On the Computability of Rectifiable Simple Curve *

(Extended Abstract)

Robert Rettinger[1] and Xizhong Zheng[2,3]**

[1] Lehrgebiet Algorithmen und Komplexität
FernUniversität Hagen, 58084 Hagen, Germany
[2] Department of Computer Science and Mathematics
Arcadia University, Glenside, PA 19038, USA
[3] Departments of Mathematics and Computer Science
Jiangsu University, Zhenjiang 212013, China

Abstract. In mathematics curves are defined as the images of continuous real functions defined on closed intervals and these continuous functions are called parameterizations of the corresponding curves. If only simple curves of finite lengths are considered, then parameterizations can be restricted to the injective continuous functions or even to the continuous length-normalized parameterizations. In addition, a plane curve can also be considered as a connected one-dimensional compact subset of points. By corresponding effectivizations, we will introduce in this paper four versions of computable curves and show that they are all different. More interestingly, we show also that four classes of computable curves cover even different sets of points.

Keywords: Computable Curve, Simple Curve, Rectifiable Curve, Point Separability

1 Introduction

In computable analysis, we are mainly interested in the computability over various continuous structures. One realistic approach to this kind of computability is the Turing-machine-based bit model (see [7, 11, 2]). In this model, real numbers are represented by effectively convergent sequences of rational numbers and these sequences are called *names* of the real numbers. Here a sequence (x_n) converges effectively means that $|x_n - x_{n+1}| \leq 2^{-n}$ for all n. A real number x is computable if it has a computable name. Furthermore, a real function f is computable if there is a Turing machine which transfers each name of a real number x in the domain of f into a name of $f(x)$. By the same principle, computability of other mathematical objects can be defined by introducing proper "naming

* This work is supported by DFG (446 CHV 113/266/0-1) and NSFC (10420130638)
** Corresponding author. email: ZhengX@Arcadia.edu

Andrej Bauer, Peter Hertling, Ker-I Ko (Eds.)
6th Int'l Conf. on Computability and Complexity in Analysis, 2009, pp. 221-232
http://drops.dagstuhl.de/opus/volltexte/2009/2273

systems". For example, the computability of subsets of the Euclidean space [1], of semi-continuous functions [12], of functional spaces [13] are all defined in this way. All these computability of mathematical objects are achieved by a kind of "effectivization" of the classic mathematic definitions.

Particularly, we can introduce the computability of curves in this way too. We consider the plane curves in this paper only. The curves of higher dimensions can be discussed in essentially the same way. Notice that, there are different mathematical approaches to define curves. For example, a curve can be defined as a connected and one-dimensional compact subset. Based on this approach we can define the computable curves by means of the computability of compact subsets of Euclidean space ([1]). Physically, a curve records the trace of a particle motion. If the particle moves according to some algorithmically definable laws, its trace should be regarded as computable. In mathematical terms, a curve is the range of a continuous function defined on a closed interval and this function is called a parametrization of the curve. If a curve has a computable parametrization, then it should be naturally considered as a computable curve (see e.g., [4, 5]).

However, the parametrization of a curve may have various extra properties, particulary if we consider the curves which do not intersect itself and have finite length. Normally, a parameterization of a plane curve C is just a continuous function $f : [0, 1] \to \mathbb{R}^2$. This paramaterization possibly traces some segment of the curve several times. That is, the parameterization f retraces the curve, or it is retraceable. If a curve does not intersect itself, then, by a classic theorem in analysis, it has always an injective parameterization (with possibly exemption at the endpoints of the interval). In addition, if C has a finite length, then it has even an arc-length normalized parameterization. Here a parametrization f is called *arc-length normalized*, if the curve-segment $f([0, t])$ has a length proportional to the parameter t, for any $t \in [0, 1]$.

In this paper we will introduce four versions of computable curves by effectivizing above four mathematical approaches to the curves. We will see that these four versions of computability about curves are all different. The difference of the computability of curves introduced by computable parameterizations and computable injective parameterizations was already shown by Gu, Lutz and Mayordomo in a recent paper [5]. The separations of four versions of computable curves shown in this paper hold actually in a more stronger sense. Namely, the point sets covered by four classes of computable curves are also different. In other words, different versions of computable curves can be separated by points and then they are "point-separable" (see definition in Section 4).

Our paper is organized as follows. In Section 2 we will briefly recall some basic notions related to curves, give the precise definition of computable curves and then show some basic properties of computable curves. In Section 3, we show a technical lemma which will be used in the proof of the main theorem. In Section 4 we prove our main results that four classes of computable curves in different sense are point-separable.

2 Computable Curves

In mathematics, a *plane curve* is defined as a subset $C \subseteq \mathbb{R}^2$ which is the range of a continuous function $f : [0;1] \to \mathbb{R}^2$, i.e., $C = \text{range}(f)$. This continuous function f is then called a *parametrization* of C. Here we use w.l.o.g. the unit interval $[0,1]$ instead of more general closed intervals of the form $[a,b]$. Obviously, any curve has infinitely many parameterizations. Geometrically, a curve records the path of a particle movement on the plane. If the particle never visit one position more than once, in other words, if the curve does not intersect itself (with possible exemption of end points), then the curve is called *simple*. A classical mathematical theorem asserts that, any simple curve has a parameterization $f : [0;1] \to \mathbb{R}^2$ which is injective on $[0;1)$. If a curve C has an injective parameterization f (meaning injective on the interval $[0;1)$) and fulfills in addition $f(0) = f(1)$, then the curve C is called *closed*.

For the simple curves, their lengths can be defined by approximation of the lengths of polygons which converges to the curves according to Jordan [6]. More precisely, Let C be a simple curve and let $f : [0;1] \to \mathbb{R}^2$ be an injective continuous parameterization of C. The *length* L of the curve C is then defined by

$$L := \sup \sum_{i=0}^{n} |f(a_i) - f(a_{i+1})|.$$

where $|f(a_i) - f(a_{i+1})|$ is the length of the straight line connecting the points $f(a_i)$ and $f(a_{i+1})$ and the supremum is taken over all possible partitions $0 = a_0 < a_1 < \ldots < a_n = 1$. The length of a curve C is denoted by $l(C) := L$. A curve of a finite length is traditionally called *rectifiable*. Not every curve has a finite length. Some curves can even fill whole space like Peano curves (see e.g. [3]). In this paper we are mainly interested in the simple rectifiable curves.

It is well known in analysis that every simple, rectifiable curve has also a length-normalized parameterization. Here a length-normalized (or simply normalized) parameterization of a curve C is an injective continuous function $f : [0,1] \to \mathbb{R}^2$ such that the curve segment $f([0,t])$ has the length $t \cdot l(C)$ for all $t \in [0,1]$. Thus, a simple rectifiable curve can have three different kind of parameterizations—continuous, injective continuous and normalized. In addition, a curve can also be defined as a connected one-dimensional compact point set. By effectivizing these approaches to curves, we can introduce four different versions of computable curves.

Remember that a real function $f : [0;1] \to \mathbb{R}$ is computable if there is a Turing machine M which transfers any name of $x \in [0,1]$ to a name of $f(x)$. Equivalently, f is computable iff there is a computable sequence $(p_n)_{n\in\mathbb{N}}$ of computable rational polygon functions which converges uniformly and effectively to f (see [10]). Naturally, a function $f : [0;1] \to \mathbb{R}^n$ is computable if all of its component functions are computable, or equivalently, if there is a Turing machine M which transfers any name of $x \in [0,1]$ into a tuple $(\alpha_1, \cdots, \alpha_n)$ of names of $f_1(x), \cdots, f_n(x)$ respectively, where $f(x) = (f_1(x), \cdots, f_n(x))$. In this case, we simply say that M computes the function f.

Now we call define the computable curves as follows.

Definition 1. *Let C be a simple plane curve.*

1. *C is called* K-computable *if there is a computable sequence (Q_n) of finite sets of rational neighborhoods such that*

$$C \subseteq \bigcup Q_n \text{ and } d_H\left(\bigcup Q_n,\ C\right) < 2^{-n} \tag{1}$$

for all $n \in \mathbb{N}$, where d_H denotes the Hausdorff distance.

2. *C is called* R-computable *if there is a computable function $f : [0;1] \to \mathbb{R}^2$ such that* range$(f) = C$.
3. *C is called* M-computable *if there is a computable function $f : [0;1] \to \mathbb{R}^2$ which is injective on $[0;1)$ such that* range$(f) = C$.
4. *C is called* N-computable *if C has a computable parameterization $f : [0;1] \to \mathbb{R}^2$ such that the length of the curve segment $f([0,t])$ is equal to $t \cdot l(C)$ for all $t \in [0,1]$.*

In the item 1 of the definition, the finite sets Q_n of rational neighborhoods are also called compact covers of the curve C. The second part of the condition (1) means that the maximal distance from C to bordering of the compact cover Q_n is bounded by 2^{-n}. In this paper, an ε-neighborhood $V_\varepsilon(a,b)$ of a point with Cartesian coordinates (a,b) means the rectangle bounded by the lines $x = a \pm \varepsilon$ and $y = b \pm \varepsilon$. A neighborhood $V_\varepsilon(a,b)$ is called rational if a, b and ε are all rational numbers. The letter K of the K-computability comes from the German word Kompakt (compact) due to the compact coverings.

In the item 2, the letter R stands for Retracable because the parametrization f of a R-computable curve C can retrace the curve C. Namely, there could be some disjoint subintervals $I_1, I_2 \subset [0,1]$ such that $f(I_1) = f(I_2)$. In this case, f traces some pieces of C more than once, or f is retraceable.

If the paramaterization of a curve C is injective, then C records the movement of a particle with a monotone direction. The letter M in M-computability stands for Monotonically directed movement. Notice that, in this paper, we call a parameterization $f : [0,1] \to \mathbb{R}^2$ injective even if it is only injective on $[0;1)$ and does not exclude the possible case $f(0) = f(1)$. This should not cause essential confusions.

Finally, if a parameterization $f : [0,1] \to \mathbb{R}^2$ satisfies the condition that the length of the curve segment $f([0,t])$ is proportional to t, then it is called arc-length normalized. Thus, N-computability stands for Normalized parameterization.

It is well know that not every curve has a finite length. For example, the famous Peano curve can even fill the two-dimensional plan (see e.g., Peano [9]) and has an infinite length. From the definition 1, an N-computable curve has always a finite length. However, the next theorem shows that an M-computable curve does not necessarily have an finite length any more. This distinguishes the N-computability from other three versions of computability immediately.

Theorem 1. *There is an M-computable curve C which has an infinite length.*

Proof. (Sketch) We can construct firstly a computable sequence (p_n) of rational polygons such that distance between p_n and p_{n+1} is bounded by 2^{-n} and p_{n+1} has doubled length of p_n by introducing many small zigzags, for all n. Then, the limit $p := \lim p_n$ is a curve of infinite length. Corresponding to each polygon p_n we can define a computable injective function $f_n : [0,1] \to \mathbb{R}^2$ as a parameterization of p_n, and in addition, we can require that $|f_n(t) - f_{n+1}(t)| \leq 2^{-n}$ is satisfied for all $n \in \mathbb{N}$ and $t \in [0,1]$. Therefore, the limit function $f := \lim f_n$ is an injective computable parameterization of the curve p and hence p is an M-computable curve with an infinite length.

Although a computable curve may have an infinite length, computable rectifiable curves seem more interesting and more important. In this paper we will mainly focus only on the computable curves of finite length and we denote by $\mathbb{C}_K, \mathbb{C}_R, \mathbb{C}_M$ and $\mathbb{C}_N$ the classes of all K-, R-, M- and N-computable rectifiable curves, respectively. By definition, it is straightforward that we have the following relationship between these four versions of computable curves.

Theorem 2. $\mathbb{C}_N \subseteq \mathbb{C}_M \subseteq \mathbb{C}_R \subseteq \mathbb{C}_K$.

Actually we will see that all these four versions of computability of curves are different and hence all the subset relations above are proper.

In the paper [5], Gu, Lutz and Mayordomo have shown that any rectifiable R-computable curve has a left computable length, where a real number x is left computable or computably enumerable (c.e. for short) if there is an increasing computable sequence (x_n) of rational numbers which converges to x. This can be strengthen further to the K-computable curves as follows.

Theorem 3. *Any rectifiable K-computable curve has a left computable length.*

Proof. (Sketch) If C is a rectifiable K-computable curve, then there is a computable sequence (Q_n) of rational compact covers of C such that $d_H\left(\bigcup Q_n,\ C\right) < 2^{-n}$ and Q_n consists of rational neighborhoods. In each cover $\bigcup Q_n$ we can find the shortest polygon which straight through the whole area. This polygon is called a "diameter polygon" of the cover Q_n. The length l_n of this polygon is a lower bound of the length of C (possible with the error $\leq 2^{-n+1}$ because of the endpoints). Since C has a finite length l, the limit $l = \lim l_n$ is left computable because $l_n - 2^{-n+1} \leq l$ for all n.

By Theorems 2 and 3, any rectifiable R-, M- and N-computable curve has left computable length. Ko [8] constructed "monster curve" which is M-computable (even in polynomial time) with a non-computable length. The fact that the length of an M-computable curve is not necessarily computable follows also from the next result.

Theorem 4. *If C is a K-computable curve with a computable length, then C must be N-computable.*

Proof. Suppose that C is K-computable whose length l is a computable real number. Then there is a computable sequence (Q_n) of rational compact covers of C and a computable sequence (l_n) of rational numbers which converges to l effectively. Let q_n be the length of the "diameter polygon" of the area $\bigcup Q_n$.

For each $n \in \mathbb{N}$, we can find a sufficiently large index s_n such that $|q_{s_n} - l_{s_n}| \leq 2^{-n}$. Such an index s_n exists because both sequences (q_s) and (l_s) converge to the same limit $l(C)$. Suppose that p_n is a rational "diameter polygon" of the area $\bigcup Q_{s_n}$ and let f_n be the length-normalized parameterization of p_n. Then (f_n) is a computable sequence of computable functions which converges effectively to a computable function f. This limit function f is a length normalized parameterization of C. Therefore, the curve C is N-computable.

Notice that, if we consider only the curves of computable length, then the K-, R-, M- and N-computability of curves are equivalent. Now let C be an M-computable rectifiable curve which is not N-computable (by Theorem 8). This curve C is of course K-computable (Theorem 2). By the Theorem 4, C does not have a computable length. In fact, by a direct construction, we can show that even an N-computable curve may have a non-computable length.

Theorem 5. *There is an N-computable curve with a non-computable length.*

Proof. (Sketch) Let l be a left computable but not computable real number. There is an increasing computable sequence (l_n) of rational numbers which converges to l. Construct a computable sequence (p_n) of rational polygons such that the distance between p_n and p_{n+1} is bounded by $2^{(n+1)}$ and $l_n = l(p_n)$ for all n. Then we can choose a normalize computable parameterization f_n of p_n such that $|f_n(t) - f_{n+1}(t)| \leq 2^{-n}$ for each n. Therefore the limit curve $p := \lim p_n$ has a computable normalized parameterization $f := \lim f_n$ and hence is N-computable. The length of the N-computable curve p is l which is not computable.

3 A Technical Lemma

In this section we will show a technical lemma which will be used for the proofs of our main results in section 4. Remember that our goal is to separate the classes of curves by points covered by the curves. That is, we are interested in the points which are covered by curves from one class of curves but cannot be covered by any curves from another class of curves.

The next lemma shows a simple fact related to two curves which separates a curve from another one by a small neighborhood as long as the first curve is not a part of the second.

Lemma 1. *Let C and C' be two rectifiable, non-closed simple curves and let $g : [0;1] \to \mathbb{R}^2$ be a parametrization of C'. If we have $C' \cap U_z \neq \emptyset$ for all points $z \in C$ and all open neighborhoods U_z of z, then there exists an interval $[a;b] \subseteq [0;1]$ such that $g([a;b]) = C$.*

Proof. Suppose that C, C' are rectifiable, non-closed simple curves. If $C' \cap U_z \neq \emptyset$ for any point $z \in C$ and any open neighborhood U_z of z, then C must be a part of C', i.e., $C \subseteq C'$. Otherwise, by the compactness of C', we can find a point z in $C \backslash C'$ which has a positive distance from C' and hence some open neighborhood of z is disjointed from C which contradicts the hypothesis.

Because C' is a rectifiable simple curve, there exists an one-to-one parameterization $f : [0; 1] \to C'$. This parameterization f must be injective since C' is non-closed. Therefore the inverse function f^{-1} exists which is also continuous and maps particularly two end points of C to $u, v \in [0; 1]$. Suppose w.l.o.g. that $u < v$. Then we have $f([u; v]) = C$.

Let $h : [0; 1] \to [0; 1]$ be a continuous function defined by $h := f^{-1} \circ g$. Since $f([0; 1]) = C \subseteq C' = g([0; 1])$, we have $[u; v] \subseteq h([0; 1])$. By the continuity of h, there exist $a \in h^{-1}(u)$ and $b \in h^{-1}(v)$ such that $h([a; b]) = [u; v]$ (we suppose w.l.o.g that $a < b$). This implies immediately that $g([a; b]) = C$.

By Lemma 1, if a curve C is not contained completely in another curve C', then there exist a point z in C and a small neighborhood U_z around z such that U is totally disjoint from the curve C'. Particularly, if C is longer than C', then C cannot be completely contained in C'. If in addition C is a rational polygon and C' is a computable curve, then such a point z and the corresponding neighborhood U_z can be effectively found. That is, we have the following lemma.

Lemma 2. *Let C be a rational polygon and let C' be a computable curve. If the curve C is not contained completely in the curve C', then we can effectively find a rational point z on C and a rational neighborhood U_z of z such that $C' \cap U_z = \emptyset$.*

4 Point-Separability

This section will prove our main results that the four versions of computable curves introduced in the Definition 1 are different. More interestingly, we will see that four classes of computable curves cover even different point sets in the plane.

The difference between the R-computable curve and M-computable curve follows from a recent result of Gu, Lutz and Mayordomo [5]. They actually show that there is a polynomial time computable curve $\mathbf{\Gamma}$ which does not have any injective computable parametrization. In other words, any computable parametrization f of $\mathbf{\Gamma}$ must be retraced in the sense that $f(I_1) = f(I_2)$ for some disjoint subintervals $I_1, I_2 \subseteq [0; 1]$. Thus, $\mathbf{\Gamma}$ is R-computable but not M-computable.

Our main theorem shows actually even more. Namely, the four classes $\mathbb{C}_K$, $\mathbb{C}_R$, $\mathbb{C}_M$ and $\mathbb{C}_N$ of computable curves are not only different, they cover also different sets of points in the plane. More precisely, they are all "point-separable" in the following sense.

Definition 2. *Let $\mathbb{C}$ and $\mathbb{C}_1$ be classes of curves.*

1. *A point x is called a* $\mathbb{C}$-point *if it is a point of some curve C in the class $\mathbb{C}$.*

2. *The classes* $\mathbb{C}$ *and* $\mathbb{C}_1$ *are called* point-separable *if the sets of* $\mathbb{C}$*-points and* $\mathbb{C}_1$*-points are different.*

Remember that a function $f : [0,1] \to \mathbb{R}^2$ is computable if there is a Turing machine which computes f. Let (M_n) be an effective enumeration of all Turing machines M_n which compute the (possibly partial) functions $\varphi_n : [0,1] \to \mathbb{R}^2$. Then (φ_n) is an effective enumeration of functions including all total computable functions from $[0,1]$ to $\mathbb{R}^2$.

Theorem 6. *There exists a* K*-computable curve* C *and a point* z *on* C *such that* z *does not belong to any* R*-computable curve* C'*. In other words, the classes* $\mathbb{C}_K$ *and* $\mathbb{C}_R$ *are point-separable.*

Proof. (Sketch) We are going to construct a K-computable curve C and a point z which satisfy the condition mention in the theorem. By Definition 1, the K-computability of the curve C requires a computable sequence of finite sets (compact covers) of rational neighborhoods which approximates the curve C effectively. Such kind of compact covers can be easily constructed from rational polygons. Therefore, we need only to construct a computable sequence (p_n) of rational polygons which converges effectively to the curve C.

If C' is an R-computable curve, then C' has a computable parameterization $\varphi_i : [0,1] \to \mathbb{R}^2$, for some i, which is computed by the Turing machine M_i. Denote this curve simply by C_i. For the technical simplicity, let C_i be an empty set (curve) if M_i does not compute a total computable function. Therefore (C_i) is an effective enumeration of all R-computable curves. Thus, it suffices to construct the K-computable curve C and a point z on C which satisfy the following requirements:

$$R_i : \text{If } C_i \text{ has a finite length, then point } z \text{ does not belong to } C_i$$

To satisfy a single requirement R_i, we choose a straight line segment of the constructed polygon C. For simplicity, consider just the line segment J which connects the points $(0,0)$ and $(1,0)$. Simulate the computation of M_i to sufficient precision. If M_i computes a parameterization of the curve C_i which is not very close to J, then, by Lemma 1, we can find a point z on J and a neighborhood V of z such that $C_i \cap V = \emptyset$. If, on the other hand, C_i looks very close to J, then we have to look at more closely how the function φ_i possibly traces the segment J.

For any $q \in [0,1]$ and $\epsilon < l(J)/2$, we say that φ_i has a (q,ϵ)-sweep if the function φ_i approximately traces from $(q,0)$ to $(q+\epsilon,0)$, back to $(q,0)$ and finally passes $(q+\epsilon,0)$ forwardly again. As a parameterization of the curve C_i, φ_i can retrace some segment of C_i several times. However, it is impossible, for a fixed ϵ, that it has (q,ϵ)-sweep for all $q \in [0,1]$. If at some stage we find that φ_i cannot have a (q,ϵ)-sweep, then replace the linear segment from $(q,0)$ to $(q+2\epsilon,0)$ by the polygon which connects the points $(q,0)$, $(q+\epsilon,0)$, (q,δ) and $(q+2\epsilon,0)$ in the given order. Where $\delta > 0$ is a rational number which should be small enough to guarantee the K-computability of the constructed curve. After this change, the constructed new polygon C is different enough from C_i so that we can apply the Lemma 1 again to find a point z on C and a neighborhood V such that $C_i \cap V = \emptyset$.

In both cases, we have a neighborhood V such that every point in this neighborhood and in C satisfies the requirement R_i. Then, we can consider the segment of C in the neighborhood V to satisfy other requirements R_j for $j > i$. Formally we need a finite injury priority construction

Theorem 6 separates the K-computability from R-computability. In [5] it is shown that the R-computability and M-computability are different too, that is, there is an R-computable curve which does not have any injective computable parameterization at all. This can also be followed from our next more strong result.

Theorem 7. *There exists an R-computable curve C and a point z on C such that z does not belong to any M-computable curves C'. That is, the classes $\mathbb{C}_R$ and $\mathbb{C}_M$ are point-separable.*

Proof. (Sketch) We are going to construct an R-computable curve C and a point z on C which satisfy all the requirements

$$R_i : \text{If } \varphi_i \text{ is an injective parameterization of } C_i, \text{ then } z \text{ is not on } C_i.$$

where (φ_i) is a computable enumeration of all (possibly partial) computable functions $\varphi_i : [0,1] \to \mathbb{R}^2$. The construction uses again the finite injury priority method. We explain the rough idea how to satisfy a single requirement R_i only.

Take a linear segment of the constructed polygon C. For simplicity, consider just the line segment J from the point $(0,0)$ to $(1,0)$ with a parameterization φ which sweeps between these points. That is, φ goes from $(0,0)$ to $(1,0)$ first, then back to $(0,0)$ and finally goes through $(1,0)$ again. This is allowed because we want to construct an R-computable curve C.

Simulate the computation of M_i which computes the function φ_i to sufficient precision. If φ_i is an injective parameterization of C_i, then consider the following cases:

Case 1. C_i is not close to J at all, then we are done by the Lemma 1.

Case 2. C_i closely passes the segment J only once. In this case, alter the segment J by a Z-sweep of height δ which is a polygon connecting the points $(0,0)$, $(1,\delta)$, $(0,-\delta)$ and $(1,0)$ in the given order. Where $\delta > 0$ is a sufficiently small rational number. Then the Lemma 1 can be applied.

Case 3. C_i is close to J and also has Z-sweeps near J. Suppose that the minimal height of all these Z-sweeps is $\epsilon > 0$. Then replace the segment J by a Z-sweep of a height δ such that $\delta < \epsilon/2$. After that we can apply the Lemma 1.

In all three cases, according to Lemma 1, we can fine a z on C and a neighborhood V of z such that $C_i \cap V = \emptyset$. Thus, the segment of C in the neighborhood V can be used to satisfy other requirements R_j for $j > i$. The priority technique guarantees that all requirements can be satisfied simultaneously.

Finally, we want show the difference between M- and N-computability of curves.

Theorem 8. *There exists an M-computable curve C and a point z on C such that z does not belong to any N-computable curves C'. That is, the classes $\mathbb{C}_M$ and $\mathbb{C}_N$ are point-separable.*

Proof. (Sketch) We use priority technique again to construct an M-computable curve C and a point z on C such that the following requirements are satisfied

$$R_i : \text{If } \varphi_i \text{ is a length-normalized parameterization of } C_i, \text{ then } z \text{ is not on } C_i.$$

Suppose that C_i is an N-computable curve and φ_i is a length-normalized parameterization of C_i. Choose a linear segment J of already constructed curve C. For simplicity, let J be the line segment connecting the points $(0, 0)$ and $(1, 0)$. Compute φ_i to sufficient precision. If C_i is not close to the segment J, then we can apply the Lemma 1 directly. Otherwise, suppose that C_i is very close to the segment J. That is, there are $t_1, t_2 \in [0, 1]$ such that the segment $\varphi_i([t_1, t_2])$ almost coincides with J. Then compute the middle point $\varphi_i((t_1 + t_2)/2)$ of the segment $\varphi_i([t_1, t_2])$ and check if it is close to the middle point of J. If it is not the case, then φ_i is not length-normalized and we are done. Otherwise, double the length of the first half of the segment J (i.e. the part from $(0, 0)$ to $(1/2, 0)$) by introducing small zigzags. This makes the new segment different enough from the curve C_i and hence we can apply the Lemma 1 to find a point on C and a neighborhood V of z such that $V \cap C_i = \emptyset$. Therefore, the standard priority construction works.

Notice that an N-computable curve has a computable parameterization which traces the curve in one direction and with a constant speed. Thus, Theorem 8 shows that some curve describes the computable particle motion in one direction but the speed of the motion cannot be constant.

Remark: In the proofs of above three theorems, we always choose a linear segment J which connects the points $(0, 0)$ and $(1, 0)$. This choice may help reader to understand how a new polygon should be constructed. However, there is a drawback for this choice of J that we cannot see how to guarantee that the constructed curve has a finite length. So in more formal constructions, we should choose the segment J with much short length so that the new curve increases the length only in a very small portion. This guarantees that the constructed curve is rectifiable.

References

1. V. Brattka and K. Weihrauch. Computability on subsets of Euclidean space I: Closed and compact subsets. *Theoretical Computer Science*, 219:65–93, 1999.
2. M. Braverman and S. Cook. Computing over real numbers: Foundation for scientific computing. *Notics of AMS*, 53(3):318–329, 2006.
3. G. A. Edgar. *Measure, Topology, and Fractal Geometry*. Undergraduate Texts in Mathematics. Springer, 1995.
4. X. Gu, J. H. Lutz, and E. Mayordomo. Points on computable curves. In *Proceedings of FOCS 2006*, pages 469–474. IEEE Computer Society Press, 2006.
5. X. Gu, J. H. Lutz, and E. Mayordomo. Curves that must be retraced, (Preprint) 2008.
6. C. Jordan. *Cours d'analyse de l'Ecole Polytechnique*. Publications mathématiques d'Orsay, 1882.
7. K.-I. Ko. *Complexity Theory of Real Functions*. Progress in Theoretical Computer Science. Birkhäuser, Boston, MA, 1991.
8. K.-I. Ko. A polynomial-time computable curve whose interior has a nonrecursive measure. *Theoretical Computer Science*, 145:241–270, 1995.
9. G. Peano. Sur une vourbe, qui remplit toute une aire plane. *Mathematische Annalen*, 36(1):157–160, 1890.
10. M. B. Pour-El and J. I. Richards. *Computability in Analysis and Physics*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1989.
11. K. Weihrauch. *Computable Analysis, An Introduction*. Springer, Berlin Heidelberg, 2000.
12. K. Weihrauch and X. Zheng. Computability on continuous, lower semi-continuous and upper semi-continuous real functions. In T. Jiang and D. Lee, editors, *Computing and Combinatorics*, volume 1276 of *Lecture Notes in Computer Science*, pages 166–175, Berlin, 1997. Springer. Third Annual Conference, COCOON'97, Shanghai, China, August 1997.
13. N. Zhong and K. Weihrauch. Computability theory of generalized functions. *J. ACM*, 50(4):469–505, 2003.