数理解析研究所講究録
1169 巻 2000 年 84-91

How to understand the computability aspects of step functions

Mariko Yasugi *

1 Objective

In the following, I assume basic knowledge in recursive functions and computability properties of real numbers, of sequences of real numbers, of continuous real functions and of basic facts in computability structures in Banach spaces. References [3] and [12] will be useful in obtaining necessary knowledge.

Our objective in this report is to give a mathematical expression to the action of drawing a graph which is not necessarily connected. (I insist on mathematical treatment, and will not go into algorithmic foundations. I would like also to emphasize that I will confine subsequent discussions to "our way" of such an effort, that is, of myself as well as my colleagues. I warn the reader that ehre are many other ways to treat such a subject, but I do not mention them here in order to avoid deviation.)

The reason of need for such considerations is the following. (Here I consider only unary real functions.) In drawing a graph of a function, one first computes and plots some real numbers on the x-axis, usually some fractions. One then computes the values of the function at such points and plots the corresponding points in the plane. Finally, one connects these points as smoothly as possible. Computer graphics would do similarly.

In sucn an action, computing the function value at a point is essential. (Here, *computation* means approximating computation of arbitrary precision.) For a continuous function, the notion of *computability* has been traditionally established, and, for a computable function, it is theoretically possible to compute its value at a computable input. On the other hand, it is a common practice to draw a graph of a discontinuous function. It has also been systematically studied in [3] how to view a discontinuous function to be *computable*. Their tool is the Banach space, and they endow some functions which are not necessarily continuous the notion of *computability* as points in a space.

A function in a Banach space is computable if it is *effectively* approximated by a *computable sequence* of continuous functions with respect to the *norm* of the space. Pour-El and Richards have formulated this notion in terms of an axiom set.

*Faculty of Science, Kyoto Sangyo University: yasugi@cc.kyoto-su.ac.jp.

Stimulated by [3], I and some of my colleagues have investigated various frameworks in analysis for use in dealing with discontinuous functions which nevertheless can be regareded as having some algorithmic attributes. Examples of such frameworks are the Fréchet space, the metric space and the uniform topological space. See [4], [2], [6], [12], [5], [9], [10], [1], [11], [13].

Note The bibliography at the end of this article is not meant at all comprehensive. In fact, I resetrict the references only to those in which I have myself been involved or which I have closely followed. For other related works, the reader is invited to the comprehensive list of references in the area at

http://www.informatik.fernuni-hagen.de/cca/ publications/bibliography.html.

Subsequently, I present a brief exposition of our speculations on a simple and familiar step function, the Gaußian function. I will also mention the Rademacher functions, a sequence of step functions, which are important in Walsh analysis. These functions are simple and innocent looking, but are good examples in distilling the algorithmic features of discontinuous functions. If one can set up a framework to talk about the computability properties of such functions, one can apply it to many significant functions. There is another reason why I take up these functions. Namely, step functions are essential tools for digital analysis, and so it is important to discuss computability properties of such functions.

My belief is that examples to be used for analyzing certain properties be simple and familiar looking, so that one needs not work hard to understand the functions themselves.

2 Preliminaries

We first quote from Sections 1 and 2 of [8] to supply with some preliminary information.

In studies of algorithm in analysis, one puts the basis of considerations on *computable reals.* Here a real number x is said to be *computable* if there is a sequence of fractions $\{r_n\}$ which approximates x and satisfies the following two conditions.

(1) There is an algorithm which constructs the fractional sequence $\{r_n\}$. (That is, one can write a program which, for each natural number n, assigns a fractional number r_n.)

(2) There is an algorithm which measures the precision of approximation. (That is, one can write a program which, for each natural number p, assigns a natural number N_p such that for every natural number $m \geq N_p$ $|x - r_m| \leq 1/2^p$ holds.)

When the condition (2) holds, we say that x if *effectively* approximated by $\{r_n\}$. In general, we use the expression *effective* when a condition similar to (2) is satisfied.

One might say that a computable real number can be approximated by fractional numbers with guarantee of infinite precision.

A *computable sequence of reals* can also be defined in a similar manner. One needs computability of a sequence of real numbers when one has to refer to the *limit*.

The *computability* of a continuous real function on a compact interval can be defined in a natural manner. There are several alternatives, but they are all more or less the same. A real function f (on a compact interval) is *computable* if (1) f preserves the sequential computability, that is, for any input of a computable sequence of reals, its output by f is also a computable sequence; (2) f has a *recursive* modulus of uniform cotinuity.

The definition can easily be extended to fractional functions.

Computability on an open interval can be defined in terms of an approximation of the interval by a sequence of compact intervals and a modulus of uniform continuity which is recursive in the approximating intervals.

As for the computability of a *discontinuous* function, one has to start with speculation of what an *effective approximation* at a discontinuity means, and there are several alternatives for it.

Typical and simple discontinuous functions which may contain some computational information are step functions (with computable jump points and computable values).

Here we will take up a very simple step function, the Gaußian function or the integer part function, as an example, and propose some ways of dealing with its computability properties. Indeed, this is a most exemplary case with respect to which our problem can be distilled.

The Gaußian function can be defined as

$$[x] = n \quad \text{if} \quad n \leq x < n+1$$

and hence the value can be determined by judging $<$ alone unless x be an integer. When x *is* an integer, $x = n$? is usually undecidable (even for a computable x), and hence there is no general computation algorithm for $[x]$.

In what sort of viewpoint can one discuss the computability of a function which has such an attribute?

In Sections 2 and 3 of [9], this problem is discussed in detail. For explanation, we quote Section 3 of [8]. In Sections 2 and 3 of [13], a similar problem is discussed for the system of Rademacher functions.

3 Computing $[x]$

Let x be a computable real number and let us consider how to compute its value $[x]$, the Gaussian of x. For the sake of simplicity, we assume $x > 0$.

For $n = 0, 1, 2, 3, \cdots$, keep asking $x < n$?. (In fact, we compute with respect to a computable sequence of fractions $\{r_m\}$ which approximates x.) One will infallibly hit an n satisfying $n < x < n+2$.

If one is fortunate so that one hits an n satisfying $n < x < n+1$, then put $[x] = n$ and the computation halts.

Otherwise, one checks

$$r_{\alpha(p)} < (n+1) - \frac{`}{1} 2^p$$

for $p = 1, 2, \cdots$. (α is a recursive modulus of convergence of $\{r_n\}$ to x.) According to its answer, we define a sequence of integers $\{N_p\}$ as follows.

While the answer is *No*, put $N_p = n + 1$. Once the answer becomes *Yes* at p, then put $N_q = n$ for all q satisfying $q \geq p$.

The sequence $\{N_p\}$ is well-defined and recursive, and it can be classically shown that, if $N_p = n + 1$ holds for all p, then the limit of the sequence is $n + 1$; otherwise, the limit is n. In either case, the sequence approximates the value $[x]$ effectively.

Now, we have to be careful here to note that it is *not* decidable whether the limit is n or $n + 1$. It is true that one of the two cases definitely holds, the limit *is* the value $[x]$, and there *is* a recursive modulus of convergence. Only we do *not* know which case holds.

This undecidability indicates that, although there is a computation algorithm for each x, it does not guarantee a master program to compute $[x]$. Indeed, there is a computable sequence of real numbers whose values do not form a computable sequence of reals.

As an example of a sequence of step functions, I have taken up the Rademacher function system $\{\phi_n\}$, which is defined as follows.

$$\phi_n(x) = \begin{cases} 1, & x \in [\frac{2i}{2^n}, \frac{2i+1}{2^n}) \\ -1, & x \in [\frac{2i+1}{2^n}, \frac{2i+2}{2^n}) \end{cases}$$

where $i = 0, 1, 2, \cdots, 2^{n-1} - 1$.

As for the computation of this sytem, for any single, computable real number $x \in [0, 1)$, the sequence of function values at x forms a computable sequence of reals ([13]).

4 Functional approach

In order to regard a (step) function as a computable element in a functional space, one has to select a space appropriately. For the function $[x]$, we have chosen two Fréchet spaces. One is to view the function as a sequence of values (at integer points) and the other is to view it as a locally integrable function.

We quote from Sections 4 and 5 of [8].

Taking this into account, for a step function f whose jump points are integers, we identify f with the sequence $\{f(n)\}_{0,\pm1,\pm2,\ldots}$. Let $R^{\mathbf{Z}}$ denote the space of integer-indexed number sequences.

For an element in this space,

$$x = \{\cdots, \xi_{-k}, \cdots, \xi_{-1}, \xi_0, \xi_1, \cdots, \xi_k, \cdots, \},$$

define a sequence of *semi-norms* by

$$p_m(x) = \max\{|\xi_k| : |k| \leq m\} \tag{1}$$

Then the space becomes a Fréchet space.

On the other hand, let Σ denote the family of the right-continuous step functions whose jump points are integers. Then

$$q_m(f) = p_m(\{f(k)\}) \tag{2}$$

becomes a sequence of semi-norms. A function on Σ can be completely determined by the values at integer points.

It is obvious that $\langle \Sigma, \{q_m\}\rangle$ is isomorphic with $\langle \mathbf{R}^{\mathbf{Z}}, \{p_m\}\rangle$. In the latter space, a computable element is a (an integer-indexed) computable sequence of reals. So, it will be natural to define a computable element of Σ to be the one whose sequence of values is computable.

The Gaussian function is certainly a Σ-function, and the sequence of its values $\{n\}_n$ is computable. So, it is computable in the sense above.

One can draw a graph of the Gaussian function according to the idea above. That is, in the $x-y$ plane, mark the point (n, n) (or a vertical arrow from $(n, 0)$ to (n, n)) for each integer n, and then draw an arrow from that point (from the tip of the arrow) to the right in a manner that its tip does not reach the next point $(n+1, n+1)$.

We next proceed to the space of locally integrable functions.

a Fréchet space with the sequence of semi-norms

$$p_k(f) = \int_{[-k,k]} |f| dx$$

Let us denote this space with $\langle L^1_{loc}(\mathbf{R}), \{p_k\}\rangle$, or $\mathcal{L}$for short.

As a generating set of the space $\mathcal{L}$, take for example the family of step functions whose jump points and values are rationals and which have compact supports of integer end-points. One can also take the sequence of monomials $1, x, x^2, x^3, \cdots, x^n, \cdots$ as a generating set.

Similarly, a function in $\mathcal{L}$ can be defined to be computable if it is effectively approximated by a recursive enumeration of rational coefficient polynomials with respect to the semi-norms $\{p_k\}$. The sequence of monomials can therefore be regarded as an effectively generating set in $\mathcal{L}$.

The family of step functions as above can also be an effective generating set.

The Gaussian function can be effectively approximated by such step functions, and so it must be computable in this space.

In order to draw a graph of the Gaussian function according to this idea, one would draw an open segment between two integer points (and put a white circle at the integer point if desired). A white circle indicates that one needs not take into account the value there.

For the Rademacher function system, we employ the Banach space of p-integrable functions on $[0, 1]$ for any computable number p, $1 \leq p < \infty$.

5 Uniform topological space and computability

So far, the functions in question were defined on sets of real numbers, in which the usual distance metric is available. In [1], another metric for the interval $[0,1)$ has been used in order to discuss the computability properties of the Rademacher and Walsh function systems.

In working with computability problems of real-valued step functions, however, one finds that the metric of the domain is *not* indispensable. It is only the uniformity of the topology as well as the computability structure of the codomain that are essential. For this reason, it is worth the effort to find out how we can work with computability properties of some real-valued functions from a uniform topological space (with countable index set). Although a uniform topological space with a countable index set can be converted to a metric space (and certainly vice versa), and also that two kinds of convergence are equivalent, it is important how things look like in the uniform topology directly so that the circumstances under which a function becomes computable will become clear.

A computability structure on a uniform topological space and its applications will be explained below very, very briefly.

An *effective* uniform topological space is a uniform topological space in which the axioms of open basis are *effectivized.* The computability structure on such a space is defined by three axioms.

We will quote from [11] for the definitions of an effective uniform topological space and a computability structure on it.

A uniform topology $\{V_n\}$ on X is called an *effective uniform topology* if there are recursive functions $\alpha_1, \alpha_2, \alpha_3$ which satisfy the following.

For every $n, m \in \mathbf{N}$ and every $x \in X$, $V_{\alpha_1(n,m)}(x) \subset V_n(x) \cap V_m(x)$ (effective A_3).

For every $n \in \mathbf{N}$ and every $x, y \in X$, $x \in V_{\alpha_2(n)}(y)$ implies $y \in V_n(x)$ (effective A_4).

For every $n \in \mathbf{N}$ and every $x, y, z \in X$, $x \in V_{\alpha_3(n)}(y), y \in V_{\alpha_3}(z)$ implies $x \in V_n(z)$ (effective A_5).

A double sequence $\{x_{l,k}\}$ from X is said to *effectively converge* to a sequence $\{x_l\}$ if there is a recursive function β satisfying

$$\forall n \forall l \forall k \geq \beta(l,n)(x_{l,k} \in V_n(x_l))$$

Let $\mathcal{S}$ be a family of sequences from X. (As usual, we include multiple sequences, such as double sequences, triple sequences, when we talk about sequences.)

$\mathcal{S}$ is called a *computability structure* if it satisfies the following.

C1: (Non-emptiness) $\mathcal{S}$ is nonempty.

C2: (Re-enumeration) If $\{x_k\} \in \mathcal{S}$ and α is a recursive function, then $\{x_{\alpha(i)}\}_i \in \mathcal{S}$.

C3: (Limit) If $\{x_{l,k}\}$ belongs to $\mathcal{S}$, $\{x_l\}$ is a sequence from X, and if $\{x_{l,k}\}$ converges to $\{x_l\}$ effectively, then $\{x_l\} \in \mathcal{S}$. That is, $\mathcal{S}$ is closed with respect to effective convergence.

A sequence belonging to $\mathcal{S}$ is called a *computable sequence*. An element of X is called *computable* if the sequence $\{x, x, \cdots\}$ is computable.

The *tree topology* of a binary tree is a uniform topology of clopen sets, and the space is compact. The family of all the recursive paths in the binary tree forms a computability structure. Furthermore, an effective enumeration of all the eventually zero paths is an effective separating set.

As an application, we can show that the system of Rademacher functions defined on the binary tree forms a *uniformly computable sequence of functions.*

The next space of our concern, denoted by $\mathcal{A}$ is an *amalgamation* of the discrete space of integers and the union of all the open intervals with integer end points with relativized open interval topology. As a set, $\mathcal{A}$ is identical with the set of real numbers, but it becomes a uniform topological space.

We can also show that the amalgamated space is not complete. In regards to the computability structure, however, we know that it is *effectively complete* and, in particular, $\mathcal{A}$ is *effectively equi-totally bounded.*

The function $[x]$ is continuous in $\mathcal{A}$.

The set of *computable sequences* of $\mathcal{A}$ is defined as follows. Let $\{x_n\}$ be a seuqnece of the set $\mathbf{A_R}$, where $x_n \in J_n$ (J_n is either $\mathbf{J^Z}$ or $\mathbf{J}^k$ for some k). $\{x_n\}$ is called *computable* if there is a double sequence of rationals $\{r_{nl}\}$ such that $\{r_{nl}\}$ is a computable sequence in the usual sense, for all l $r_{nl} \in \mathbf{J_n}$, and $\{r_{nl}\}$ converges effectively to $\{x_n\}$ in the usual topology of $\mathbf{R}$.

The computable sequences in the sense above form a computability structure for $\mathcal{A}$, and the uniform computability of $[x]$ with respect to this computability structure follows.

References

[1] T.Mori,*On the computability of Walsh functions*, manuscript.

[2] T.Mori,Y.Tsujii and M.Yasugi, *Computability structure on metric spaces*, Combinatorics, Complexity and Logic (Proceedings of DMTCS'96), ed. by Bridges et.al,Springer(1996),351-362.

[3] M.B.Pour-El and J.I.Richards, *Computability in Analysis and Physics*, Perspectives in Mathematical Logic, Springer-Verlag, 1989.

[4] M.Washihara, *Computability and Fréchet space*, Mathematica Japonica, 42(1995),1-13.

[5] M.Washihara, *Computability and tempered distributions*, ibid.,50,1(1999),1-7.

[6] M.Washihara and M.Yasugi, *Computability and metrics in a Fréchet space*, ibid., 43(1996), 1-13.

[7] M.Yasugi, *Computability of discontinuous functions: a report at Dagstuhl seminar*, http://www.kyoto-su.ac.jp/~yasugi/Recent.

[8] M.Yasugi, *Computability of discontinuous functions*, Manuscript (presented at Josai University, January 2000).

[9] M. Yasugi, V. Brattka and M. Washihara, *Computability properties of the Gaußian function*, Manuscript (1999): http://www.kyoto-su.ac.jp/~yasugi/Recent.

[10] M.Yasugi, T.Mori and Y.Tsujii, *Effective Properties of Sets and Functions in Metric Spaces with Computability Structure*, Theoretical Computer Science, 219(1999), 467-486.

[11] M.Yasugi, Y.Tsujii and T.Mori, *Uniform topological space and computability of some discontinuous functions*, Manuscript(2000): http://www.kyoto-su.ac.jp/~yasugi/Recent.

[12] M.Yasugi and M.Washihara, *Computability structures in analysis*, (in Japanese),Sugaku(MSJ),vol.50,no.2(1998),130-148.

[13] M.Yasugi and M.Washihara, *Rademacher functions and computability*, Manuscript. (Abstract for Ito Conference 00: http://www.kyoto-su.ac.jp/~yasugi/Recent.)