2)$.

Case 2: For every integer n, $v_{n,1} = v_{n,2}$. Let $d = |t_1 - t_2|$ and $v = v_{n,1}$. Then, $\text{leng}(I_{v_n}) \geqslant d$ for all $n \geqslant 0$. So, for sufficiently large n, v_n must be of type B or type C. Let the highest type-A node containing t_1 and t_2 be v_k. That is, for all $n \geqslant k+1$, $k_{v_n} = k+1$. Then, we can see that $\text{leng}(I_{v_n}) \leqslant 6^{-k}$ and $\text{leng}(L_{v_n}) \geqslant \delta_{k+1}$ for all $n \geqslant k+1$. Thus, for each $n \geqslant k+1$, $|g_n(t_1) - g_n(t_2)| = d \cdot \text{leng}(I_{v_n})^{-1} \cdot \text{leng}(L_{v_n}) \geqslant d \cdot 6^k \cdot \delta_{k+1} > 0$. Thus, $|g(t_1) - g(t_2)| \geqslant d \cdot 6^{-k} \cdot \delta_{k+1} > 0$.

Finally, we claim that the above 2 cases are exhaustive, and hence the theorem is proven. To prove the claim, we assume, by way of contradiction, that neither case holds. Then, it must be true that for sufficiently large n, $v_{n,1} \neq v_{n,2}$ and they are neighbors. It is easy to observe that it also must be true that for sufficiently large n, both $v_{n,1}$ and $v_{n,2}$ are of type A, because the two neighboring children of two neighboring nodes must be of type A. But this is not possible, because then the distance between t_1 and t_2 is smaller than $\text{leng}(I_{v_{n,1}}) + \text{leng}(I_{v_{n,2}}) = 2 \cdot 6^{-n}$ for all n, and must be 0. □

4.3. *Length of the curves*

Recall that Γ_n is the curve defined by function g_n in Section 3. In this subsection, we show that the sequence $\{l_n\}$ of the length of Γ_n is a polynomial-time computable sequence. To prove this result, we define the following values about the tree T:

a_n = the number of type-A nodes v at depth n, $n \geqslant 0$;

b_n = the number of type-B nodes v at depth n, $n \geqslant 1$;

c_n = the number of type-C nodes v at depth n, $n \geqslant 1$;

$b_{n,i}$ = the number of type-B nodes v at depth n with $k_v = i$, $1 \leqslant i \leqslant n$;

$c_{n,i}$ = the number of type-C nodes v at depth n with $k_v = i$, $1 \leqslant i \leqslant n$.

Then, we observe from the definition of g_n that if $n \geqslant 1$ is even then

$$l_n = a_n \cdot \alpha_n + b_n(\beta_n - \delta_n) + c_n(\alpha_{n+1} - \delta_{n-1}) + \sum_{i=1}^{n} (b_{n,i} + c_{n,i})\delta_i, \tag{1}$$

and if $n \geqslant 1$ is odd then

$$l_n = a_n \cdot \beta_n + b_n(\alpha_n - \delta_n) + c_n(\beta_{n+1} - \delta_{n-1} + \sum_{i=1}^{n} (b_{n,i} + c_{n,i})\delta_i. \tag{2}$$

Therefore, the sequence $\{l_n\}$ is a polynomial-time computable sequence as long as the sequences $\{a_n\}$, $\{b_n\}$, $\{c_n\}$, $\{b_{n,i}\}$ and $\{c_{n,i}\}$ are polynomial-time computable.

Lemma 4.9. *The following holds for all $n \geqslant 1$ and $1 \leqslant i \leqslant n$:*

(a) $a_n = 4 \cdot 5^{n-1}$.

(b) $b_n = \begin{cases} (5^n - 1)/3 & \text{if } n \text{ is even}, \\ (5^n + 1)/3 & \text{if } n \text{ is odd}. \end{cases}$

(c) $c_n = \begin{cases} (5^{n-1} - 1)/3 & \text{if } n \text{ is odd}, \\ (5^{n-1} + 1)/3 & \text{if } n \text{ is even}. \end{cases}$

(d) $b_{n,i} = \begin{cases} 0 & \text{if } n-i \text{ is odd}, \\ 2a_{i-1} & \text{if } n-i \text{ is even}. \end{cases}$

(e) $c_{n,i} = \begin{cases} 0 & \text{if } n-i \text{ is even}, \\ 2a_{i-1} & \text{if } n-i \text{ is odd}. \end{cases}$

Proof. We can prove these relations together by induction on n. First, we observe that

$$a_0 = 1, \quad a_1 = 4, \quad a_2 = 20;$$

$$b_1 = b_{1,1} = 2, \quad b_2 = b_{2,2} = 8, \quad b_{2,1} = 0;$$

$$c_1 = c_{1,1} = 0, \quad c_2 = c_{2,1} = 2, \quad c_{2,2} = 0.$$

Thus the above relations (a)–(e) hold for $n = 1, 2$ and $i = 1, 2$.

Now let $k \geqslant 2$ and assume that the relations (a)–(e) hold for all $n \leqslant k$ and $1 \leqslant i \leqslant k$, and consider the case $n = k + 1$.

(a) We observe that each depth-k, type-A node has 4 type-A children and each depth-k, type-B or type-C node has 2 type-A children, and there is no other way to produce type-A nodes at depth $k+1$. Therefore, we have $a_{k+1} = 4 \cdot a_k + 2(b_k + c_k) = 4 \cdot 5^k$.

(b) We observe that each depth-k, type-A node has 2 type-B children and each depth-k, type-C node has 1 type-B child, and there is no other way to produce type-B nodes at depth $k+1$. Thus, $b_{k+1} = 2 \cdot a_k + c_k = 2 \cdot 4 \cdot 5^{k-1} + (5^{k-1} \pm 1)/3 = (5^{k+1} \pm 1)/3$, where $\pm$ is $+$ if k is even and $\pm$ is $-$ if k is odd.

(c) We observe that each depth-k, type-B node has 1 type-C child, and there is no other way to produce type-C nodes at depth $k+1$. Thus, $c_{k+1} = b_k = (5^k \pm 1)/3$, where $\pm$ is $+$ if k is odd and $\pm$ is $-$ if k is even.

(d) Assume that $i < k + 1$. Then, each depth-$(k+1)$, type-B node v with $k_v = i$ is a child of depth-k, type-C node u with $k_u = i$. Therefore $b_{k+1,i} = c_{k,i} = 0$ if $k - i$ is even, and $b_{k+1,i} = c_{k,i} = 2a_{i-1}$ if $k - i$ is odd.

If $i = k + 1$, then each depth-$(k + 1)$, type-B node v with $k_v = k + 1$ is a child of a depth-k, type-A node. There are 2 such children for each depth-k, type-A node. Therefore, $b_{k+1,k+1} = 2a_k$.

(e) Similar to (d). □

The exact length l_n of the curve Γ_n can now be computed from the above lemma. The following calculation is to be used in the next section.

Let H_n be the set of all horizontal nodes v in T that have depth n, and V_n be the set of all vertical nodes v in T that have depth n. Also let $l_n^{(h)}$ be the total length of all line segments L_v with $v \in H_n$ and $l_n^{(v)}$ be the total length of all line segments L_v with $v \in V_n$.

Lemma 4.10. (a) *For any even* $n > 0$, $l_n^{(h)} = \frac{5}{3}5^{n/2} + n/6 + O(1)$, *and* $l_n^{(v)} = \frac{5}{6}5^{n/2} + 5n/12 + O(1)$.

(b) *For any odd* $n > 0$, $l_n^{(h)} = \frac{1}{3}5^{(n+1)/2} + n/6 + O(1)$ *and* $l_n^{(v)} = \frac{5}{6}5^{(n+1)/2} + 5n/12 + O(1)$.

Proof. (a) From Lemma 3.4 and formula (1), we know that

$$\begin{aligned}
l_n^{(h)} &= a_n\alpha_n + c_n(\alpha_{n+1} - \delta_{n+1}) + \sum_{i=1}^{n} c_{n,i}\delta_i \\
&= a_n\alpha_n + c_n(\alpha_{n+1} - \delta_{n+1}) + \sum_{1 \le i \le n,\, i\,\mathrm{odd}} 2a_{i-1}\delta_i \\
&= 4\cdot 5^{n-1}(2\cdot 5^{-n/2} + \tfrac{1}{2}5^{-n}) + \frac{5^{n-1}+1}{3}(5^{-n/2} + \tfrac{1}{4}5^{-n} - \tfrac{5}{24}5^{-n}) \\
&\quad + \sum_{1 \le i \le n,\, i\,\mathrm{odd}} 8\cdot 5^{i-2}\cdot\tfrac{1}{24}\cdot 5^{-(i-2)} \\
&= \tfrac{8}{5}5^{n/2} + \tfrac{2}{5} + \tfrac{1}{15}5^{n/2} + c_n + \tfrac{1}{6}n \\
&= \tfrac{5}{3}5^{n/2} + \tfrac{1}{6}n + c_n',
\end{aligned}$$

where $0 < c_n < \frac{1}{2}$ and $0 < c_n' < 1$. Also,

$$\begin{aligned}
l_n^{(v)} &= b_n(\beta_n - \delta_n) + \sum_{i=1}^{n} b_{n,i}\delta_i = b_n(\beta_n - \delta_n) + \sum_{1 \le i \le n,\, i\,\mathrm{even}} 2a_{i-1}\delta_i \\
&= \frac{5^n - 1}{3}(\tfrac{1}{2}5^{-(n/2-1)} + \tfrac{1}{8}5^{-(n-2)} - \tfrac{1}{48}5^{-(n-3)}) \\
&\quad + \sum_{1 \le i \le n,\, i\,\mathrm{even}} 8\cdot 5^{i-2}\cdot\tfrac{1}{48}\cdot 5^{-(i-3)} \\
&= \tfrac{5}{6}5^{n/2} + \tfrac{5}{12}n + d_n,
\end{aligned}$$

where $0 < d_n < 1$.

(b) The calculation is similar, we omit the details. Or, alternatively, we notice that $l_n^{(v)} = l_{n+1}^{(v)}$ if n is odd, and $l_n^{(h)} = l_{n+1}^{(h)}$ if n is even. For instance, we observe that each horizontal node v of type A has exactly two horizontal child nodes w_1, w_2 of type B with $\mathrm{leng}(L_{w_1}) = \mathrm{leng}(L_{w_2}) = \mathrm{leng}(L_v)/2$. Also, each horizontal node v of type C has exactly one horizontal child node w of type B such that $\mathrm{leng}(L_w) = \mathrm{leng}(L_v)$. Therefore, for each even $n \geqslant 0$, the total length of horizontal line segments at depth $n+1$ is equal to that at depth n. Similar observations show that for each odd $n \geqslant 1$, the total length of vertical line segments at depth $n+1$ is equal to that of depth n. □

Corollary 4.11. *Let $n \geqslant 1$ and let l_n be the length of the curve Γ_n. Then,* $5^{n/2} < l_n < 5^{n/2+1}$.

Proof. We note that, for even $n > 0$, the term $O(1)$ in Lemma 4.10(a) is actually bounded by 1. Thus, $l_n = l_n^{(h)} + l_n^{(n)}$ is very close to $\frac{5}{2} 5^{n/2}$ and it is easy to check that the above inequality holds even for small n. □

Corollary 4.12. *The sequence $\{l_n\}$ of the lengths of the curve Γ_n is a polynomial-time computable sequence.*

4.4. Measure of the monster curve

For any set $S \subseteq \mathbb{R}^2$, we let $\mu^*(S)$ denote its outer measure in $\mathbb{R}^2$, and let $\mu(S)$ denote its Lebesgue measure if S is measurable. Recall that Γ is the curve computed by function g in Section 3. Since Γ is a closed set in $\mathbb{R}^2$, it is measurable. In this subsection, we show that $\mu(\Gamma) = 5$. We first introduce the following notion of *Minkowski measure*. For any point $\mathbf{z} \in \mathbb{R}^2$ and any $\varepsilon > 0$, we let $\mathrm{N}(\mathbf{z}; \varepsilon)$ be the open neighborhood of center $\mathbf{z}$ and radius ε. For any set $S \subseteq \mathbb{R}^2$ and any $\varepsilon > 0$, we let $\mathrm{N}(S; \varepsilon) = \bigcup_{\mathbf{z} \in S} \mathrm{N}(\mathbf{z}; \varepsilon)$ (called a *Minkowski sausage* in [6]). Note that for any set $S \subseteq \mathbb{R}^2$, $\mathrm{N}(S; \varepsilon_1) \subseteq \mathrm{N}(S; \varepsilon_1) \subseteq \mathrm{N}(S; \varepsilon_2)$ if $\varepsilon_1 < \varepsilon_2$.

Definition 4.13. For any set $S \subseteq \mathbb{R}^2$, the Minkowski measure of S is $m(S) = \inf_{\varepsilon > 0} \mu^*(\mathrm{N}(S; \varepsilon))$.

The Minkowski measure $m(S)$ of a set S is at least as large as the outer measure $\mu^*(S)$, but it is not necessarily equal to $\mu^*(S)$. However, the two quantities are equal if S is bounded and closed.

Proposition 4.14. *If $S \subseteq \mathbb{R}^2$ is bounded and closed, then $m(S) = \mu(S)$.*

Proof. It is clear from the definition that $m(S) \geqslant \mu(S)$. Assume, by way of contradiction, that $m(S) > \mu(S)$. Let $\delta = m(S) - \mu(S)$. Choose an open set Q that contains S and has $\mu(Q) < \mu(S) + \delta/2$. Then, for each $\mathbf{z} \in S$, there is an $\varepsilon_{\mathbf{z}} > 0$ such that $\mathrm{N}(\mathbf{z}; \varepsilon_{\mathbf{z}}) \subseteq Q$. The collection $\{\mathrm{N}(\mathbf{z}; \varepsilon_{\mathbf{z}})\}_{\mathbf{z} \in S}$ forms an open covering for S. Since S is bounded and closed, there is a finite subcovering of S: $\{N(\mathbf{z}_i; \varepsilon_{\mathbf{z}_i})\}_{i=1}^k$. Note that $\mu(\bigcup_{i=1}^k N(\mathbf{z}_i; \varepsilon_{\mathbf{z}_i})) \leqslant \mu(Q) < \mu(S) + \delta/2$.

Note that each $N(\mathbf{z}_i; \varepsilon_{\mathbf{z}_i})$ has a finite boundary. Let ℓ be the sum of the length of the boundaries of these neighborhoods. Choose an $\varepsilon > 0$ such that $\ell \cdot \varepsilon < \delta/2$ and $\varepsilon < \varepsilon_{\mathbf{z}_i}$ for all i, $1 \leqslant i \leqslant k$. Then, consider the set $T = \bigcup_{i=1}^k N(\mathbf{z}_i; \varepsilon_{\mathbf{z}_i} + \varepsilon)$. We observe that for any $\mathbf{z} \in S$, $\mathbf{z} \in N(\mathbf{z}_i; \varepsilon_{\mathbf{z}_i})$ for some i, $1 \leqslant i \leqslant k$, and so $N(\mathbf{z}; \varepsilon) \subseteq N(\mathbf{z}_i; \varepsilon_{\mathbf{z}_i} + \varepsilon) \subseteq T$. Thus, $\mu(T) \geqslant m(S)$.

On the other hand, we see that $\varepsilon \cdot \ell < \delta/2$ implies that

$$\mu(T) \leqslant \mu\left(\bigcup_{i=1}^k N(\mathbf{z}_i; \varepsilon_{\mathbf{z}_i})\right) + \varepsilon \cdot \ell < \mu(S) + \frac{\delta}{2} + \frac{\delta}{2} = m(S).$$

This is a contradiction. □

Recall that H_n (and V_n) denotes the set of all horizontal (and, respectively, vertical) nodes in tree T that are of depth n. We consider the sets $B_n = \bigcup_{v \in H_n \cup V_n} \mathrm{Box}(v)$. We note that each B_n is a closed set and so is measurable. From Lemmas 4.6 and 4.5, we know that $B_1 \supseteq B_2 \supseteq \cdots \supseteq \Gamma$. Therefore, $\{\mu(B_n)\}$ is a nondecreasing sequence of real numbers each greater than $\mu(\Gamma)$. It implies that $\lim_{n\to\infty} \mu(B_n)$ exists and is $\geqslant \mu(\Gamma)$. The following lemma shows that this limit is actually equal to $\mu(\Gamma)$, and it allows us to calculate $\mu(\Gamma)$ easily.

Lemma 4.15. $\lim_{n\to\infty} \mu(B_n) = \mu(\Gamma)$.

Proof. Let $b = \lim_{n\to\infty} \mu(B_n)$. We have already seen that $b \geqslant \mu(\Gamma)$. Assume, by way of contradiction, that $\mu(\Gamma) = b - \delta$ for some $\delta > 0$. Then, by Proposition 4.14, $m(\Gamma) = b - \delta$. That is, there exists an $\varepsilon > 0$ such that $\mu(N(\Gamma; \varepsilon)) \leqslant b - \delta/2$. Choose an integer $n > 0$ such that $\alpha_n + \beta_n + \sigma_n + \tau_n \leqslant \varepsilon$. Note that for each point $\mathbf{z} \in B_n$, there is a point $g_n(t)$ in Γ_n such that $|\mathbf{z} - g_n(t)| \leqslant \sigma_n + \tau_n$. It follows that there is a point $g(t)$ in Γ such that

$$|\mathbf{z} - g(t)| \leqslant |\mathbf{z} - g_n(t)| + |g(t) - g_n(t)| \leqslant \sigma_n + \tau_n + \alpha_n + \beta_n \leqslant \varepsilon,$$

where the second inequality follows from Theorem 3.10. This implies that $B_n \subseteq N(\Gamma; \varepsilon)$ and so leads to a contradiction: $\mu(B_n) \leqslant b - \delta/2$. Therefore, we may conclude that $\mu(\Gamma) = b$. □

Theorem 4.16. $\mu(\Gamma) = 5$.

Proof. Recall that $l_n^{(h)} = \sum_{v \in H_n} \mathrm{leng}(L_v)$ and $l_n^{(v)} = \sum_{v \in V_n} \mathrm{leng}(L_v)$. We claim that $\mu(B_n) = 2\tau_n l_n^{(h)} + 2\sigma_n l_n^{(v)} + 4\sigma_n\tau_n$. The proof of the claim is as follows: For each $v \in H_n$, $\mu(\mathrm{Box}(v)) = 2\tau_n \cdot \mathrm{leng}(L_v) + 4\sigma_v\tau_n$, and for each $v \in V_n$, $\mu(\mathrm{Box}(v)) = 2\sigma_n \cdot \mathrm{leng}(L_v) + 4\sigma_n\tau_n$. Note from Lemma 4.7 that two neighboring boxes $\mathrm{Box}(u)$ and $\mathrm{Box}(v)$ have an intersection of size exactly $4\sigma_n\tau_n$, and two nonneighboring boxes have an empty intersection. Thus,

$$\begin{aligned}\mu(B_n) &= \sum_{v\in H_n} \mu(\mathrm{Box}(v)) + \sum_{v\in V_n} \mu(\mathrm{Box}(v)) - 4\sigma_n\tau_n(|H_n| + |V_n| - 1)\\ &= \sum_{v\in H_n} 2\tau_n\cdot \mathrm{leng}(L_v) + \sum_{v\in V_n} 2\sigma_n\cdot \mathrm{leng}(L_v) + 4\sigma_n\tau_n\\ &= 2\tau_n l_n^{(h)} + 2\sigma_n l_n^{(h)} + 4\sigma_n\tau_n,\end{aligned}$$

where $|H_n|$ and $|V_n|$ denote the sizes of H_n and V_n, respectively.

Thus, from Lemmas 4.10 and 4.4, we have, for each even $n > 0$,

$$\begin{aligned}\mu(B_n) &= (\beta_n - \delta_n)l_n^{(h)} + (\alpha_{n+1} - \delta_{n+1})\, l_n^{(v)} + 4\sigma_n\tau_n\\ &= (\tfrac{5}{2}\, 5^{-n/2} + \tfrac{25}{48}\, 5^{-n})(\tfrac{5}{3}\, 5^{n/2} + \tfrac{1}{6}n + \mathrm{O}(1))\\ &\quad + (5^{-n/2} + \tfrac{1}{24}\, 5^{-n})(\tfrac{5}{6}\, 5^{n/2} + \tfrac{5}{12}n + \mathrm{O}(1))\\ &= 5 + \mathrm{O}(n\cdot 5^{-n/2}).\end{aligned}$$

If follows that $\mu(\Gamma) = \lim_{n\to\infty} \mu(B_n) = 5$. □

5. The area problem

In this section, we construct a polynomial-time computable function $f\colon [0,1] \to \mathbb{R}^2$ defining a simple, closed curve Γ_f such that the measure of the interior S of Γ_f is nonrecursive. We first give an overview of the idea of the construction. Let Γ_{sq} be the boundary of the square $[0,1]\times[-1,0]$. The curve Γ_f is to be obtained from Γ_{sq} by substituting a simple curve Λ_n for a line segment $L_n \subseteq [0,1]\times\{0\}$, where L_n's are pairwisely disjoint. Pick a recursively enumerable but nonrecursive set K. The curves Λ_n are defined in such a way that if $n\in K$, then Λ_n is of measure 0 and is symmetric with respect to its center and so its substitution for L_n does not change the measure of the interior of Γ_{sq}. On the other hand, if $n \notin K$, then Λ_n is a scaled-down image of the monster curve Γ that has a measure $c\cdot 2^{-2n}$ for some rational constant $c > 0$. Since Γ and hence Λ_n is symmetric with respect to its center, its substitution for L_n decreases the measure of Γ_{sq} by $(\frac{1}{2})\cdot c\cdot 2^{-2n}$. Together, the change from Γ_{sq} to Γ_f then decreases the measure by $c\cdot\sum_{n\notin K} 2^{-(2n+1)}$, which is a nonrecursive real number.

We first extend the functions g_n and g to the domain $[-\frac{1}{2}, \frac{3}{2}]$: g_n and g map $[-\frac{1}{2}, 0]$ linearly to the line segment $L_0 = \mathrm{line}(\langle -2, 0\rangle, \langle -\alpha_1, 0\rangle)$, and they map $[1, \frac{3}{2}]$ linearly to the line segment $L_1 = \mathrm{line}(\langle \alpha_1, 0\rangle, \langle 2, 0\rangle)$. We keep the names Γ_n and Γ for the curves defined by g_n and g, respectively, on $[-\frac{1}{2}, \frac{3}{2}]$. We observe that these extended curves are still one-to-one.

Theorem 5.1. (a) *Functions g_n, $n \geqslant 1$, and g are one-to-one on* $[-\frac{1}{2}, \frac{3}{2}]$.

(b) *The ranges of functions g_n and g on domain* $(-\frac{1}{2}, \frac{3}{2})$ *are contained in the interior of the square* $[-2, 2] \times [-2, 2]$.

Proof. (a) It is easy to verify that g_1 is one-to-one on $[-\frac{1}{2}, \frac{3}{2}]$. For $n \geqslant 2$, consider a node v of depth n. If v is not the leftmost node of depth n, then L_v and L_0 have either a horizontal distance $> \alpha_n$ or a vertical distance $\geqslant \beta_n$. Thus, by the same argument as in Lemma 4.7, $\mathrm{Box}(v) \cap L_0 = \emptyset$, and hence g_n on I_v does not meet L_0. If v is the leftmost node of depth n, we can show, by a simple induction, that g_n on I_v meets L_0 only at the point $\langle -\alpha_1, 0\rangle$. The relation between g_n and L_1 is similar. This shows that g_n is one-to-one on $[-\frac{1}{2}, \frac{3}{2}]$.

For the function g, we have just proved that for any $k \geqslant 2$, $\mathrm{Box}(v) \cap L_0 = \emptyset$ for all but the first node v of depth k. It follows that if $t \in (0, 1]$ then $g(t) \notin L_0$. The relation between g and L_1 is similar. We conclude that g is one-to-one on $[-\frac{1}{2}, \frac{3}{2}]$.

(b) The curves defined by g_n and g on $[0, 1]$ are contained in $\mathrm{Box}(v_0)$, where v_0 is the root of the tree T. We observe that $\sigma_{v_0} = \sum_{j=1}^{\infty} \alpha_{2j} < \frac{3}{4} = 2 - \alpha_1$, and $\tau_{v_0} = \sum_{j=1}^{\infty} \beta_{2j+1} < 2$. Thus, $\mathrm{Box}(v_0)$ is contained in the interior of the square $[-2, 2] \times [-2, 2]$. □

Now we are ready to define the function f. Let K be a recursively enumerable but nonrecursive set of integers. Let M be a Turing machine that recognizes set K; that is, $M(n)$ halts if and only if $n \in K$. For each $n \in K$, let $T(n)$ be the number of moves made by M before it halts on input n. Without loss of generality, assume that $0 \notin K$ and that $T(n) \geqslant n + 2$ for all $n \in K$.

For each $n \geqslant 1$, we define two linear mappings $G_n : \mathbb{R} \to \mathbb{R}$ and $F_n : \mathbb{R}^2 \to \mathbb{R}^2$ as follows:

$$G_n(t) = 2^{n+2}t - \tfrac{5}{2},$$

$$F_n(\langle x, y\rangle) = \langle 2^{-(n+3)}x + 3 \cdot 2^{-(n+2)}, 2^{-(n+3)}y\rangle.$$

Note that G_n maps the interval $[2^{-(n+1)}, 2^{-n}]$ to the interval $[-\frac{1}{2}, \frac{3}{2}]$ and F_n maps the square $[-2, 2] \times [-2, 2]$ to the square $[2^{-(n+1)}, 2^{-n}] \times [-2^{-(n+2)}, 2^{-(n+2)}]$.

Definition 5.2 (*Function f on* $[0, 1]$). (1) On $[\frac{1}{2}, 1]$, f maps $[\frac{1}{2}, \frac{5}{8}]$ linearly to the line segment line($\langle \frac{1}{2}, 0\rangle$, $\langle 1, 0\rangle$); f maps $[\frac{5}{8}, \frac{3}{4}]$ linearly to the line segment line($\langle 1, 0\rangle$, $\langle 1, -1\rangle$); f maps $[\frac{3}{4}, \frac{7}{8}]$ linearly to the line segment line($\langle 1, -1\rangle$, $\langle 0, -1\rangle$); and f maps $[\frac{7}{8}, 1]$ linearly to the line segment line($\langle 0, -1\rangle$, $\langle 0, 0\rangle$).

(2) $f(0) = \langle 0, 0\rangle$.

(3) For each $n \notin K$, $n \geqslant 1$, $f(t) = F_n(g(G_n(t)))$ if $t \in [2^{-(n+1)}, 2^{-n}]$.

(4) For each $n \in K$, $f(t) = F_n(g_{T(n)}(G_n(t)))$ if $t \in [2^{-(n+1)}, 2^{-n}]$.

We prove that this function f satisfies our needs.

Theorem 5.3. *This function f is one-to-one except $f(0) = f(1)$, and hence defines a simple closed curve Γ.*

Proof. Immediate from Lemma 5.1. □

Theorem 5.4. *The function f is polynomial-time computable.*

Proof. First we prove that f has a polynomial modulus. That is, we need to find a polynomial p such that $|t_1 - t_2| \leqslant 2^{-p(k)}$ implies $|f(t_1) - f(t_2)| \leqslant 2^{-k}$. Since f is piecewise linear on $[\frac{1}{2}, 1]$, we only need to show this on $[0, \frac{1}{2}]$.

We let $p(k) = 5k + 15$. Fix an integer $k > 0$ and assume that $t_1, t_2 \in [0, \frac{1}{2}]$ and $|t_1 - t_2| \leqslant 2^{-p(k)}$.

Case 1: $t_1, t_2 \in [0, 2^{-(k+2)}]$. Then, both $f(t_1)$ and $f(t_2)$ are in the rectangle $[0, 2^{-(k+2)}] \times [-2^{-(k+4)}, 2^{-(k+4)}]$, and so $|f(t_1) - f(t_2)| \leqslant 2^{-(k+1)}$.

Case 2: $t_1, t_2 \in [2^{-(n+1)}, 2^{-n}]$ for some $n \leqslant k+1$, $n \notin K$. In Theorem 4.1(b), we proved that if $|a - b| \leqslant 6^{-k}$ then $|g(a) - g(b)| \leqslant 7\beta_k \leqslant 2^{-(k-5)}$. So $|t_1 - t_2| \leqslant 2^{-p(k)}$ implies $|G_n(t_1) - G_n(t_2)| \leqslant 6^{-(k+4)}$, and hence

$$|f(t_1) - f(t_2)| = 2^{-(n+3)} \cdot |g(G_n(t_1)) - g(G_n(t_2))| \leqslant 2^{-(n+3)} \cdot 2^{-(k-1)} \leqslant 2^{-(k+2)}.$$

Case 3: $t_1, t_2 \in [2^{-(n+1)}, 2^{-n}]$ for some $n \leqslant k+1$, $n \in K$, with $T(n) \leqslant k$. In Lemma 4.1(a), we have shown that if $|a - b| \leqslant 6^{-T(n)} \cdot 2^{-k}$, then $|g_{T(n)}(a) - g_{T(n)}(b)| \leqslant 3\beta_{T(n)} \cdot 2^{-k} \leqslant 3(3 \cdot 5^{-T(n)/2}) \cdot 2^{-k} \leqslant 2^{-(k+T(n)-4)}$. Thus, $|t_1 - t_2| \leqslant 2^{-p(k)}$ implies that $|G_n(t_1) - G_n(t_2)| \leqslant 2^{-4k} \leqslant 6^{-T(n)} \cdot 2^{-k}$, and hence

$$\begin{aligned} |f(t_1) - f(t_2)| &= 2^{-(n+3)} \cdot |g_{T(n)}(G_n(t_1)) - g_{T(n)}(G_n(t_2))| \\ &\leqslant 2^{-(n+3)} \cdot 2^{-(k+T(n)-4)} \leqslant 2^{-(k+1)}. \end{aligned}$$

Case 4: $t_1, t_2 \in [2^{-(n+1)}, 2^{-n}]$ for some $n \leqslant k+1$, $n \in K$, with $T(n) > k$. From Theorem 3.10, we know that $|g_{T(n)}(x) - g(x)| \leqslant 2\beta_{T(n)}$. Therefore, by Lemma 4.1(b), $|a - b| \leqslant 6^{-k}$ implies $|g_{T(n)}(a) - g_{T(n)}(b)| \leqslant |g(a) - g(b)| + 4\beta_{T(n)} \leqslant 11\beta_k \leqslant 2^{-(k-6)}$. So, this case can be proved similar to Case 2.

Case 5: None of the above, Assume that $t_1 < t_2$ and $t_2 \in [2^{-(n+1)}, 2^{-n}]$ for some $n \leqslant k+1$. Then, we must have $t_1 \in [2^{-(n+2)}, 2^{-(n+1)}]$. Thus, both pairs $(t_1, 2^{-(n+1)})$ and $(2^{-(n+1)}, t_2)$ must satisfy conditions of Cases 1–4. It follows that $|f(t_1) - f(t_2)| \leqslant |f(t_1) - f(2^{-(n+1)})| + |f(2^{-(n+1)}) - f(t_2)| \leqslant 2^{-(k+1)} + 2^{-(k+1)} = 2^{-k}$.

The above completes the proof that f has a polynomial modulus on $[0, \frac{1}{2}]$. From Proposition 2.2, we only need to show that f is polynomial-time computable on dyadic rational points $t \in [0, 1]$. This can be done by the following algorithm.

Algorithm for f. On an dyadic rational input t, to compute $f(t)$ within an error 2^{-k}, we perform the following steps:

(1) If $t \in [0, 2^{-(k+1)}]$ then output $\langle 0, 0 \rangle$.

(2) If $t \in [\frac{1}{2}, 1]$, then compute $f(t)$ from the definition of f directly.

(3) If $t \in [2^{-(n+1)}, 2^{-n}]$ for some n, $1 \leqslant n \leqslant k$, then simulate $M(n)$ for $k+2$ moves.
  (3.1) If $M(n)$ does not halt in $k+2$ moves, then output a point $\mathbf{z}$ such that $|\mathbf{z} - F_n(g(G_n(t)))| \leqslant 2^{-(k+1)}$.
  (3.2) Otherwise, if $M(n)$ halts in $j \leqslant k+2$ moves, then output a point $\mathbf{z}$ such that $|\mathbf{z} - F_n(g_j(G_n(t)))| \leqslant 2^{-(k+1)}$.

End of Algorithm

The above algorithm is correct within an error $2^{-(k+1)}$ in case (3.2), and in the case (3.1) when $n \notin K$. The only place where possible extra errors may occur is in case (3.1) when $n \in K$. In this case, we must have $T(n) > k+2$ and hence, by Case 4 above, $|g(G_n(t)) - g_{T(n)}(G_n(t))| \leqslant 2\beta_{T(n)} \leqslant 2\beta_{k+2}$. Therefore,

$$|\mathbf{z} - f(t)| \leqslant |\mathbf{z} - F_n(g(G_n(t)))| + |F_n(g(G_n(t))) - F_n(g_{T(n)}(G_n(t)))|$$

$$\leqslant 2^{-(k+1)} + 2^{-(n+3)} \cdot 2\beta_{k+2} \leqslant 2^{-(k+1)} + 2^{-(n+3)} \cdot 6 \cdot 5^{-(k+2)/2}$$

$$\leqslant 2^{-(k+1)} + 2^{-(k+1)} = 2^{-k}. \qquad \square$$

Lemma 5.5. *The real number $r = \sum_{n \notin K} 2^{-2n}$ is nonrecursive.*

Proof. Assume by way of contradiction that r is recursive. Then, consider the following procedure to determine for each $n \geqslant 1$ whether $n \in K$.

We will decide whether $n \in K$ recursively. Suppose that we have already decided whether $k \in K$ for all $k < n$. Then, we form the number $r_n = \sum_{k<n, k \notin K} 2^{-2k}$. Then, we compute an approximation value s to r such that $|s - r| \leqslant 2^{-(2n+3)}$. If $s \leqslant r_n + 2^{-(2n+1)}$, then we decide that $n \in K$, and otherwise $n \notin K$.

Note that if $n \notin K$, then $r_n + 2^{-2n} < r$, and so $s \geqslant r - 2^{-(2n+3)} > r_n + 2^{-2n} - 2^{-(2n+3)} > r_n + 2^{-(2n+1)}$ and the decision of the above procedure is correct. If $n \in K$, then $r < r_n + \sum_{k=n+1}^{\infty} 2^{-2k} = r_n + (1/3) \cdot 2^{-2n} < r_n + 3 \cdot 2^{-(2n+3)}$. Thus, $s \leqslant r + 2^{-(2n+3)} < r_n + 4 \cdot 2^{-(2n+3)} = r_n + 2^{-(2n+1)}$, and the decision of the above procedure is again correct.

Therefore the above procedure correctly decides whether $n \in K$ or $n \notin K$. This contradicts our assumption that K is nonrecursive. $\square$

Theorem 5.6. *The interior of the curve Γ_f defined by f on $[0, 1]$ has a nonrecursive measure.*

Proof. The square $\Gamma_{sq} = [0, 1] \times [-1, 0]$ has measure 1. For each $n \in K$, the curve defined by f on $[2^{-(n+1)}, 2^{-n}]$ is of measure 0 and is symmetric with respect to the point $\langle 3 \cdot 2^{-(n+2)}, 0 \rangle$, and so this part of the curve does not change the measure of the interior of Γ_f.

For each $n \notin K$, the curve defined by f on $[2^{-(n+1)}, 2^{-n}]$ is the image of the curve Γ under the linear transformation F_n; let us call it Λ_n. The measure of the curve Λ_n is $2^{-2(n+3)} \cdot \mu(\Gamma) = 5 \cdot 2^{-(2n+6)}$. Notice that the curve Λ_n is also symmetric with respect to the point $\langle 3\cdot 2^{-(n+2)}, 0\rangle$. Therefore, half of the curve Λ_n lies inside the square Γ_{sq}, and that contributes to a decrease of the measure of the interior of Γ_f. That is, the effect of f on $[2^{-(n+1)}, 2^{-n}]$ is a decrease of the interior by a measure of $\frac{1}{2}\cdot 5\cdot 2^{-(2n+6)} = 5\cdot 2^{-(2n+7)}$. Thus the area of the region inside Γ_f is

$$1 - \sum_{n\notin K} 5\cdot 2^{-(2n+7)} = 1 - \frac{5}{128}\sum_{n\notin K} 2^{-2n},$$

which is nonrecursive by Lemma 5.5. □

6. The length problem

In this section, we construct a polynomial-time computable, simple curve which is of finite length but whose length is a nonrecursive real number. As in Section 5, we let K be a recursively enumerable but nonrecursive set of nonnegative integers. We also let M be a Turing machine accepting K, and for each $n \in K$, let $T(n)$ be the number of moves for $M(n)$ to halt. Without loss of generality, we assume that $0, 1, 2 \notin K$ and $T(n) \geqslant n + 2$ for all $n \in K$.

For each $n \geqslant 2$, let l_n be the length of the curve Γ_n defined by function g_n on $[0, 1]$. Note that we proved in Corollary 4.12 that $\{l_n\}$ is a polynomial-time computable sequence. For each $n \in K$, define

$$\lambda_n = 2^{-2n}\cdot(\tfrac{2}{5}\cdot l_{T(n)} - 1)^{-1}.$$

From Corollary 4.11, we know that $5^{T(n)/2} < l_{T(n)} < 5^{T(n)/2+1}$. Therefore, $5^{T(n)/2-1} < (2/5)l_{T(n)} - 1 < 5^{T(n)/2+1}$, and hence

$$2^{-2n}\cdot 5^{-(T(n)/2+1)} < \lambda_n < 2^{-2n}\cdot 5^{-(T(n)/2-1)}.$$

Next we define, for each $n \in K$, two linear mappings $G_n: \mathbb{R} \to \mathbb{R}$ and $F_n: [-2, 2]\times[-2, 2] \to \mathbb{R}^2$ as follows:

$$G_n = \frac{1}{2} + \frac{t - 2^{-n}}{2\lambda_n},$$

$$F_n(\langle x, y\rangle) = \left\langle \frac{4\lambda_n}{5}\cdot x + 2^{-n}, \frac{4\lambda_n}{5}\cdot y \right\rangle.$$

For each $n \in K$, let $J_n = [2^{-n} - \lambda_n, 2^{-n} + \lambda_n]$. We observe that for each $n \in K$, G_n maps the interval J_n to the interval $[0, 1]$, and F_n maps the square $[-2, 2]\times[-2, 2]$ to the square $Q_n = [2^{-n} - 8\lambda_n/5, 2^{-n} + 8\lambda_n/5]\times[-8\lambda_n/5, 8\lambda_n/5]$. By Theorem 5.1(b), we see that F_n maps all curves Γ and Γ_m, $m \geqslant 0$, into square Q_n. Since, for $n \in K$, $\lambda_n \leqslant 2^{-2n}\cdot 5^{-(T(n)/2-1)} < 2^{-(n+3)}$ (note that $n \in K$ implies that $n \geqslant 3$), the square Q_n does not overlap with any other square Q_m for all $n, m \in K$. In addition, F_n maps the line segment line$(\langle -\alpha_1, 0\rangle, \langle \alpha_1, 0\rangle)$ to the line segment line$(\langle 2^{-n} - \lambda_n, 0\rangle, \langle 2^{-n} + \lambda_n, 0\rangle)$.

Now we define the function f on [0, 1] as follows:
(1) If $t \notin J_n$ for any $n \in K$, then let $f(t) = \langle t, 0\rangle$.
(2) If $t \in J_n$ for some $n \in K$, then let $f(t) = F_n(g_{T(n)}(G_n(t)))$.

Lemma 6.1. *The function f defined above is polynomial-time computable.*

Proof. We first show that f has a polynomial modulus on [0, 1]. We are going to show that if $|t_1 - t_2| \leqslant 2^{-q(k)}$, then $|f(t_1) - f(t_2)| \leqslant 2^{-k}$, where $q(k) = 8(k+2)$. Let us fix an integer $k > 0$. Assume that $|t_1 - t_2| \leqslant 2^{-q(k)}$, and consider the following cases:

Case 1: Both t_1 an t_2 are not in J_n for any $n \in K$. Then, $|f(t_1) - f(t_2)| = |t_1 - t_2| \leqslant 2^{-(k+2)}$.

Case 2: Both t_1 and t_2 are not in J_n for some $n \in K$ with $T(n) \geqslant k+3$. Then both $f(t_1)$ and $f(t_2)$ are in the square Q_n. Thus, $|f(t_1) - f(t_2)| \leqslant (\frac{32}{5})\lambda_n \leqslant 2^{-(2n-5)} \cdot 5^{-T(n)/2} \leqslant 2^{-T(n)} \leqslant 2^{-(k+2)}$.

Case 3: Both t_1 and t_2 are in J_n for some $n \in K$ with $T(n) \leqslant k+2$. In Lemma 4.1(a), we have shown that if $|a - b| \leqslant 6^{-T(n)} \cdot 2^{-k}$ then $|g_{T(n)}(a) - g_{T(n)}(b)| \leqslant 3\beta_{T(n)} \cdot 2^{-k}$. Thus, $|t_1 - t_2| \leqslant 2^{-q(k)}$ implies

$$|G_n(t_1) - G_n(t_2)| \leqslant \frac{2^{-q(k)}}{2\lambda_n} \leqslant 2^{-8(k+2)} \cdot 2^{2n-1} \cdot 5^{T(n)/2+1}$$
$$\leqslant 2^{-6(k+2)} \cdot 4^{T(n)} \leqslant 6^{-T(n)} \cdot 2^{-k}.$$

It follows that

$$|f(t_1) - f(t_2)| \leqslant \frac{4\lambda_n}{5} \cdot |g_{T(n)}(G_n(t_1)) - g_{T(n)}(G_n(t_2))| \leqslant \frac{4\lambda_n}{5} \cdot 3\beta_{T(n)} \cdot 2^{-k} \leqslant 2^{-(k+2)}.$$

Case 4: $t_1 \in J_n$ for some $n \in K$ and $t_2 \in I_m$ for some $m \in K$ with $m > n$. Let t_3 be the right endpoint of J_n and t_4 be the left endpoint of J_m. Then, all $|t_1 - t_3|$, $|t_3 - t_4|$ and $|t_4 - t_2|$ are bounded by $2^{-q(k)}$ and the pairs (t_1, t_3) and (t_4, t_2) satisfy the condition of either Case 2 or Case 3, the pair (t_3, t_4) satisfies the condition of Case 1. Thus, we have

$$|f(t_1) - f(t_2)| \leqslant f(t_1) - f(t_3)| + |f(t_3) - f(t_4)| + |f(t_2) - f(t_4)|$$
$$\leqslant 3 \cdot 2^{k+2} < 2^{-k}.$$

Case 5: t_1 is in J_n for some $n \in K$, and t_2 is not in J_m for any $m \in K$. This case can be proved similar to Case 4.

The above completes the proof that f has a polynomial modulus. To see that f is polynomial-time computable on dyadic rationals, we consider the following algorithm for f:

Algorithm for f. On the dyadic rational input t, to compute $f(t)$ within an error 2^{-k}, we perform the following steps:
(1) If $t \in [0, 2^{-(k+1)} + 2^{-(k+3)}]$ then output $\langle t, 0\rangle$.
(2) If $t \notin [2^{-n} - 2^{-(n+2)}, 2^{-n} + 2^{-(n+2)}]$ for any $n \leqslant k$, then output $\langle t, 0\rangle$.

(3) If $t \in [2^{-n} - 2^{-(n+2)}, 2^{-n} + 2^{-(n+2)}]$ for some $n \leqslant k$, then simulate $M(n)$ for k moves.
  (3.1) If $M(n)$ does not halt in k moves, then output $\langle t, 0 \rangle$.
  (3.2) Otherwise, if $M(n)$ halts in k moves, then we have found $T(n) \leqslant k$. We determine whether $t \in J_n$. (a) If $t \notin J(n)$, then output $\langle t, 0 \rangle$. (b) Otherwise, we output a point $\mathbf{z}$ such that $|\mathbf{z} - F_n(g_{T(n)}(G_n(t)))| \leqslant 2^{-k}$.

End of Algorithm

To see that the above algorithm for f is correct, we verify the following cases:

(a) If $t \in [0, 2^{-(k+1)} + 2^{-(k+3)}]$ then $f(t)$ is either $\langle t, 0 \rangle$ or is in Q_n for some $n \in K$, $n \geqslant k+1$. In either case, $|f(t) - \langle t, 0 \rangle| \leqslant (\frac{35}{5})\lambda_n \leqslant (\frac{32}{5}) \cdot 2^{-(2k+2)} \leqslant 2^{-k}$.
(b) If $t \notin J_n$ for any $n \in K$, then the above algorithm must output the correct point $\langle t, 0 \rangle$ (in Case (1), (2), (3.1) or (a) of (3.2).
(c) If $t \in J_n$ for some $n \in K$, $n \leqslant k$ and $T(n) \leqslant k$, then the above algorithm must enter Case (3.2.b) and the output is correct within an error 2^{-k}.
(d) If $t \in J_n$ for some $n \in K$, $n \leqslant k$ and $T(n) > k$, then the above algorithm outputs $\langle t, 0 \rangle$ in Case (3.1). Note that both $f(t)$ and $\langle t, 0 \rangle$ are in Q_n and so

$$|f(t) - \langle t, 0 \rangle| \leqslant 32\lambda_n/5 \leqslant 2^{-(2n-5)} \cdot 5^{-T(n)/2} < 2^{-T(n)} < 2^{-k}.$$

Finally, we verify that the above algorithm can be implemented in polynomial time. We note that steps (1), (2) and (3.1) can all be done in time polynomial in k. In Case (3.2), we have already found $T(n) \leqslant k$, and so we can compute $l_{T(n)}$ and hence λ_n, and also $g_{T(n)}$ in polynomial time. This completes the proof of the lemma. □

Lemma 6.2. *The length of the curve Γ'_f defined by function f on* [0, 1] *is not a recursive real number.*

Proof. Suppose that $n \in K$. Then, the curve defined by f on the interval J_n is the image of the curve $\Gamma_{T(n)}$ under the linear transformation F_n. The length of the curve $\Gamma_{T(n)}$ is $l_{T(n)}$, and the function F_n shrinks it by the factor $4\lambda_n/5$. Thus, the length of the curve Γ'_f between $2^{-n} - \lambda_n$ and $2^{-n} + \lambda_n$ is $4\lambda_n l_{T(n)}/5$. Therefore, the total length of the curve Γ'_f is

$$1 - \sum_{n \in K} 2\lambda_n + \sum_{n \in K} \frac{4}{5} \cdot l_{T(n)} \cdot \lambda_n = 1 + \sum_{n \in K} 2\lambda_n \left(\frac{2}{5} l_{T(n)} - 1 \right) = 1 + \sum_{n \in K} 2^{-(2n-1)},$$

which is not recursive (Lemma 5.5). □

Thus, the following theorem is proven.

Theorem 6.3. *There exists a polynomial-time computable, one-to-one function $f: [0, 1] \to \mathbb{R}^2$ that defines a simple curve Γ'_f such that Γ'_f has a finite length but its length is a nonrecursive real number.*

7. The membership problem

In Section 5, we proved that the interior S of the curve Γ_f has a nonrecursive measure; thus, the membership problem for set S is, intuitively, not recursive either, for otherwise we could compute its measure by a sampling technique. In this section we prove, however, that the membership problem for S is still polynomial-time solvable, if we relax our requirement on the membership algorithm on points close to the boundary. That is, we prove that the set S is polynomial-time recognizable in the following sense.

For any set $S \subseteq \mathbb{R}^2$, let $\chi_S(\mathbf{z}) = 1$ if $\mathbf{z} \in S$ and $\chi_S(\mathbf{z}) = 0$ if $\mathbf{z} \notin S$, and let Γ_S be the set of all points $\mathbf{z}$ in $\mathbb{R}^2$ such that for all $\varepsilon > 0$, the neighborhood $\mathrm{N}(\mathbf{z}; \varepsilon)$ intersects with both S and the complement of S.

Definition 7.1 (*Chou and Ko* [1]). *A set $S \subseteq \mathbb{R}^2$ is polynomial-time recognizable* if there exist a two-oracle Turing machine M and a polynomial p such that for any oracles ϕ, ψ that binary converge to x and y, respectively, and for any input n, $M^{\phi,\psi}(n)$ outputs $\chi_S(\langle x, y\rangle)$ correctly whenever $\langle x, y\rangle$ has a distance $\geqslant 2^{-n}$ to Γ_S.

The notion of polynomial-time recognizability, and its relation to other notions of polynomial-time computability of sets in $\mathbb{R}^2$, as well as their applications, are discussed in [1]. (For instance, from [1], we can conclude that the interior S of Γ_f is not *polynomial-time approximable.*)

Before we prove our main result, we first consider a simpler region. We extend the curve Γ defined by function g on $[-\frac{1}{2}, \frac{3}{2}]$ to a simple closed curve by adding the curve Λ_0 to it, where Λ_0 is the boundary of the square $[-2, 2] \times [-2, 0]$ with the top line segment removed. We still call this new curve Γ. Similarly, we extend Γ_n to include Λ_0.

Lemma 7.2. *The interior S of the curve Γ is polynomial-time recognizable.*

Proof. In the following we present a recursive algorithm to determine whether a point $\mathbf{z}$ belongs to the set S. It suffices to consider the points $\mathbf{z} = \langle x, y\rangle$ for some dyadic rationals x and y in $\mathbf{D}$, since for each point $\mathbf{z}_1$, we can find such a *dyadic point* $\mathbf{z}$ close to it such that if $\mathbf{z}_1$ has a distance $\geqslant 2^{-k}$ to Γ then $\mathbf{z}$ has a distance $\geqslant 2^{-(k+1)}$ to Γ. (The following algorithm follows the notations of Sections 4.2 and 5.)

Algorithm for S. Assume that the input consists of a *dyadic point* $\mathbf{z}$ and an integer $k > 0$. (The algorithm is expected to compute $\chi_S(\mathbf{z})$ if $\mathbf{z}$ has a distance greater than 2^{-k} to the curve Γ.) Initially, we construct the box $\mathrm{Box}(v_0)$ where v_0 is the root of the tree T. Then we determine whether $\mathbf{z}$ is in $\mathrm{Box}(v_0)$. If not, then we can easily determine whether $\mathbf{z}$ is in S or not in S. (More precisely, $\mathbf{z}$ is in S iff $\mathbf{z}$ is in the square $[-2, 2] \times [-2, 0]$.) Next, for each n, $1 \leqslant n \leqslant k + 4$, we recursively do one of the following:

(1) Find three neighboring nodes v_i, $i = 1, 2, 3$, of depth n (where v_i is the left neighbor of v_{i+1}, $i = 1, 2$) such that $\mathbf{z} \in \mathrm{Box}(v_2) - \mathrm{Box}(v_1) \cup \mathrm{Box}(v_3)$; or
(2) Find four neighboring nodes v_i, $i = 1, 2, 3, 4$, of depth n (where v_i is the left neighbor of v_{i+1}, $i = 1,2,3$) such that $\mathbf{z} \in [\mathrm{Box}(v_2) \cap \mathrm{Box}(v_3)] - [\mathrm{Box}(v_1) \cup \mathrm{Box}(v_4)]$; or
(3) Determine that $\mathbf{z} \notin \mathrm{Box}(u)$ for any node u of depth n; then determine whether $\mathbf{z}$ is in the interior of Γ_n and halt.

(In the above, by finding a node v_1 we mean to find the node type of v_1, the endpoints of the line segment L_{v_1}, and label(v_1).)

Assume that after $k + 4$ steps, we are in either case (1) or case (2); then we output that $\mathbf{z} \in S$.

End of Algorithm.

We first show that the above recursive steps can be done in polynomial time. We first assume that we have already found four neighboring nodes v_i, $i = 1, 2, 3, 4$, of depth n such that $\mathbf{z} \in [\mathrm{Box}(v_2) \cap \mathrm{Box}(v_3)] - [\mathrm{Box}(v_3) \cup \mathrm{Box}(v_4)]$, and we want to do either (1), (2) or (3) with respect to depth $n + 1$. To do so, we first generate all children u of the nodes v_2 and v_3, and determine whether $\mathbf{z}$ is in the box $\mathrm{Box}(u)$. Note that for each node u, the size of the box $\mathrm{Box}(u)$ is easily computable from its definition, as long as the node type, the endpoints of L_u and label(u) are known. By Lemmas 4.6 and 4.7, we know that there are at most two nodes u of depth $n + 1$ such that $\mathbf{z} \in \mathrm{Box}(u)$. Furthermore, if there are two such nodes, then these two nodes must be neighbors. So, if such nodes exist, we can determine either case (1) or case (2).

Assume that $\mathbf{z} \notin \mathrm{Box}(u)$ for all children nodes u of v_2 and v_3. Then we have reached case (3). Let $u_1, \ldots, u_j$ be the children nodes of nodes v_2 and v_3. Then, the line segments $L_{u_1}, \ldots, L_{u_j}$ form a section of the curve Γ_{n+1} that cuts through $\mathrm{Box}(v_2) \cap \mathrm{Box}(v_3)$ and divides it into 2 regions. (By Lemma 4.7, there is no other line segment of Γ_{n+1} in $\mathrm{Box}(v_2) \cap \mathrm{Box}(v_3)$.) This curve is a directed curve such that the "right-hand side" of the curve is the interior of the curve Γ_{n+1}, and the "left-hand side" is the exterior. So, we just determine whether $\mathbf{z}$ is in the "right-hand" region of $\mathrm{Box}(v_2) \cap \mathrm{Box}(v_3)$ or is in the "left-hand" region, and output accordingly.

The above gave a polynomial-time implementation of the step $n + 1$ of the above algorithm, assuming we were in case (2) at step n. The case when we were in case (1) in step n can be implemented in a similar way. We omit the details.

It is left to show that the above alborithm is correct. First we show that if we reach case (3) within $k + 4$ steps, then the decision must be correct (i.e., if we halt at step $n + 1$, then $\mathbf{z}$ is in S iff $\mathbf{z}$ is in the interior of Γ_{n+1}). This is actually quite obvious: if we were in case (1) at step n, then the curve Γ within the area $\mathrm{Box}(v_2) \cap \mathrm{Box}(v_3)$ must lie within the union of $\mathrm{Box}(u)$ over all children nodes u of v_2 and v_3. Therefore, $\mathbf{z}$ is not on the curve Γ. Furthermore, the orientation of the curve Γ and Γ_{n+1} are identical, and so $\mathbf{z}$ is in the interior of Γ iff it is in the interior of Γ_{n+1}.

Next we claim that if $\mathbf{z}$ has a distance $\delta > 2^{-k}$ to the curve Γ then $\mathbf{z} \notin \mathrm{Box}(u)$ for any node u in tree T of depth $k + 4$. This claim implies that we must reach case (3) before the step $k + 4$, and so the algorithm must be correct as long as the distance between $\mathbf{z}$ and Γ is greater than 2^{-k}. To see this, we assume, for the sake of contradiction, that $\mathbf{z}$ is in Box(u) for some node u of depth $m = k + 4$. Then, there is a point $\mathbf{z}_u$ on the line segment L_u such that

$$|\mathbf{z} - \mathbf{z}_u| \leqslant \sigma_m + \tau_m \leqslant \alpha_m + \beta_m < 2^{-(k+1)}.$$

Furthermore, from Theorem 3.10, there is a point $\mathbf{z}_0$ on the curve Γ such that $|\mathbf{z}_0 - \mathbf{z}_u| \leqslant \alpha_m + \beta_m < 2^{-(k+1)}$. Together, we conclude that the distance between $\mathbf{z}$ and $\mathbf{z}_0$, and hence the distance between $\mathbf{z}$ and the curve Γ, is less than 2^{-k}. This contradicts our assumption, and so the claim is proven. □

Corollary 7.3. *The interior S_m of the curve Γ_m is polynomial-time recognizable.*

Proof. We make the following simple modification to the algorithm of Lemma 7.2: If $m < k + 4$ and the original algorithm does not halt in $m - 1$ steps, then in step m, compute Γ_m within the relevent boxes (Box(v_2) in case (1) and Box(v_2) $\cap$ Box(v_3) in case (2)) and determine precisely whether $\mathbf{z}$ is in Γ_m.

Note that if the original algorithm halts in case (3) at step $n < m$, then the decision is correct, since the difference between Γ_m and Γ_n only affect points $\mathbf{z}$ in Box(v) for some nodes v of depth n. Also, if $m \leqslant k + 4$ and the original algorithm does not halt by step $m - 1$, then our modification recognizes $\mathbf{z}$ precisely. Finally, if $m > k + 4$ and the original algorithm does not halt in case (3) by step $k + 4$, then we can see, by an analysis similar to that in Lemma 7.2, that there is a point $\mathbf{z}_0$ in Γ_m such that $|\mathbf{z} - \mathbf{z}_0| < 2^{-k}$ and so $\mathbf{z}$ has a distance less than 2^{-k} to Γ_m and the correctness of the algorithm is irrelevant. □

Now we consider the curve Γ_f defined by function f on $[0, 1]$ of Section 5.

Theorem 7.4. *The interior S of the curve Γ_f defined in Section 5 is polynomial-time recognizable.*

Proof. For any dyadic point $\mathbf{z} \in \mathbb{R}^2$ and any integer k, we determine whether $\mathbf{z}$ is in the interior of Γ_f as follows:

(1) If $\mathbf{z} \in [0, 2^{-(k+4)}] \times [-2^{-(k+6)}, 2^{-(k+6)}]$, then the distance between $\mathbf{z}$ and Γ_f is less than $2^{-(k+3)}$, and we output 1.

(2) If $\mathbf{z}$ is not in the square $E_n = [2^{-(n+1)}, 2^{-n}] \times [-2^{-(n+2)}, 2^{-(n+2)}]$ for any $n \leqslant k + 3$, then determine $\mathbf{z}$ is in S iff $\mathbf{z}$ is in the square $[0, 1] \times [-1, 0]$.

(3) If $\mathbf{z}$ is in E_n for some $n \leqslant k + 3$, then simulate Turing machine M on input n for $k + 4$ moves. Define $\mathbf{z}_1 = F_n^{-1}(\mathbf{z})$, where F_n is the linear transformation defined in Section 5; i.e., if $\mathbf{z} = \langle x, y\rangle$, then $\mathbf{z}_1 = \langle 2^{n+3}(x - 3 \cdot 2^{-(n+2)}, 2^{n+3} \cdot y\rangle$.

   (3.1) If $M(n)$ halts in $k + 4$ moves, then we know $k + 4 \geqslant T(n)$. We apply the algorithm of Corollary 7.3 to determine whether $\mathbf{z}_1$ is in the interior of $\Gamma_{T(n)}$ with respect to the error parameter k.

(3.2) If $M(n)$ does not halt in $k + 4$ moves, then we apply the algorithm of Lemma 7.2 to determine whether $\mathbf{z}_1$ is in the interior of Γ with respect to the error parameter k.

Assume that $\mathbf{z}$ has a distance $\geqslant 2^{-k}$ to Γ_f. Then, in case (3.1), it is easy to see that $\mathbf{z}_1$ has a distance $\geqslant 2^{-k} \cdot 2^{n+3} > 2^{-(k-1)}$ to the curve $\Gamma_{T(n)}$. Furthermore, $\mathbf{z}_1$ is in the interior of $\Gamma_{T(n)}$ iff $\mathbf{z}$ is in the interior of Γ_f. So the algorithm works correctly when the distance between $\mathbf{z}$ and Γ_f is greater than 2^{-k}. Similarly, the algorithm works correctly in the subcase of (3.2) when $n \notin K$.

Finally, in the subcase of (3.2) when $n \in K$, we have $T(n) > k + 4$. We claim that if $\mathbf{z}$ has a distance $\geqslant 2^{-k}$ to Γ_f then $\mathbf{z}_1$ is in the interior of $\Gamma_{T(n)}$ iff $\mathbf{z}_1$ is the interior of Γ. The correctness of the algorithm then follows.

To see that the claim is true, we note that $\Gamma_{T(n)}$ and Γ has the relation that

$$|g_{T(n)}(t) - g(t)| \leqslant 2 \cdot \beta_{T(n)} \leqslant 6 \cdot 5^{-T(n)/2} < 2^{-k}.$$

Thus, a point $\mathbf{z}_2$ in the interior of one curve but in the exterior of the other curve must ave a distance $\leqslant 2^{-k}$ to both curves. Since $\mathbf{z}$ has a distance $\geqslant 2^{-k}$ to Γ_f, $\mathbf{z}_1$ has a distance $\geqslant 2^{-(k-n-3)}$ to $\Gamma_{T(n)}$. This implies that $\mathbf{z}_1$ must be in the interior of both curves or in the exterior of both curves. □

Acknowledgements

The author has benefited greatly from many insightful comments and suggestions of Professor Arthur Chou. The author is grateful to the anonymous referee for pointing out a technical error in the first draft of the paper.

References

[1] A. Chou and K. Ko, Computational complexity of two-dimensional regions, *SIAM J. Comput.*, to appear; see also: A. Chou and K. Ko, Some complexity issues on the simply connected regions of the two-dimensional plane, *Proc. 34th IEEE Symp. Foundations of Computer Science* (1993) 1–10.
[2] H. Friedman, On the computational complexity of maximization and integration, *Adv. in Math.* **53** (1984) 80–98.
[3] K. Ko, *Complexity Theory of Real Functions* (Birkhäuser, Boston, MA, 1991).
[4] K. Ko and H. Friedman, Computational complexity of real functions, *Theoret. Comput. Sci.* **20** (1982) 323–352.
[5] H. Lebesgue, Sur le problème des aires, in Lebesgue, H., *Oeuvres Scientifiques*, Vol. IV (Enseignement Mathématique, Genève, 1903) 29–35.
[6] B. Mandelbrot, *The Fractal Geometry of Nature* (Freeman, New York, 1983).
[7] W.H. Osgood, A Jordan curve of positive area, *Trans. Amer. Math. Soc.* **4** (1903) 107–112.