<table>
<thead>
<tr>
<th>項目</th>
<th>内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Computability problems of piecewise continuous functions (Feasibility of Theoretical Arguments of Mathematical Analysis on Computer)</td>
</tr>
<tr>
<td>Author(s)</td>
<td>Yasugi, Mariko; Tsujii, Yoshiki</td>
</tr>
<tr>
<td>Citation</td>
<td>数理解析研究所講究録 (2004), 1381: 118-127</td>
</tr>
<tr>
<td>Issue Date</td>
<td>2004-06</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/2433/25682">http://hdl.handle.net/2433/25682</a></td>
</tr>
<tr>
<td>Type</td>
<td>Departmental Bulletin Paper</td>
</tr>
<tr>
<td>Textversion</td>
<td>publisher</td>
</tr>
</tbody>
</table>

京都大学
Computability problems of piecewise continuous functions

Mariko Yasugi and Yoshiki Tsujii
Faculty of Science, Kyoto Sangyo University
yasugi@cc.kyoto-su.ac.jp, tsujiiy@cc.kyoto-su.ac.jp

Main Subject: Interrelation between computability in the limit and computability in the uniform space

1 Introduction

In studies of algorithm in analysis, one puts the basis of considerations on the computability of real numbers and the computability of continuous functions.

The notions of computability respectively of a real number (a sequence of real numbers) and of a continuous function (a sequence of continuous functions) are generally agreed to be natural and in a sense the strongest.

For a continuous function, computability means that there is a way to nicely approximate the values for computable inputs, and this notion depends on the continuity.

Very often, however, we compute values and draw a graph of a discontinuous function. We thus hope that some class of discontinuous functions can be attributed a certain kind of computability. In an attempt of computing a discontinuous function, a problem arises in the computation of the value at a jump point (a point of discontinuity). The problem is caused by the fact that it is not generally decidable if a real number is a jump point, that is, the question "x = a?" is not decidable even for computable x and a.

One method of dissolving this problem was proposed in [10] by Pour-El and Richards. In their theory, a function is regarded as computable as a point in a function space. This is sufficient in order to draw a rough graph of the function, but does not supply us with information how to compute individual values.

There are many ways of characterizing computation of a discontinuous function. Here we will first report two of the approaches to this problem by Brattka, Mori, Tsujii, Washihara and Yasugi, and then claim that they are equivalent for piecewise continuous functions which jump at a computable sequence of real numbers, worked out by Tsujii and Yasugi [18]. One is to express the value

\[ l \]

This work has been supported in part by JSPS Science Grant 12440031 and by Institute for Comprehensive Research, Kyoto Sangyo University (Yasugi Project 03).
of a function at a jump point in terms of limiting recursive functions instead of recursive functions ([16]). Another is to change the topology of the domain of a function ([12]).

References of related works and some other approaches are listed in References, details of which will not be mentioned here. Pour-El theory as well as its succeeding works on computability structures for Frechet spaces and metric spaces are also explained in [21].

2 Preliminaries

We first give some basic definitions. Details are seen in, for example, [10] and [21].

A sequence of rationals \( \{r_n\} \) is recursive if it is represented as

\[
r_n = (-1)^\beta(n) \frac{\gamma(n)}{\delta(n)}
\]

with \( \beta, \gamma, \delta \) recursive.

A real number \( x \) is computable (\( R \)-computable) if

\[
\forall m \geq \alpha(p).|x - r_m| < \frac{1}{2^p}
\]

for recursive \( \alpha \) and \( \{r_m\} \). We will write this relation as

\[
x \simeq (r_m, \alpha).
\]

These definitions can be extended to sequences of rationals and reals.

A real (continuous) function \( f \) is computable if (i) and (ii) below hold.

(i) \( f \) preserves sequential computability: If \( \{x_n\} \) is computable, then \( \{f(x_n)\} \) is computable.

(ii) \( f \) is continuous with a recursive modulus of continuity, say \( \beta \):

\[
\forall p \forall n \in \mathbb{N}^+ \forall k \geq \beta(n, p) \forall x, y \in [n, n+1],
\]

\[
|x - y| < \frac{1}{2^k} \Rightarrow |f(x) - f(y)| < \frac{1}{2^p}.
\]

The definition can be extended to a computable sequence of functions.

The notions of computability of real numbers and continuous functions as above are generally agreed.

3 Points of discontinuity and various approaches

We would like to give some expression to a notion of computation of some discontinuous functions.
The problem that arises when computing the value of a discontinuous function is identifying a point of discontinuity, that is, \( x = a \) is not decidable even for computable \( x \) and \( a \).

There are various approaches to this problem. Our approach is a mathematical one, that is, using the mathematical language except for some elementary properties of recursive functions. Since we do not resort to any particular theory of computation, our approaches amounts to an axiomatic one.

What attracted us first was the functional approach proposed by Pour-El and Richards [10]. In their theory, a function is regarded as computable as a point in a space. Pour-El and Richards considered the computability structure on the Banach space and the Hilbert space. Succeeding them, Washihara investigated the computability structure on Fréchet spaces, such as the spaces of the \( \delta \)-function, the function space of bounded variation; Yoshikawa has studied the interpolation space, the Hilbert space [23] and others; Zhong has worked on the Solev space, to list a few.

In these theories, pointwise computation is not in sight. However, very often we need and indeed do pointwise computation of a function which is discontinuous here and there. How can we describe such deeds with mathematical terms? We have attempted two methods, one by the limiting recursive functions and one by changing the topology of the domain into a certain uniform topological space.

Below is a simplified map of the circumstances.

\[
\begin{array}{c}
\text{Logical method} \quad \text{Change of topology of domain} \\
\text{LEM} \quad \text{limiting recursive} \quad \text{uniform space}
\end{array}
\]

Logical method
Yasugi, Brattka, Washihara  Computation in the limit [16]
Change of topology of domain
Tsujii, Mori, Yasugi  Computability on the general metric space [9][17]
Tsujii, Mori, Yasugi  Computability on the uniform space [12][20]
Mori  Fine metric [7]
Equivalence
Yasugi, Tsujii  Limiting recursive method and uniform space method [18][19]
4 Topological computability

Definition 4.1 (Effective uniformity) Let $X$ be a non-empty set. \{V_n\} will denote an effective uniformity on $X$, that is, it is a map

$$V_n : X \rightarrow P(X)$$

and satisfies the following five axioms. ($T = (X, \{V_n\})$ is then called an effective uniform topological space.)

$$A_1 & A_2 : \cap_n V_n(x) = \{x\}.$$

There are recursive functions $\alpha_1, \alpha_2, \alpha_3$ such that

$$\forall n, m \in \mathbb{N} \forall x \in X. V_{\alpha_1(n,m)}(x) \subset V_n(x) \quad \text{(effective } A_3) \quad \forall n \in \mathbb{N} \forall x \in V_{\alpha_2(n)}(y) \rightarrow y \in V_n(x) \quad \text{(effective } A_4).$$

$$\forall n \in \mathbb{N} \forall x, y, z \in X. x \in V_{\alpha_3(n)}(y), y \in V_{\alpha_3(n)}(z) \rightarrow x \in V_n(z) \quad \text{(effective } A_5).$$

Definition 4.2 (Effective convergence) \{x_k\} $\subset X$ effectively converges to $x$ in $X$ if there is a recursive function $\gamma$ satisfying $\forall x \forall k \geq \gamma(n)(x_k \in V_n(x)).$

The definition can be extended to effective convergence of a multiple sequence.

Definition 4.3 (Computability structure) (1) Let $S$ be a family of sequences from $X$ (Multiple sequences included).

$S$ is called a computability structure if:

C1: (Non-emptiness) $S$ is nonempty.

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

This condition can be extended to multiple sequences.

C3: (Limit) If \{x_{lk}\} belongs to $S$, \{x_i\} is a sequence from $X$, and \{x_{lk}\} converges to \{x_i\} effectively, then $\{x_i\} \in S$. ($S$ is closed with respect to effective convergence.)

This condition can be extended to multiple sequences.

(2) A sequence belonging to $S$ is called computable.

(3) $x$ is computable if \{x, x, \cdots\} is computable.

(4) We write

$$C_T = (X, \{V_n\}, \alpha_1, \alpha_2, \alpha_3, S).$$

Definition 4.4 (Effective approximation) \{e_k\} $\in S$ is an effective approximating set of $S$: $\forall \{x_i\}$ computable, there is a recursive function $\nu$ such that

$$\forall n \forall l (e_{\nu(n,l)} \in V_n(x_l)).$$

Definition 4.5 (Effective separability) Suppose \{e_k\} is an effective approximating set and dense in $X$:

$$\forall n \forall x \exists k (e_k \in V_n(x))$$

Then $C_T$ is effectively separable, and \{e_k\} is called an effective separating set.
Definition 4.6 (Relative effective completeness) (1) \( \{x_j\} \in X \) is effectively Cauchy if
\[
\forall n \forall j \geq \alpha(n) (x_j \in V_n(x_{\alpha(n)}))
\]
for a recursive \( \alpha \).

This can be extended to multiple sequences.

(2) \( E_T = \langle X, \{V_n\}, \alpha_1, \alpha_2, \alpha_3, S \rangle \) is relatively effectively complete (with respect to \( S \)) if every computable and effectively Cauchy multiple sequence \( \{x_m\} \) effectively converges to a sequence \( \{x_m\} \in X \).

Definition 4.7 (Relative computability) (1) \( f : X \rightarrow \mathbb{R} \) is relatively computable (with respect to \( S \)) if:

(i) \( f \) preserves sequential computability:
\( \{x_m\} \) computable \( \rightarrow \) \{\( f(x_m) \)\} computable

(ii) For any \( \{x_m\} \in S \) there exists a recursive function \( \gamma(m, p) \) such that \( y \in V_{\gamma(m,p)}(x_m) \) implies \( |f(y) - f(x_m)| \leq \frac{1}{2^p} \).

(2) can be extended to a sequence of functions.

Fact: \( f \) is relatively computable in the sense of (1) if and only if the sequence \( \{f, f, f, \ldots\} \) is relatively computable in the sense of (2).

Definition 4.8 (Computable function) (1) \( f : X \rightarrow \mathbb{R} \) is computable if:

(i) \( f \) preserves sequential computability.

(ii) \( f \) is relatively computable, and there exist an effective approximating set, say \( \{e_k\} \in S \), and a recursive function \( \gamma_0(k, p) \) for which
\[
y \in V_{\gamma_0(k,p)}(e_k) \text{ implies } |f(y) - f(e_k)| \leq \frac{1}{2^p}
\]
and
\[
\bigcup_{k=1}^{\infty} V_{\gamma_0(k,p)}(e_k) = X
\]
for \( p \).

(2) Can be extended to a sequence of functions.

Definition 4.9 (Uniform computability) \( f \) is uniformly computable if \( f \) preserves sequential computability and there is a recursive modulus of uniform continuity for \( f \).

5 Computation in the limit

A limiting recursive function was first proposed by Gold [4]. It is defined to be the limit of a recursive function (if the limit exists).

Definition 5.1 (Limiting recursion: Gold) Let \( r, s \geq 0 \) and \( g, \phi_1, \ldots, \phi_r \) be recursive. Then the partial function \( h \) as defined below is called limiting recursive.
\[
h(p_1, \ldots, p_s) = \lim_{n} g(\bar{\phi}_1(n), \ldots, \bar{\phi}_r(n), p_1, \ldots, p_s, n)
\]
where \( \phi(n) \) is a code for 
\[
\langle \phi(0, p_1, \cdots, p_s), \cdots, \phi(n, p_1, \cdots, p_s) \rangle,
\]

**Example**
\[
h(p_1, \cdots, p_s) = \lim_{n} \phi(n, p_1, \cdots, p_s).
\]

**Definition 5.2** (Computation by limit) A function \( f : \mathbb{R} \to \mathbb{R} \) is said to be computed by limit if
\[
x \simeq \langle r_p, \alpha \rangle \mapsto f(x) \simeq \langle s_p, \beta \rangle,
\]
where \( \{r_p\} \) and \( \{s_p\} \) are recursive, while \( \beta \) is limiting recursive.

### 6 Equivalence

We will set the following assumption.

[Assumption] \( \{a_k\}_{k \in \mathbb{Z}} \), an \( \mathbb{R} \)-computable sequence, is called a basic sequence if it holds
\[
a_k < a_{k+1}, \cup_k [a_k, a_{k+1}] = \mathbb{R}.
\]

Assume \( a_k \simeq \langle v_{kp}, \gamma \rangle \), with \( \{v_{kp}\} \) and \( \gamma \) recursive.

We will consider the computation of real functions relative to the basic sequence \( \{a_k\} \).

**Definition 6.1** (\( A \)-space) \( A_k = \{a_k\}, J_k = (a_k, a_{k+1}), J = \cup J_k \),
\[
A = \{a_k : k \in \mathbb{Z}\} = \cup_k A_k, A_\mathbb{R} = A \cup J
\]
(As a set, \( A_\mathbb{R} = \mathbb{R} \))
\[
n = 1, 2, 3, \cdots, x \in A_\mathbb{R}
\]
\[
U_n(x) := \{x\} = \{a_k\} \text{ if } x \in A_k;
\]
\[
U_n(x) := \{y : y \in J_k, |x - y| < \frac{1}{2^n}\} \text{ if } x \in J_k
\]
\[
A = \langle A_\mathbb{R}, \{U_n\} \rangle
\]

**Corollary 1** \( \{U_n\} \) is an effective uniformity on \( A_\mathbb{R} \)

**Definition 6.2** (\( A \)-computability)
\( A_Q = \mathbb{J} \cap \mathbb{Q} \)
\[
k \in \mathbb{Z} \mapsto \iota_k \text{ (\( \iota \)-symbol): a "symbolic name" for } a_k
\]
\[
A_\mathbb{Q}^* = A_\mathbb{Q} \cup \cup_k \{\iota_k\}
\]
\( \{q_{\mu n}\} \subset A_\mathbb{Q}^* \) is an \( A \)-sequence:
for each \( \mu \),
\[
\exists k \in \mathbb{Z}, q_{\mu n} = \iota_k \text{ for all } \mu \text{ or } \{q_{\mu n}\} \subset J_k
\]
An \( A \)-recursive sequence: a recursive \( A \)-sequence
\[
\{x_m\} \subset A_\mathbb{R}
\]
\[
\{x_m\} \simeq_A \langle q_{mn}, \alpha_A(m, p) \rangle \quad (*)
\]
\[
\{q_{mn}\} \subset J_k \text{ if } x_m \in J_k, \\
\{q_{mn}\} = \{\iota \} \text{ if } x_m \in A_k
\]

\forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \forall \for...
Definition 6.4 (Sequential computability) (1) $f$ is $\mathcal{L}$-sequentially computable: for any $\mathcal{R}$-computable $\{x_m\}$ ($x_m \simeq (r_m, \alpha)$), we can construct a recursive sequence of rational numbers $\{s_{mp}\}$ and a function $\delta$ which is "recursive in $\{l_m\}" such that $f(x_m) \simeq (s_{mp}, \delta)$.

(2) $f$ is $\mathcal{A}$-sequentially computable: for any $\mathcal{A}$-computable sequence of real numbers $\{x_m\}$, we can construct a recursive sequence of rational numbers $\{s_{mp}\}$ and a recursive function $\beta$ such that $f(x_m) \simeq (s_{mp}, \beta)$.

Theorem 2 (Equivalence of two notions of sequential computability) (1) (From $\mathcal{L}$ to $\mathcal{A}$) An $\mathcal{L}$-sequentially computable function $f$ is $\mathcal{A}$-sequentially computable.

(2) (From $\mathcal{A}$ to $\mathcal{L}$) An $\mathcal{A}$-sequentially computable function $f$ is $\mathcal{L}$-sequentially computable.

Definition 6.5 (Piecewise computable function) $f: \mathcal{R} \rightarrow \mathcal{R}$ is called piecewise computable if the following hold.

(i) For each $(\mathcal{R})$-computable real number $x$, $f(x)$ is $\mathcal{R}$-computable.

(ii) There is a recursive function $\kappa$ with which, for any $x, y$ such that $a_k < x, y < a_{k+1}$ and $|x - y| < \frac{1}{2^\kappa(x, \rho)}$, $|f(x) - f(y)| \leq \frac{1}{2^n}$.

Conclusion A piecewise continuous function whose jump points form a computable sequence of real numbers can be regarded as computable if it is piecewise computable and is $\mathcal{L}$- (hence $\mathcal{A}$-) sequentially computable.

References


