Effective uniformity versus limiting recursion in sequential computability of a function

Mariko Yasugi *

February 26, 2008

Contents

Abstract

The major objective of this article is a refinement of treatments of the mutual relationship between two notions of sequential computability of a function which is possibly Euclidean-discontinuous, one using limiting recursion and one using effective uniformity. We also speculate on these methods from a mathematician's viewpoint.

1 Introduction

The objective of this article is to distill the general situation in which two methods of computing some Euclidean-discontinuous functions become equivalent. Those are the methods developed respectively on "effective uniformity" and on "limiting recursion." In so doing, we speculate on the two notions of "sequential computability."

*e-mail: yasugi@cc.kyoto-su.ac.jp This work has been supported in part by Research Grant from KSU No.113 and No.114.

The domain of our discourse is the real line or a subinterval of the real line as well as some (possibly Euclidean-discontinuous) functions on it.

It has been an old practice to review mathematics from the algorithmic viewpoints. It is based on the recursive function. On the continuum, a computable object is approximated by a recursive sequence of rational numbers or a recursive sequence from a discrete structure with a recursive modulus of convergence (effective approximation).

Investigation of computability on the continuum is based on the "computable sequence of reals." Computability of real functions was originally defined for continuous functions (cf. Chapter 0 in [5], for example). A continuous real function is called computable if it maps any computable sequence of real numbers to a computable sequence (sequential computability), and it has a recursive modulus of continuity (effective continuity). The sequential computability is required for the following reason. In order to claim that a function be computable, one must have a general algorithm to compute the value of that function for any computable real number. In order to secure it, it is known to be sufficient to assume the sequential computability.

One might expect that one can compute the values of a function without the assumption of (effective) continuity, but it is not so. A simple function such as the integer-part function $[x]$ (called also the Gaußian function), which jumps at each integer but is continuous (constant) on the interval of two adjacent integers, does not preserve sequential computability ([9]).

Such a problem has been discussed in [9], [8], [10]. In fact, we easily compute such a function at any (computable) point. We know that, for any integer n, on the interval $[n, n+1)$, $[x]$ satisfies the requirements on the computability of a continuous function (and the value is n). It is all too easy! Of course there are many Euclidean-discontinuous functions which are far more complex than $[x]$ and still maintain some notion of sequential computability.

We wished to express such an intellectual activity of the human mind in a mathematical language, and have proposed two such treatments: one expressing such a computation in terms of "limiting recursive functions" of natural numbers ([9], [8]), and one in terms of "changing topology" of the domain of a function, thus regarding a (Euclidean-) discontinuous function like $[x]$ as continuous in the new topology so that we can conceive the computability of a function as that of a continuous function ([6]).

Both methods have been well developed and applied to many examples of Euclidean-discontinuous functions ([7], [9], [11], [12], [14]). Analyzing these individual treatments, we have pinpointed a *general framework* under which two notions of sequential computability concide. The framework will be introduced as the assumption $[\mathcal{A}]$ and the condition $[\mathcal{C}]$ in Section 3. In this framework, the equivalence of the two notions (methods) is mathematically established. On the other hand, each of them can be interpreted as expressing a certain human intellectual activity of a same phenomenon from different viewpoints.

It is notable that this equivalence holds notwithstanding that the two approaches are methodologically quite different. This disparity has forced the author to speculate on the meaning of the two approaches. As for the limiting recursion method, we have already discussed its signigicance and problems in [8]. Here we will put emphasis on the contrast between the two methods.

To us, the method of uniform spaces is the most natural and intuitive. It represents the freedom that our mind enjoys.

We present very briefly in Section 2 some basics of computability on the continuum. No mathematical details will be supplied except for what is necessary for our present purpose. For basics of computability in analysis, we refer the reader to [5] and [13].

Our interest lies in a real function which is Euclidean-discontinuous but is farily tame so that one can attribute to it some kind of computability property. We will hence set up a framework to meet our purpose in Section 3. Two (extended) notions of "sequential computability" of a function in our framework are then formulated in Section 4.

Our main mathematical result, the equivalence of the two notions of sequential computability in our general framework, is proved in Section 5.

An example of computation in the respective method according to our framework is explained in Section 6. The article is concluded with a speculation on the limiting recursion versus effective uniformity in Section 7.

We have also worked on a sequence of uniformities and its limit; some mathematical results as well as the significance of such a theory are seen in [10].

2 Preliminaries

We will list some of the basic notions and notations which are just necessary to our discussion.

Definition 2.1 (Computable real sequence) (i) A sequence of real numbers $\{x_m\}$ is called **R**-*computable* (computable in the Euclidean topology) if the following hold ([5]).

(1) There is a recursive (double) sequence of rational numbers $\{r_{mn}\}$ which approximates $\{x_m\}$.

(2) There is a recursive modulus of convergence of $\{r_{mn}\}$ to $\{x_m\}$, say β, that is,

$$n \geq \beta(m,p) \rightarrow |x_m - r_{mn}| < \frac{1}{2^p}.$$

In such a case, and in any similar situation, we say that $\{r_{mn}\}$ *effectively* approximates (converges to) $\{x_m\}$.

(ii) A number-theoretic function η is called after Gold [2] *limiting recursive* if it is defined to be the limit of a recursive function, that is, there is a recursive function h satisfying $\eta(p) = \lim_n h(p,n)$ if the limit exists. (In fact, a function which is recrusvie in a limiting recursive function will also be called limiting recursive.)

(A formal treatment of mathematics with limiting recursion has been developed in [3].)

(iii) If in (2) of (i) above the recursive β be replaced by a limiting recursive η, then we say that $\{x_m\}$ has a *weak representation* by $\{r_{mn}\}$ and η.

We will define the effective uniform topology on an arbitrary non-empty set X although the universe of our discourse is the set of real numbers $\mathbf{R}$ or its subinterval. $\mathbf{N}$ will denote the set of positive integers $\{1, 2, 3, \cdots\}$. (We have employed the definition of (classical) uniformity in [4].)

Definition 2.2 (Effective uniformity:[6]) A uniformity $\{U_n\}$ on X is called an *effective uniformity* if U_n is a map from X to the powerset of X, and there are recursive functions $\alpha_1, \alpha_2, \alpha_3$ which satisfy the following.

$$\forall x \in X. \bigcap_n U_n(x) = \{x\}.$$

$$\forall n, m \in \mathbf{N} \forall x \in X. U_{\alpha_1(n,m)}(x) \subset U_n(x) \cap U_m(x).$$

$$\forall n \in \mathbf{N} \forall x, y \in X. x \in U_{\alpha_2(n)}(y) \to y \in U_n(x).$$

$$\forall n \in \mathbf{N} \forall x, y, z \in X. x \in U_{\alpha_3(n)}(y) \wedge y \in U_{\alpha_3(n)}(z) \to x \in U_n(z).$$

It is known that $\mathcal{U} = \langle X, \{U_n\} \rangle$ is a uniform toplogical space with $\{U_n(x)\}$ as the system of fundamental neighborhoods.

Effective convergence and the "computability structure" on $\mathcal{U} = \langle X, \{U_n\} \rangle$ are also defined in [6].

Definition 2.3 (Effective $\mathcal{U}$-convergence:[6]) A double sequence $\{r_{mn}\}$ from X is said to *effectively $\mathcal{U}$-converge* to a sequence $\{x_m\}$ if there is a recursive function γ satisfying $\forall m \forall n \forall k \geq \gamma(m, n). x_{mk} \in U_n(x_m)$. We also say that $\{x_m\}$ is the effective $\mathcal{U}$-limit of $\{r_{mn}\}$ and γ.

If in this definition γ is replaced by a limiting recursive function η, then we say that $\{x_m\}$ has a weak $\mathcal{U}$-representation by $\{r_{mn}\}$ and η.

Definition 2.4 ($\mathcal{U}$-computable sequences) Let X be $\mathbf{R}$. A sequence of real numbers $\{x_m\}$ is called *$\mathcal{U}$-computable* if $\{x_m\}$ is the effective $\mathcal{U}$-limit of a recursive sequence of rational numbers $\{r_{mn}\}$ and a recursive function γ.

The definition can be extended to any multiple sequence of real numbers.

A real number x is called $\mathcal{U}$-computable if $\{x, x, x, \cdots\}$ is.

The set of $\mathcal{U}$-computable sequences (multiple sequences included) is closed under any recursive re-enumeration and the effective $\mathcal{U}$-limit.

Proposition 2.1 (Recursive sequence of rationals) A recursive sequence of rationals (hence of natural numbers) is $\mathcal{U}$-computable.

It is known that the set of $\mathcal{U}$-computable sequences has a nice property, but we will not go into details. In all the examples we have been concerned with, a $\mathcal{U}$-computable sequence of real numbers is $\mathbf{R}$-computable, but not conversely, while a single real number is $\mathcal{U}$-computable if and only if it is $\mathbf{R}$-computable.

3 Framework

We will here set up a framework in order to attain our purpose. We will first place an overall assumption.

Assumption $[\mathcal{A}]$ We work in an effective uniform space $\mathcal{U} = \langle \mathbf{R}, \{U_n\}\rangle$, and assume that (***) $\mathbf{R}$-computable numbers and $\mathcal{U}$-computable numbers concide and (***) that every $\mathcal{U}$-compurable sequence is $\mathbf{R}$-computable.

We further assume a condition on $\mathcal{U}$, denoted by $[\mathcal{C}]$.

Condition $[\mathcal{C}]$ on $\mathcal{U}$ Given an $\mathbf{R}$-computable sequence $\{x_m\}$, there is a $\mathcal{U}$-computable sequence $\{z_{mp}\}$ and a limiting recursive function ν such that $\{x_m\}$ has a weak representation by $\{z_{mp}\}$ and ν (cf. Definition 2.3), that is,

$$\forall m, n \forall p \geq \nu(m, n). z_{mp} \in U_n(x_m). \tag{1}$$

Proposition 3.1 If $\{x_m\}$ is $\mathcal{U}$-computable, then ν can be recursive, since by definition $z_{mp} = x_m$ will do.

Definition 3.1 (Framework) The *framework* of our study of real functions consists of $[\mathcal{A}]$ and $[\mathcal{C}]$.

Note (i) The condition $[\mathcal{C}]$ signifies that an $\mathbf{R}$-computable sequence may not be $\mathcal{U}$-computable, but it is "almost" $\mathcal{U}$-computable.

(ii) All the uniform spaces we have dealt with satisfy the assumption $[\mathcal{A}]$ and the condition $[\mathcal{C}]$ ([6], [7], [10], [11], [12]).

4 Sequential computabilities

We subsequently define two notions of sequential computability of a real function within the framework of Section 3. Although the definitions are stated for the function whose domain is the whole real line, the definitions can be easily modified to any interval with computable end-points.

Definition 4.1 (Sequential computability of a function:[11]) (i) ($\mathcal{L}$-sequential computability, relative to ν) f is $\mathcal{L}$-sequentially computable if, for any $\mathbf{R}$-computable sequence of real numbers $\{x_m\}$, the sequence of function values $\{f(x_m)\}$ is weakly represented, that is there exist a recursive sequence of rational numbers $\{s_{mn}\}$ and a function η which is recursive in ν (as claimed in $[\mathcal{C}]$) so that $\{f(x_m)\}$ is $\mathcal{U}$-approximated by $\{s_{mn}\}$ with a limiting recursive modulus of convergence η.

(ii) ($\mathcal{U}$-sequential computability) f is called $\mathcal{U}$-sequentially computable if, for any $\mathcal{U}$-computable sequence of real numbers $\{x_m\}$, the sequence of function values $\{f(x_m)\}$ is $\mathbf{R}$-computable.

Note It should be noted that the input sequence $\{x_m\}$ is computable in the above definition. Our interest is to see how a function (its values) behaves for computable inputs.

5 Equivalence results

We will prove the equivalence of two notions of sequential computability of a real function as has been defined in the previous section in the general framewok of Section 3. The proof is similar to the one in [11] for a special case.

Theorem 1 (From $\mathcal{L}$-sequential computability to $\mathcal{U}$-sequential computability) If f is $\mathcal{L}$-sequentially computable (relative to ν), then f is $\mathcal{U}$-sequentially computable.

Proof of Theorem 1 Suppose f is $\mathcal{L}$-sequentially computable, and let $\{x_m\}$ be $\mathcal{U}$-computable. Then, by $[\mathcal{A}]$, $\{x_m\}$ is **R**-computable. So, by $\mathcal{L}$-sequential computability, there is a recursive sequence of rational numbers $\{t_{mp}\}$ and a function δ which is recursive in ν (cf. $[\mathcal{C}]$) satisfying

$$\forall m, p \forall q \geq \eta(m,p). |f(x_m) - t_{mq}| < \frac{1}{2^p}.$$

By virtue of Proposition 3.1, one can take a recursive ν for such a sequence $\{x_m\}$, and so we can take a recursive η so that $\{f(x_m)\}$ is **R**-computable by $\{t_{mq}\}$ and η, and hence f is $\mathcal{U}$-sequentially computable.

Notice that so far we have not assumed any kind of continuity on the function f. The converse of Theorem 1 will be proved under a weak kind of effective $\mathcal{U}$-continuity, which claims that a function is fairly well-behaved.

Definition 5.1 (Relatively effectively $\mathcal{U}$-continuous function:[6]) A function f is called *relatively effectively $\mathcal{U}$-continuous* if the following holds. For any $\mathcal{U}$-computable sequence $\{x_m\}$, there is a recursive function $\gamma(m,p)$ such that $y \in U_{\gamma(m,p)}(x_m)$ implies $|f(y) - f(x_m)| < \frac{1}{2^p}$.

Theorem 2 (From $\mathcal{U}$-sequential computability to $\mathcal{L}$-sequential computability) If f is $\mathcal{U}$-sequentially computable and relatively effectively $\mathcal{U}$-continuous, then f is $\mathcal{L}$-sequentially computable.

Proof of Theorem 2 Suppose f is $\mathcal{U}$-sequentially computable. Let $\{x_m\}$ be an **R**-computable sequence of real numbers. Then by the condition $[\mathcal{C}]$, there is a $\mathcal{U}$-computable sequence $\{z_{mp}\}$ and a limiting recursive function ν as in Equation (1) of Section 4. Since $\{z_{mp}\}$ is $\mathcal{U}$-computable, $\{f(z_{mp})\}$ is **R**-computable, and hence it is approximated by a recursive sequence of rational numbers $\{s_{mql}\}$ and a recursive function β, that is,

$$l \geq \beta(m,q,n) \to |f(z_{mq}) - s_{mql}| < \frac{1}{2^n}. \tag{2}$$

Then define a recursive sequence of rational numbers $\{t_{mn}\}$ by

$$t_{mn} = s_{mn\beta(m,n,n)}. \tag{3}$$

Recall that, since f is relatively effectively $\mathcal{U}$-continuous,

$$z_{mq} \in U_{\gamma(m,n)}(x_m) \to |f(z_{mq}) - f(x_m)| < \frac{1}{2^n}. \tag{4}$$

In (1), put $n = \gamma(m,n)$. Then we have

$$q \geq \nu(m, \gamma(m,n)) \to z_{mq} \in U_{\gamma(m,n)}(x_m). \tag{5}$$

Combining Equations (4) and (5) as well as Equations (2) and (3), we obtain that, if $q \geq \max(\nu(m,\gamma(m,n)), n)$, then since $\frac{1}{2^q} \leq \frac{1}{2^n}$,

$$|f(x_m) - t_{mq}| \leq |f(x_m) - f(z_{mq})| + |f(z_{mq}) - t_{mq}| < 2\frac{1}{2^n}. \tag{6}$$

So, with $\eta(m,n) = \max(\nu(m, \gamma(m,n+1)), n+1)$, $\{f(x_m)\}$ is weakly represented by $\{t_{mn}\}$ and η since η is recursive in ν.

Note (i) It is worth noticing that separability of the uniform space is not assumed in the equivalence proofs.

(ii) Theorem 2 is proved under the relatively effective $\mathcal{U}$-continuity of f. Relatively effective $\mathcal{U}$-continuity is a preliminary condition of effective $\mathcal{U}$-continuity of a function, but is much weaker than effective $\mathcal{U}$-continuity (cf. [6]). Since the uniformity has been introduced to make a function continuous in the new topology, relative $\mathcal{U}$-continuity is a reasonable condition to be assumed.

6 Example of seuential computation

There are many examples whose sequential computabilities have been successfully treated; among them are the Gaußian and the Rademacher functions ([6], [9], [14]), and Brattka's Fine continuous function ([1]). Brattka's function is an example of a Fine continuous but not locally uniformly Fine continuous function.

Here we explain how the requirements of the framework are met and how sequential computability can be established with an easy example of the Gaußian function $[x]$, by partly reviving the corresponding content in [9].

Recall that the value $[x]$ is an integer, a computable number, for any real number x. There is thus no sense in questioning about the computability of the function value at a single point x. It *is* computable. With a sequence of values, it takes on a new aspect.

We define an $\mathbf{R}$-computable sequence of rational numbers, $\{x_m\}$, which is not recursive. Let $a : \mathbf{N} \to \mathbf{N}$ $(n = 1, 2, 3, \cdots)$ be a recursive injection whose range is not recursive. Consider the sequence of reals $\{x_m\}$ defined by

$$x_m = \left\{ \begin{array}{ll} 1 - \frac{1}{2^l} & \text{if } m = a(l) \text{ for some } l, \\ 1 & \text{otherwise.} \end{array} \right\}$$

$\{x_m\}$ is computable since it is effectively approximated by $\{r_{mk}\}$ defined below with a recursive modulus of convergence, say α.

$$r_{mk} = \left\{ \begin{array}{ll} 1 - \frac{1}{2^l} & \text{if } m = a(l) \text{ for some } l \leq k, \\ 1 & \text{otherwise.} \end{array} \right\}$$

From the definition we have

$$[x_m] = \left\{ \begin{array}{ll} 0 & \text{if } m = a(l) \text{ for some } l, \\ 1 & \text{otherwise.} \end{array} \right\}$$

Now, suppose $\{[x_m]\}$ were a computable sequence. Then, it can be shown (similarly to Example 4, Chapter 0 of [5]) that the range of a would be recursive, yielding a contradiction. So, $\{[x_m]\}$ cannot be an $\mathbf{R}$-computable sequence.

This counter-example assures us of the following fact: the Gaußian function does not necessarily preserve $\mathbf{R}$-sequential computability.

With the function $[x]$, we associate a uniform space $\mathcal{U} = \langle \mathbf{R}, \{U_n\} \rangle$ by mutually isolating the half-open intervals $[l, l+1)$ for all integer l. Namely,

$$U_n(x) = (x - \frac{1}{2^n}, x + \frac{1}{2^n}) \cap [l, l+1) \quad \text{if} \quad x \in [l, l+1).$$

Corollary 1 A $\mathcal{U}$-computable sequence is $\mathbf{R}$-computable, and hence the assumption $[\mathcal{A}]$ is satisfied in $\mathcal{U}$.

Proposition 6.1 The sequence $\{x_m\}$ deinfed above is not $\mathcal{U}$-computable.

Proof Suppose $\{x_m\}$ were $\mathcal{U}$-sequentially computable. Then there are a recursive sequence of rational numbers, say $\{s_{mk}\}$ and a recursive function γ such that $\forall k \geq \gamma(m,n). s_{mk} \in U_n(x_m)$. Then

$$s_{m,\gamma(m,1)} \in [0,1) \leftrightarrow x_m < 1,$$

and since the left-hand side is effectively decidable (recursive in m), so is the right-hand side. But then the range of the function a would be recursive, contradicting the property of a.

Proposition 6.2 The sequence $\{x_m\}$ has a weak $\mathcal{U}$-representation and hence satisfies the condition $[\mathcal{C}]$.

Proof As for $\{z_{mk}\}$, it suffices to take $\{r_{mk}\}$. Since this is a recursive sequence of rational numbers, it is $\mathcal{U}$-computable (Proposition 2.1). Define a function h as follows.

$$h(m,k) = 1 \quad \text{if} \quad \forall l \leq k. r_{ml} = 1;$$

$$h(m,k) = k_0 + 1 \quad \text{if } k_0 \text{ is the least } l \leq k. r_{ml} < 1$$

h is recursive, and it is easy to see that $\nu(m) = \lim_k h(m,k)$ exists. $\nu(m) = 1$ or $= k_0$, and ν serves as the modulus of convergence of $\{r_{mn}\}$ to $\{x_m\}$.

Now, an attempt of computing $\{[x_m]\}$ goes very roughly like this (cf. [9],[8]): It can be decided that $0 < x_m < 2$ for all m. In order to determine whether $0 < x_m < 1$ or $x_m \geq 1$, we define a recursive sequence of rational numbers (integers as a matter of fact) $\{N_{mp}\}$ as follows.

$$N_{mp} = \left\{ \begin{array}{ll} 1 & \text{if } r_{m\alpha(m,p)} \geq 1 - \frac{1}{2^p}, \\ 0 & \text{if } r_{m\alpha(m,p)} < 1 - \frac{1}{2^p}. \end{array} \right\}$$

It can be easily shown that there is a limiting recursive function η (recursive in ν) so that $\{[x_m]\}$ is weakly represented by $\{N_{mp}\}$ and η. This indicates a way to establish the $\mathcal{L}$-sequential computability of $[x]$ (cf. [9]). Since $\{x_m\}$ is not $\mathcal{U}$-computable by Proposition 6.1, we need not compute $\{[x_m]\}$ in $\mathcal{U}$.

Proposition 6.3 The function $[x]$ is relatively effectively $\mathcal{U}$-continuous.

The proposition can be easily proved if one notices that, for any $\mathcal{U}$-computable $\{x_m\}$, $x_m \in [l, l+1) \leftrightarrow r_{m\gamma(m,n)} \in [l, l+1)$ and that the right-hand side is decidable. This fact also implies the $\mathcal{U}$-sequential computability of $[x]$.

7 Limiting recursion versus effective uniformity

The definition of $\mathcal{L}$-sequential computability and that of $\mathcal{U}$-sequential computability appear mutually quite different. Let us see this with the example $[x]$. With $\mathcal{L}$-sequential computability, given any $\mathbf{R}$-computable sequence $\{x_m\}$, start innocently computing the values $\{[x_m]\}$, step by step for $p = 1, 2, 3, \cdots$, trying to see if a recursive condition $R(m,p) \equiv r_{m\alpha(m,p)} < 1 - \frac{1}{2^p}$ is satisfied. One either has a luck to hit a p satisfying $R(m,p)$ or keep checking. In any case, if one can go ad infinitum, then the computation result can be settled. "Observing the computation from the infinity" corresponds to accepting the limit of a recursive process. On the other hand, with the uniform topology, one does not even attempt to compute the function value for certain $\mathbf{R}$-computable sequences such as $\{x_m\}$ in Section 6.

The (mathematical) equivalence of the two notions of sequential computability (under a certain condition) therefore needs some speculation.

Let us first observe the limiting recursion method. Here one attempts to compute the function values mechanically. The input values for a function are supplied with a recursive sequence of rationals and a recursive modulus of convergence, but the outputs, viz. the function values, are represented by a recursive sequence of rationals with a limiting recursive modulus of convergence which may not be recursive. This is discussed in [9] and [8] in detail.

The good of this method lies in its simplicity. The only tool in need beyond the recursive function is the limit of a recursive function. The function value at a jump point is represented with a recursive sequence of rational numbers. At each step of computation one is approaching the right value, and one knows that eventually one gets the proper value, though not knowing when. It may be a bit tantalizing if one wishes to know where one is now. It is, however, assuring and in a way sufficient to know that one stands on the right track. It is along the straight extension of the computation of continuous functions; only the speed of convergence needs limiting recursion instead of recursion. The idea and the knowledge are simple and easily understood. No extra knowledge is required. This is its advantage. A disadvantage is that it does not represent the mental activity of a mathematician computing the function value at a point of discontinuity.

In the method of effective uniformity, with each function is associated a uniform space in which it becomes continuous. The theory of computability structure in such a space is developed, and a function is defined to be computable as a continuous function in this topology. We can thus adhere to the computability problem of a continuous function. A wide range of functions can be regarded as computable with this method. Except for recursive functions, we do not need any special tool beyond ordinary mathematical knowledge. For each instance of a function, we only need to associate a uniformity by isolating the points of discontinuity or intervals determined by the points of discontinuity.

This approach is also quite intuitive. The values of a function in each interval (possibly consisting of a singleton) of continuity can be computed as in the case of a Euclidean-continuous function. Only one must judge, for example, whether $x \in [0,1)$ or $x \in [1,2)$.

In the effective uniformity method, it is important to notice that one can recognize the jump points "intuitively". In the case of $[x]$, these are integers, and they are the most obvious points that a humen being can recognize on the real line. Consider another example. Let τ denote the function which coincides with the tan function where tan is defined, and takes the value 0 where tan is not defined. The computation of τ at a jump point like $\frac{\pi}{2}$, which should cause a problem in a mechanical computation, is the easier part: the value is 0. Isolating the jump points (and the computation of the function values at them), which is not a decidable procedure, is thus intuitively appealing. The effective uniformity method thus describes the human mental activity of computing a Euclidean-discontinuous function.

We need not attempt to judge effectively if a real number is a jump point. The judgement is taken care of in the definition of the effective uniformity; $U_n(\frac{\pi}{2}) = \{\frac{\pi}{2}\}$ for τ, for example. It is a mathematical activity, and we are at liberty to do that. In that sense, the theory of effective uniformity yields a "supple method" (according to the phrasing of Nakatogawa) for computing a discontinuous function. It reflects the flexibility that a mathematician wishes to experience.

Incidentally we may consider isolating some points or intervals as similar to the type system of a program language. We leave judgement of the type of data outside a program itself.

Acknowledgement The author is indebted to K. Nakatogawa for many hours of discussion on the notion of computability and to S. Hayashi for having introduced Gold's theory to her. Her gratitude goes also to T. Mori and Y. Tsujii for mathematical collaborations.

References

[1] Brattka, V., *Some Notes on Fine Computability*, JUCS, 8(2002), 382-395.

[2] Gold, E.M., *Limiting recursion*, JSL, 30-1 (1965), 28-48.

[3] Hayashi, S., Nakata, M., *Towards Limit Computable Mathematics*, International Workshop, Type 2000, Selected Papers, LNCS 2277, Springer (2002) 125-144.

[4] Kawada, Y., Mimura, Y., *Outline of Modern Mathematics* (in Japanese), Iwanami Shoten (1965).

[5] Pour-El, M.B., Richards, J.I., *Computability in Analysis and Physics*, Springer-Verlag (1989).

[6] Tsujii, Y., Yasugi, M., Mori, T., *Some properties of the effectively uniform topological space*, Computability and Complexity in Analysis, LNCS 2064, Springer (2001) 336-356.

[7] Tsujii, Y., Yasugi, M., Mori, T., *Sequential computability of a function - Diagonal space and limiting recursion -* , ENTCS 120(2005),187-199.

[8] Yasugi, M., *Limiting computability of discontinuous functions-significance and problems*, J. of Japan Assoc. for Phil. of Sci., vol.30,no2(2003), 13-18 (in Japanese).

[9] Yasugi, M., Brattka, V., Washihara, M., *Computability aspects of some discontinuous functions*, Scientiae Mathematicae Japonicae (SCMJ) Online, vol.5 (2001), 405-419.

[10] Yasugi, M. Mori, T., Tsujii, Y., *The effective sequence of uniformities and its limit: as a methodology in computable analysis*, Annals of Philosophy of Science, vol.15, no.2 (2007), 99-121(47-69).

[11] Yasugi, M., Tsujii, Y., *Computability of a function with jumps-Effective uniformity and limiting recursion-*, Topology and its Applications (Elsevier Science), 146-147(2005), 563-582.

[12] Yasugi, M., Tsujii, Y., Mori, T., *Sequential computability of a function - Effective Fine space and limiting recursion -*, JUCS 11-12(2005), 2179-2191; http://www.jucs.org/

[13] Yasugi, M., Washihara, M., *Computability structures in analysis*, Sugaku Expositions (AMS) 13 (2000) no.2, 215-235.

[14] Yasugi, Y., Washihara, M., *A note on Rademacher functions and computability*, Words, Languages and Combinatorics III, World Scientific (2003) 466-475.