滑らかな常微分方程式の計算量

太田浩行* 河村彰星†

マルチン・ツィーグラー‡ カルステン・レースニク§

概要

The computational complexity of the solution h to the ordinary differential equation $h(0) = 0$, $h'(t) = g(t, h(t))$ under various assumptions on the function g has been investigated in hope of understanding the intrinsic hardness of solving the equation numerically. Kawamura showed in 2010 that the solution h can be PSPACE-hard even if g is assumed to be Lipschitz continuous. We place further requirements on the smoothness of g and obtain the following results: the solution h is still PSPACE-hard if g is assumed to be continuously differentiable; for each $k \geq 2$, the solution h is hard for the counting hierarchy if g is assumed to be k-times continuously differentiable.

1 序論

連続実関数 $g\colon [0,1] \times \mathbf{R} \to \mathbf{R}$ に対して次の常微分方程式を考える.

$$h(0) = 0, \qquad Dh(t) = g(t, h(t)) \quad (t \in [0,1]) \tag{1.1}$$

ただし Dh は h の導関数. 本稿では g が多項式時間計算可能であるとき, 解 h がどれほど複雑でありうるかを考える. ここで多項式時間を初めとする計算量は**計算可能解析学** (Computable Analysis) [14] における概念であり, 実関数を指定された精度で近似する計算の困難さを測るものである (本稿では 2 節で説明する).

g に多項式時間計算可能であることの他に何の制限も設けない場合, 解 h (一般に一意でない) は計算不能でありうるため, 様々な制限の下で h の計算量が研究されている (表 1.1). この表では下に向うにつれて左列の条件が強まっており, g が (大域的) Lipschitz 条件を満たせば解 h が一意であるが, 表の第 3 行にある通りこのとき唯一解 h はちょうど PSPACE 困難でありうることがわかっている [3]. 本稿ではより強く g に滑らかさの仮定を置いたときの h の複雑さを調べる.

一般に数値計算においてはしばしば, 或る種の算法を適用できるようにするため, 或いは解析しやすくするために, 与えられる関数に何らかの滑らかさ (十分な回数微分可能であるなど) を仮定

* 東京大学, hota@is.s.u-tokyo.ac.jp

† 東京大学

‡ Martin Ziegler, ダルムシュタット工科大学

§ Carsten Rösnick, ダルムシュタット工科大学

表 1.1　g が多項式時間計算可能であるときの常微分方程式 (1.1) の解 h の計算量

制限	上界	下界
—	—	計算不可能になりうる [11]
h が g の唯一解	計算可能 [1]	任意の時間がかかりうる [6, 9]
g が Lipschitz 条件を満たす	多項式領域計算可能 [6]	PSPACE 困難になりうる [3]
g が $(\infty, 1)$ 回連続微分可能	多項式領域計算可能	PSPACE 困難になりうる (定理 1.1)
g が (∞, k) 回連続微分可能 (k は任意の定数)	多項式領域計算可能	CH 困難になりうる (定理 1.2)
g が解析的	多項式時間計算可能 [10, 8, 2]	—

すると都合のよいことがある．しかしこれは経験則にすぎず, 実際に滑らかさの仮定が解の複雑さを計算量の意味で抑える効果をもつのかについてはあまり論ぜられて来なかった．

本稿で扱う微分方程式についていえば, 極端なのは g が解析的である場合であり, このときにはテイラー級数による解法により, 表の最下列にあるように h は g と同じく多項式時間計算可能になる．本稿では Lipschitz 条件より強いが解析的よりは弱い滑らかさの仮定を考える (表の第 4, 5 行)．ここで (i, j) 回連続微分可能とは, 第一, 第二変数についてそれぞれ i 回, j 回微分でき, その導関数が連続であることである[*1]．

Theorem 1.1. 多項式時間計算可能かつ $(\infty, 1)$ 回連続微分可能な実関数 $g\colon [0,1] \times [-1,1] \to \mathbf{R}$ であって, 方程式 (1.1) が PSPACE 困難な解 $h\colon [0,1] \to \mathbf{R}$ を持つものが存在する．

Theorem 1.2. 任意の自然数 k に対して, 多項式時間計算可能かつ (∞, k) 回連続微分可能な実関数 $g\colon [0,1] \times [-1,1] \to \mathbf{R}$ であって, 方程式 (1.1) が CH 困難な解 $h\colon [0,1] \to \mathbf{R}$ を持つものが存在する．但し CH $\subseteq$ PSPACE は計数階層 (3.2 節) である．

ここで $g\colon [0,1] \times \mathbf{R} \to \mathbf{R}$ でなく $g\colon [0,1] \times [-1,1] \to \mathbf{R}$ と書いたのは, 本稿では実関数の多項式時間計算可能性を, 定義域が有界閉領域のときにのみ定義するからである．このため h が区間 $[-1,1]$ の外に値を取ることがあると方程式 (1.1) が意味をなさなくなるが, 定理 1.1 において h が解であるというのは, 任意の $t \in [0,1]$ について $h(t) \in [-1,1]$ が満たされることも含めて述べている．なお両定理とも Lipschitz 条件よりも強い仮定を置いているため, そのような h は g に対して, 存在すれば唯一である．

[*1] 二変数関数 g が $i + j \le k$ を満たす任意の自然数 i, j について (i, j) 回連続微分可能であることを k 回連続微分可能と言うこともあるが, 本稿では各変数について微分できる回数をこのように分けて書く．

本稿のように対象を滑らかな関数に制限することによる計算量の変化について, 常微分方程式以外の問題では次のような否定的な結果がある. 多項式時間計算可能な関数から積分により得られる関数は, もとの関数を無限回微分可能なものに限ってもなお一般の場合と同じく #P 困難である. [7, 定理 5.33]. 最大化でも同様に, 無限回微分可能な関数に限っても一般の場合と同じく NP 困難である [7, 定理 3.7][*2] (なお対象を解析的な関数に限ると, やはり級数を用いた議論により, これらは多項式時間計算可能になる). 一方常微分方程式については, 定理 1.2 は各 k についてそれぞれ成立つが, g が (∞, ∞) 回微分可能であると仮定したときの h の計算量については依然不明である.

記法

自然数の集合を $\mathbf{N}$, 実数の集合を $\mathbf{R}$, 有理数の集合を $\mathbf{Q}$ と表す.

A, B を $\mathbf{R}$ の有界閉区間とする. 実関数 $f\colon A \to \mathbf{R}$ に対して $|f| = \sup_{x \in A} f(x)$ と書く.

連続な一変数関数 $f\colon A \to \mathbf{R}$ が i 回連続微分可能であるとは, 導関数 $Df, D^2f, \ldots, D^if$ が存在し, すべて連続であることと定義する. 連続な二変数関数 $g\colon A \times B \to \mathbf{R}$ が (i,j) **回連続微分可能**であるとは, 導関数 $D_1^n D_2^m g$ が, 任意の $0 \le n \le i$, $0 \le m \le j$ を満たす n, m において存在し連続であることと定義する. g が (i,j) 回連続微分可能であるとき, その導関数 $D_1^i D_2^j g$ を $D^{(i,j)}g$ と表記する. また任意の i について g が (i,j) 回連続微分可能であるとき, g は (∞, j) 回連続微分可能であると定義する.

2 Computational Complexity of Real Functions

2.1 Computation of Real Function

Real numbers are represented by functions from string to string.

Definition 2.1. A function $\phi\colon \{0\}^* \to \{0,1\}^*$ is a *name* of a real number x if for all $n \in \mathbf{N}$, $\phi(0^n)$ is the binary representation of $\lfloor x \cdot 2^n \rfloor$ or $\lceil x \cdot 2^n \rceil$, where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are rounding down and up functions.

In effect, a name of a real number x receive n and return approximation of x with precision 2^{-n}.

We use *oracle Turing machines*(henceforth just machines) to work on names of real numbers(Figure 2.1).

Let M be a machine and ϕ be a function from string to string, We write $M^\phi(0^n)$ as the output string by computation of M given ϕ as oracle and string 0^n as input. So we also regard

[*2] ただし葛 [7, 定理 3.7] の証明において関数 f を

$$f(x) = \begin{cases} u_s & \text{if not } R(s,t) \\ u_s + 2^{-(p(n)+2n+1)\cdot n} \cdot h_1(2^{p(n)+2n+1}(x - y_{s,t})) & \text{if } R(s,t) \end{cases}$$

に修正する必要がある.

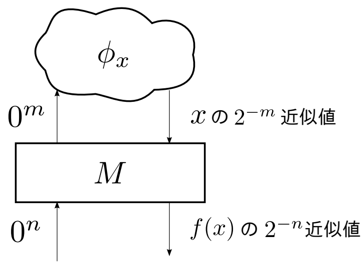

図 2.1　A machine M computing a real function f

M^ϕ as a function from string to string.

Definition 2.2. Let A be a bounded closed interval of $\mathbf{R}$. A machine M *computes* a real function $f\colon A \to \mathbf{R}$ if for any $x \in A$ and any name ϕ_x of it, M^{ϕ_x} is a name of $f(x)$.

When A is a bounded closed interval of $\mathbf{R}^2$, we define a machine computing $f\colon A \to \mathbf{R}$ in a similar way using machines with two oracles. A real function is (*polynomial-time*) *computable* if there exists some machine that computes it (in polynomial time).

When a machine M computes a real function f, for each demanded precision 2^{-n}, precision 2^{-m}. Computable real functions are continuous. Giving all approximations of functions at rational points an relation between n and m, we can characterize (polynomial-time) computability of real function without oracle Turing machines

Lemma 2.3. A real function is (polynomial-time) computable if and only if there exist a (polynomial-time) computable function $\phi\colon (\mathbf{Q} \cap [0,1]) \times \{0\}^* \to \mathbf{Q}$ and polynomial $p\colon \mathbf{N} \to \mathbf{N}$ such that for all $d \in \mathbf{Q} \cap [0,1]$ and $n \in \mathbf{N}$,

$$|\phi(d, 0^n) - f(d)| \leq 2^{-n}, \tag{2.1}$$

and for all $x, y \in [0,1]$, $m \in \mathbf{N}$,

$$|x - y| \leq 2^{-p(m)} \Rightarrow |f(x) - f(y)| \leq 2^{-m}. \tag{2.2}$$

where each rational number in $\mathbf{Q}$ is represented by a pair of integers in binary representation.

2.2 Reduction and Hardness

A language $L \subseteq \{0,1\}^*$ is identified with the function $L\colon \{0,1\}^* \to \{0,1\}$ such that $L(u) = 1$ if and only if $u \in L$.

Definition 2.4 (Reduction)**.** A Language L reduces to a real function $f\colon [0,1] \to \mathbf{R}$ if there exists a polynomial-time function S and a polynomial-time oracle Turing machine M (Figure 2.2) such that for any string u:

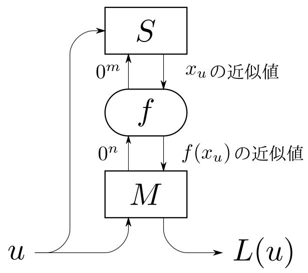

図 2.2　Reduction from a language L to a function $f\colon [0,1] \to \mathbf{R}$

(i) $S(u, \cdot)$ is a name of x_u;

(ii) $M^\phi(u)$ accepts if and only if $u \in L$ for any name ϕ of $f(x_u)$.

As a matter of form this definition is different from that by Kawamura but they have same power as reduction. Let C be a complexity class, a function f is *C-hard* if for all language in C is reducible to f.

3 Proof of The Theorems

We give a proof sketch of Theorem 1.1 and Theorem 1.2. First we define a discrete version of differential equations in Section 3.1, and show PSPACE-hardness or CH-hardness of the problem with restrictions similar to once differentiable or k-times differentiable in Section 3.2. In Section 3.3, we show that these discrete problems is simulatable by certain families of smooth differential equations. We construct the smooth differential equations stated in theorems putting one family into one real function in Section 3.4. Henceforth we call the discrete version of differential equations as difference equations.

The idea to simulate difference equations with differential equations is essentially from the proof of Lipschitz version [3]. In this paper we focus on the structure of difference equations to analyze precisely the effect of the smoothness restriction. Consequently we show with the restriction differentiable more than once differential equations can simulate difference equations whose height is enough small, and it gives the CH-hardness of smooth differential equations.

3.1 Difference Equations

This section we define difference equations, which is a discrete version of differential, and show that PSPACE-hardness or CH-hardness depending on restrictions on height.

Let $[n]$ denote $\{0, \ldots, n-1\}$. Let $G\colon [P] \times [Q] \times [R] \to \{-1, 0, 1\}$ and $H\colon [P+1] \times [Q+1] \to [R]$, H is the solution of the *difference equation* given by G if for all $i \in [P]$ and

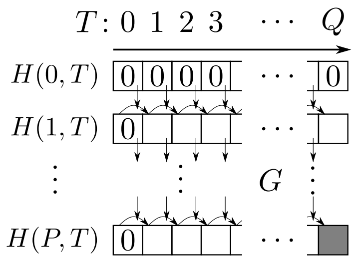

図 3.1　差分方程式 G とその解 H

$T \in [Q]$(Figure 3.1),

$$H(i,0) = H(0,T) = 0 \tag{3.1}$$
$$H(i+1,T+1) - H(i+1,T) = G(i,T,H(i,T)). \tag{3.2}$$

We call P, Q and R as *height*, *width*, *cell size* of the difference equation. The equation (3.1) is similar to the initial condition $h(0) = 0$, and (3.2) is similar to $Dh(t) = g(t,h(t))$ in (1.1). In Section 3.3, we simulate difference equations by differential equations using this similarity.

We regard a family of difference equations $(G_u)_u$ computing L, by regarding the value of the bottom right cell as $L(u)$ for each u. A family $(G_u)_u$ of functions $G_u\colon [P_u]\times[Q_u]\times[R_u] \to \{-1,0,1\}$ *recognizes* a language L if let $(H_u)_u$ be as the solution of the difference equation given by G_u $H_u(P_u,Q_u) = L(u)$. A family $(G_u)_u$ is *uniform* if functions from u to its height and width and cell size G_u の段数, 列数, 欄の大きさそれぞれを u を入力とする関数と見做したときそれらが多項式時間計算可能であり, かつ与えられた (u,i,T,Y) から多項式時間で $G_u(i,T,Y)$ が計算できることと定義する. そのようなとき G_u の段数, 列数及び欄の大きさは, $|u|$ の多項式の指数 $(2^{\mathrm{poly}(|u|)})$ で抑えられる. G_u の段数を P_u と置くとき, 多項式 p が存在して $P_u \le p(|u|)$ が成立つとき, 族 $(G_u)_u$ は**多項式段**であるという. また定数 c,d が存在して $P_u \le c\log(|u|)+d$ が成立つとき, 族 $(G_u)_u$ は**対数段**であるという. この用語を使うと, 河村が Lipschitz 連続な場合の解析に用いた補題 [3, 補題 4.7] は次のように書ける.

Lemma 3.1. 多項式段の一様な差分方程式族によって認識される PSPACE 困難な言語が存在する[*3].

河村 [3] はこの多項式段の一様な差分方程式族を Lipschitz 連続な微分方程式で模倣することで, 表 1.1 第三行の結果を得た. 本稿の定理 1.1 はこの構成に手を加えて, $(\infty,1)$ 回微分可能な模倣に作り替えることにより (3.3, 3.4 節), やはり補題 3.1 から得られる.

本稿では更に, 対象とする差分方程式族を対数段に制限すれば, $(\infty,2)$ 回以上微分可能な関数によっても模倣できることを示し (3.3, 3.4 節), これと次の補題から定理 1.2 を得る.

[*3] 差分方程式によって認識される言語のクラスはカープ帰着において閉じており, 多項式段一様な関数族によって認識される言語は PSPACE と一致する.

Lemma 3.2. 対数段の一様な差分方程式族によって認識される CH 困難な言語が存在する.

計数階層 CH の定義と差分方程式の関係及び補題 3.2 の証明は 3.2 節にて述べる.

3.2 The Counting Hierarchy and Difference Equations of Logarithmic Height

The polynomial hierarchy PH is defined using non-deterministic polynomial-time oracle Turing machines:

$$\Sigma_0^p = \mathsf{P}, \qquad \Sigma_{n+1}^p = \mathsf{NP}^{\Sigma_n^p}, \qquad \mathsf{PH} = \bigcup_n \Sigma_n^p. \tag{3.3}$$

In the same way, the counting hierarchy CH [13] is defined using probabilistic polynomial-time oracle Turing machines[*4]:

$$\mathsf{C}_0\mathsf{P} = \mathsf{P}, \qquad \mathsf{C}_{n+1}\mathsf{P} = \mathsf{PP}^{\mathsf{C}_n\mathsf{P}}, \qquad \mathsf{CH} = \bigcup_n \mathsf{C}_n\mathsf{P}. \tag{3.4}$$

It is known that $\mathsf{PH} \subseteq \mathsf{CH} \subseteq \mathsf{PSPACE}$, but we do not know whether $\mathsf{PH} = \mathsf{PSPACE}$.

Each level of the counting hierarchy has a complete problem defined as follows. For every formula $\phi(X)$ with the list X of l free propositional variables, we write

$$\mathsf{C}^m X\phi(X) \longleftrightarrow \sum_{X\in\{0,1\}^l} \phi(X) \geq m, \tag{3.5}$$

where $\phi(X)$ is identified with the function $\phi\colon \{0,1\}^l \to \{0,1\}$ such that $\phi(X) = 1$ if and only if $\phi(X)$ is true. This "counting quantifier" C^m generalizes the usual quantifiers $\exists$ and $\forall$, because $\mathsf{C}^1 = \exists$ and $\mathsf{C}^{2^l} = \forall$. For lists X_1, ldots, X_n of variables and a formula $\phi(X_1,\dots,X_n)$ with all free variables listed, we define

$$\langle \phi(X_1,\dots,X_n), m_1,\dots,m_n\rangle \in \mathsf{C}_n B_{be} \longleftrightarrow \mathsf{C}^{m_1}X_1\cdots\mathsf{C}^{m_n}X_n\phi(X_1,\dots,X_n). \tag{3.6}$$

Lemma 3.3 ([13, Theorem 7])**.** For every $n \geq 1$, the problem $\mathsf{C}_n B_{be}$ is $\mathsf{C}_n\mathsf{P}$-complete.

We define a problem $\mathsf{C}_{\log} B_{be}$ by

$$\langle 0^{2^n}, u\rangle \in \mathsf{C}_{\log} B_{be} \longleftrightarrow u \in \mathsf{C}_n B_{be}. \tag{3.7}$$

We show that $\mathsf{C}_{\log} B_{be}$ is CH-hard and recognized by a logarithmic-height uniform function family, as required in Lemma 3.2.

Proof of Lemma 3.2. First we prove that $\mathsf{C}_{\log} B_{be}$ is CH-hard. For each problem A in CH, there is a constant n such that $A \in \mathsf{C}_n\mathsf{P}$. From Lemma 3.3, for each $u \in \{0,1\}^*$ there is a polynomial-time function f_n such that $u \in A \leftrightarrow f_n(u) \in \mathsf{C}_n B_{be}$. So

$$u \in A \longleftrightarrow \langle 0^{2^n}, f_n(u)\rangle \in \mathsf{C}_{\log} B_{be}. \tag{3.8}$$

[*4] This characterization, introduced by Torán in [12], is different from Wagner's original one.

Since $\langle 0^{2^n}, f_n(\cdot)\rangle$ is polynomial time computable, A is reducible to $\mathsf{C}_{\log}B_{be}$.

Next we construct a logarithmic-height uniform function family $(G_u)_u$ recognizing $\mathsf{C}_{\log}B_{be}$. Let $u = \langle 0^{2^n}, \langle \phi(X_1, \dots, X_n), m_1, \dots, m_n\rangle\rangle$, where n, $m_1, \dots, m_n$ are nonnegative integers and ϕ is a formula. (If u is not of this form, then $u \notin \mathsf{C}_{\log}B_{be}$.)

We write $l_i = |X_i|$ and $s_i = i + \sum_{j=1}^{i} l_j$. For each $i \in \{0, \dots, n\}$ and $Y_{i+1} \in \{0,1\}^{l_{i+1}}, \dots, Y_n \in \{0,1\}^{l_n}$, we write $\phi_i(Y_{i+1}, \dots, Y_n)$ for the truth value of the subformula $\mathsf{C}^{m_i}X_i \cdots \mathsf{C}^{m_1}X_1\phi(X_1, \dots, X_i, Y_{i+1}, \dots, Y_n)$, so that $\phi_0 = \phi$ and $\phi_n() = \mathsf{C}_{\log}B_{be}(u)$. We regard the quantifier C^m as a function from $\mathbf{N}$ to $\{0,1\}$:

$$C^m(x) = \begin{cases} 1 & \text{if } x \geq m, \\ 0 & \text{if } x < m. \end{cases} \tag{3.9}$$

Thus,

$$\phi_{i+1}(Y_{i+2}, \dots, Y_n) = C^{m_{i+1}}\left(\sum_{X_{i+1} \in \{0,1\}^{l_i}} \phi_i(X_{i+1}, Y_{i+2}, \dots, Y_n)\right). \tag{3.10}$$

For $T \in \mathbf{N}$, we write T_i for the ith digit of T written in binary, and $T_{[i,j]}$ for the string $T_{j-1}T_{j-2}\cdots T_{i+1}T_i$.

For each $(i, T, Y) \in [n+1] \times [2^{s_n}+1] \times [2^{|u|}]$, we define $G_u(i, T, Y)$ as follows. The first row is given by

$$G_u(0, T, Y) = (-1)^{T_{s_1}}\phi(T_{[1,s_1]}, T_{[s_1+1,s_2]}, \dots, T_{[s_{n-1}+1,s_n]}), \tag{3.11}$$

and for $i \neq 0$, we define

$$G_u(i, T, Y) = \begin{cases} (-1)^{T_{s_{i+1}}}C^{m_i}(Y) & \text{if } T_{[1,s_i+1]} = 10\cdots 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}$$

Define H_u from G_u by (3.1) and (3.2).

We prove by induction on i that $H_u(i, T) \in [2^{l_u}]$ for all T, and that

$$G_u(i, T, H_u(i, V)) = (-1)^{V_{s_{i+1}}}\phi_i(V_{[s_i+1,s_{i+1}]}, \dots, V_{[s_{n-1}+1,s_n]}) \tag{3.13}$$

for all $i \in [n+1]$ and $V \in [2^{s_n}+1]$ such that $V_{[1,s_i+1]} = 10\cdots 0$.

For $i = 0$, the claims follows from (3.11). For the induction step, assume (3.13). We have

$$H_u(i+1, T) = \sum_{V=0}^{T-1} G_u(i, V, H_u(i, V)) = \sum_{V}(-1)^{V_{s_{i+1}}}\phi_i(V_{[s_i+1,s_{i+1}]}, \dots, V_{[s_{n-1}+1,s_n]}), \tag{3.14}$$

where we write $\overline{U}$ for the number represented by string U as binary expression. For each $V < \overline{T_{[s_{i+1}+1,s_n]}0\cdots 0}$ in (3.14), the term $\phi_i(V_{[s_i+1,s_{i+1}]}, \dots, V_{[s_{n-1}+1,s_n]})$ and $-\phi_i(V_{[s_i+1,s_{i+1}]}, \dots, V_{[s_{n-1}+1,s_n]}) = 0$ cancel each other out. Then

$$H_u(i+1, T) = \sum_{X \in \{0,1\}^{l_i}} \phi_i(X, T_{[s_{i+1}+1,s_{i+2}]}, \dots, T_{[s_{n-1}+1,s_n]}). \tag{3.15}$$

From this equation and (3.10), it follows that

$$\begin{aligned} G_u(i+1,T,H_u(i+1,T)) &= (-1)^{T_{s_{i+2}}} C^{m_{i+1}}(H_u(i+1,T)) \\ &= (-1)^{T_{s_{i+2}}} \phi_{i+1}(T_{[s_{i+1}+1,s_{i+2}]},\dots,T_{[s_{n-1}+1,s_n]}). \end{aligned} \tag{3.16}$$

By substituting n for i and 2^{s_n} for T in (3.13), we get $G_u(n,2^{s_n},H_u(n,2^{s_n})) = \phi_n() = \mathsf{C}_{\log}B_{be}(u)$. Hence $H_u(n+1,2^{s_n}+1) = \mathsf{C}_{\log}B_{be}(u)$.

The height $n+1$, the width $2^{s_n}+1$, and the cell size $2^{|u|}$ of G_u are polynomial-time computable from u, and $n+1 \le \log(|0^{2^n}|)+1 \le \log|u|+1$. Hence $(G_u)_u$ is uniform and has logarithmic height. □

The language class recognized by uniform function families with i rows contains $\mathsf{C}_i\mathsf{P}$ (the ith level of the counting hierarchy) and is contained in $\mathsf{C}_{i+1}\mathsf{P}$. While the class $\mathsf{C}_i\mathsf{P}$ is defined by (3.4) using oracle Turing machines, it is also characterized as those languages Karp-reducible to C_iB_{be}, or as those accepted by a polynomial-time alternating Turing machine extended with "threshold states" and having at most i alternations. Likewise, the language class accepted by uniform function families of logarithmic height coincided with languages Karp-reducible to $\mathsf{C}_{\log}B_{be}$ and with those accepted by an extended alternating Turing machine with logarithmic alternations. Since this class contains CH, we only state CH-hard in Lemma 3.4 and Theorem 1.2, but it is not known such class how hard the class is between CH and PSPACE.

3.3 Families of Real Functions Simulating Difference Equations

We show that families of smooth differential equations can simulate PSPACE-hard or CH-hard difference equations stated in previous section.

Before stating Lemma 3.4 and Lemma 3.5, we extend the definition of polynomial-time computability of real function to families of real functions. A machine M *computes* a family $(f_u)_u$ of functions $f_u\colon A \to \mathbf{R}$ indexed by strings u if for any $x \in A$ and any name ϕ_x of x, the function taking v to $M^{\phi_x}(u,v)$ is a name of $f_u(x)$. We say a family of real functions $(f_u)_u$ is polynomial-time if there is a polynomial-time machine computing $(f_u)_u$.

Lemma 3.4. There exist a CH-hard language L and a polynomial μ such that for all $k \ge 1$ and polynomials γ, there are a polynomial ρ and families $(g_u)_u$, $(h_u)_u$ of real functions having following properties that $(g_u)_u$ is polynomial-time computable and for any string u:

(i) $g_u\colon [0,1]\times[-1,1] \to \mathbf{R}$, $h_u\colon [0,1] \to [-1,1]$;
(ii) $h_u(0) = 0$ and $Dh_u(t) = g_u(t,h_u(t))$ for all $t \in [0,1]$;
(iii) g_u is (∞,k)-times continuously differentiable;
(iv) $D^{(i,0)}g_u(0,y) = D^{(i,0)}g_u(1,y) = 0$ for all $i \in \mathbf{N}$ and $y \in [-1,1]$;
(v) $\left|D^{(i,j)}g_u(t,y)\right| \le 2^{\mu(i,|u|)-\gamma(|u|)}$ for all $i \in \mathbf{N}$ and $j \in \{0,\dots,k\}$;

(vi) $h_u(1) = 2^{-\rho(|u|)}L(u)$.

Lemma 3.5. There exist a PSPACE-hard language L and a polynomial μ, such that for all polynomials γ, there are a polynomial ρ and families $(g_u)_u$, $(h_u)_u$ of real functions such that $(g_u)_u$ is polynomial-time computable and for any string u satisfying (i)–(vi) of Lemma 3.4 with $k = 1$.

We will prove Lemma 3.4 using Lemma 3.2 as follows. Let a function family $(G_u)_u$ be as in Lemma 3.2, and let $(H_u)_u$ be the solution of the difference equation given by $(G_u)_u$. We construct h_u and g_u from H_u and G_u such that $h_u(T/2^{q(|u|)}) = \sum_{i=0}^{p(|u|)} H_u(i,T)/B^{d_u(i)}$ for each $T = 0, \ldots, 2^{q(|u|)}$ and $Dh_u(t) = g_u(t, h_u(t))$. The polynomial-time computability of $(g_u)_u$ follows from that of $(G_u)_u$. We can prove Lemma 3.5 from Lemma 3.1 in the same way.

In Lemma 3.4, we have the new conditions (iii)–(v) about the smoothness and the derivatives of g_u that were not present in [3, Lemma 4.1]. To satisfy these conditions, we construct g_u using the smooth function f in following lemma.

Lemma 3.6 ([7, Lemma 3.6])**.** There exist a polynomial-time infinitely differentiable function $f\colon [0,1] \to \mathbf{R}$ and a polynomial s such that

(i) $f(0) = 0$ and $f(1) = 1$;
(ii) $D^n f(0) = D^n f(1) = 0$ for all $n \geq 1$;
(iii) f is strictly increasing;
(iv) $D^n f$ is polynomial-time computable for all $n \geq 1$;
(v) $|D^n f| \leq s(n)$ for all $n \geq 1$.

Although the existence of the polynomial s satisfying the condition (v) is not stated in [7, Lemma 3.6], it can be shown easily.

We only prove Lemma 3.4 here and omit the analogous and easier proof of Lemma 3.5.

Proof of Lemma 3.4. Let a function family $(G_u)_u$ be as in Lemma 3.2, and let a function family $(H_u)_u$ be the solution of the difference equation given by $(G_u)_u$.

By a similar argument to the beginning of the proof[3, Lemma 4.1], we may assume that there exist polynomial-time functions p, j_u and polynomial q, r satisfying following properties:

$$G_u\colon [p(|u|)] \times [2^{q(|u|)}] \times [2^{r(|u|)}] \to \{-1, 0, 1\}; \tag{3.17}$$

$$H_u(i, 2^{q(|u|)}) = \begin{cases} L(u) & (i = p(|u|)) \\ 0 & (i < p(|u|)) \end{cases}; \tag{3.18}$$

$$i \neq j_u(T) \to G_u(i, T, Y) = 0. \tag{3.19}$$

Since G_u has logarithmic height, there exists polynomial σ such that $(k+1)^{p(x)} \leq \sigma(x)$

We construct the families real functions $(g_u)_u$ and $(h_u)_u$ simulating G_u and H_u, in particular, $h_u(T/2^{q(|u|)}) = \sum_{i=0}^{p(|u|)} H_u(i,T)/B^{d_u(i)}$. We define the constant B and the function $d_u\colon [p(|u|)+1] \to \mathbf{N}$ as

$$B = 2^{\gamma(|u|)+r(|u|)+s(k)+k+3} \qquad\qquad d_u(i) = \begin{cases} \sigma(|u|) & (i = p(|u|)) \\ (k+1)^i & (i < p(|u|)). \end{cases} \tag{3.20}$$

For each $(t,y) \in [0,1] \times [-1,1]$, there exists unique $N \in \mathbf{N}$, $\theta \in [0,1)$, $Y \in \mathbf{Z}$ and $\eta \in [-1/4, 3/4)$ such that $t = (T+\theta)2^{-q(|u|)}$ and $y = (Y+\eta)B^{-d_u(j_u(T))}$. Let f and a polynomial s be as in Lemma 3.6, we define $\delta_{u,Y}\colon [0,1] \to \mathbf{R}$, $g_u\colon [0,1]\times[-1,1] \to \mathbf{R}$ and $h_u\colon [0,1] \to [-1,1]$ as

$$\delta_{u,Y}(t) = \frac{2^{q(|u|)}Df(\theta)}{B^{d_u(j_u(T)+1)}} G_u\big(j_u(T), T, Y \bmod 2^{r(|u|)}\big), \tag{3.21}$$

$$g_u(t,y) = \begin{cases} \delta_{u,Y}(t) & (\eta \le \frac{1}{4}) \\ (1 - f(\frac{4\eta-1}{2}))\delta_{u,Y}(t) + f(\frac{4\eta-1}{2})\delta_{u,Y+1}(t) & (\eta > \frac{1}{4}), \end{cases} \tag{3.22}$$

$$h_u(t) = \sum_{i=0}^{p(|u|)} \frac{H_u(i,T)}{B^{d_u(i)}} + \frac{f(\theta)}{B^{d_u(j_u(T)+1)}} G_u\big(j_u(T), T, H_u(j_u(T),T)\big). \tag{3.23}$$

We will verify that $(g_u)_u$ and $(h_u)_u$ defined above satisfy all the conditions stated in Lemma 3.4. The condition(ii) are verified by applying the same argument of [3, Lemma 4.1].

We will show that the condition(iii) holds, that is, g_u is (∞,k)-times continuously differentiable. For each $i \in \mathbf{N}$,

$$D^i\delta_{u,Y}(t) = \frac{2^{(i+1)q(|u|)}D^{i+1}f(\theta)}{B^{d_u(j_u(T)+1)}} G_u\left(j_u(T),\ T,\ Y \bmod 2^{r(|u|)}\right), \tag{3.24}$$

$$D^{(i,0)}g_u(t,y) = \begin{cases} D^i\delta_{u,Y}(t) & (-\frac{1}{4} < \eta < \frac{1}{4}) \\ \left(1 - f\left(\frac{4\eta-1}{2}\right)\right) D^i\delta_{u,Y}(t) + f\left(\frac{4\eta-1}{2}\right) D^i\delta_{u,Y+1}(t) & (\frac{1}{4} < \eta < \frac{3}{4}), \end{cases} \tag{3.25}$$

and for each $j \in \{1,\dots,k\}$,

$$D^{(i,j)}g_u(t,y) = \begin{cases} 0 & (-\frac{1}{4} < \eta < \frac{1}{4}) \\ (2B^{d_u(j_u(T))})^j D^j f(\frac{4\eta-1}{2})(D^i\delta_{u,Y+1}(t) - D^i\delta_{u,Y}(t)) & (\frac{1}{4} < \eta < \frac{3}{4}). \end{cases} \tag{3.26}$$

It is easy to check that they are also continuous on the boundary($\theta = 0$, $\eta = -1/4, 3/4$). Hence g_u is (∞, j)-times continuously differentiable. Substituting $t = 0,1$ ($\theta = 0$) into (3.25), we get $D^{(i,0)}g_u(0,y) = D^{(i,0)}g_u(1,y) = 0$, so the condition (iv) holds.

We show that the condition (v) holds with $\mu(x,y) = (x+1)q(y) + s(x+1)$. Note that μ is a polynomial independent from k or γ. Since $|D^i\delta_{u,Y}(t)| \le 2^{(i+1)q(|u|)+s(i+1)}B^{-d_u(j_u(|u|)+1)}$ by (3.21), for all $i \in \mathbf{N}$ and $j \in \{0,\dots,k\}$,

$$|D^{(i,j)}g_u| \le 2^k B^{k\cdot j_u(T)} 2^{s(k)} \cdot 2 \cdot \frac{2^{(i+1)q(|u|)+s(i+1)}}{B^{d_u(j_u(|u|)+1)}} \le \frac{2^{\mu(i,|u|)+s(k)+k+1}}{B} \le 2^{\mu(i,|u|)-\gamma(|u|)} \tag{3.27}$$

by our choice of B.

We have (vi) with $\rho(x) = \sigma(x) \cdot (\gamma(x) + r(x) + s(k) + k + 3)$ because

$$h_u(1) = \frac{H_u(p(|u|), 2^{q(|u|)})}{B^{d_u(p(|u|))}} = \frac{L(u)}{2^{\sigma(|u|)\cdot(\gamma(|u|)+r(|u|)+s(k)+k+3)}} = 2^{-\rho(|u|)} L(u). \tag{3.28}$$

$\square$

To prove Lemma 3.5, let a function family $(G_u)_u$ be as Lemma 3.1 and a function family $(H_u)_u$ be the solution of the difference equation given by $(G_u)_u$, and define $(g_u)_u$ and $(h_u)_u$ as (3.22) and (3.23) with $d_u(i) = i$. It is shown in the same way above that they match all the conditions stated in Lemma 3.5.

3.4 The Proof of Main Theorems

We describe the sketch of proof. Let $(g_u)_u$ and $(h_u)_u$ be as lemmas in the previous section, the real functions g and h in theorems are constructed from $(g_u)_u$ and $(h_u)_u$ as follows. Divide $[0, 1)$ into infinity many subintervals $[l_u^-, l_u^+]$, with midpoints c_u. We construct h putting a scaled copy of h_u onto $[l_u^-, c_u]$ and putting a horizontally reversed scaled copy of $h_u(t)$ onto $[c_u, l_u^+]$ so that $h(l_u^-) = 0$, $h(c_u) = 2^{-\rho'(|u|)}L(u)$ and $h(l_u^+) = 0$ where ρ' is a polynomial. In the same way, g is constructed from $(g_u)_u$ so that g and h satisfy (1.1).

We omit the proof of Theorem 1.1 since we can get it applying the same argument as the proof of Theorem 1.2 and using Lemma 3.5.

Proof of Theorem 1.2. Let L and μ be as Lemma 3.5. Define

$$\lambda(x) = 2x + 2, \qquad \gamma(x) = x\mu(x, x) + x\lambda(x). \tag{3.29}$$

and for each u

$$\Lambda_u = 2^{\lambda(|u|)}, \qquad c_u = 1 - \frac{1}{2^{|u|}} + \frac{2\bar{u} + 1}{\Lambda_u}, \qquad l_u^{\mp} = c_u \mp \frac{1}{\Lambda_u} \tag{3.30}$$

where $\bar{u} \in \{0, \dots, 2^{|u|} - 1\}$ is the number represented by u as binary expression. Let ρ, $(g_u)_u$, $(h_u)_u$ be as Lemma 3.4 with γ.

We define

$$g\left(l_u^{\mp} \pm \frac{t}{\Lambda_u}, \frac{y}{\Lambda_u}\right) = \begin{cases} \pm \displaystyle\sum_{l=0}^{k} \frac{D^{(0,l)} g_u(t, 1)}{l!} (y - 1)^l & (1 < y) \\ \pm g_u(t, y) & (-1 \le y \le 1) \\ \pm \displaystyle\sum_{l=0}^{k} \frac{D^{(0,l)} g_u(t, -1)}{l!} (y + 1)^l & (1 < y) \end{cases} \tag{3.31}$$

$$h\left(l_u^{\mp} \pm \frac{t}{\Lambda_u}\right) = \frac{h_u(t)}{\Lambda_u} \tag{3.32}$$

for each string u, $t \in [0,1)$, $y \in [-1,1]$. Let $g(1,y) = 0$ and $h(1) = 0$ for any $y \in [-1,1]$

It can be shown in a similar way as the Lipschitz version that g and h satisfy (1.1) and g is polynomial-time computable, so see the proof of [3, Theorem 3.2] for more details. We only show that g is (∞, k)-times continuously differentiable.

For each $i \in \mathbf{N}$ and $j \in \{0,\dots,k\}$ differentiating (3.31) (i,j)-times,

$$D^{(i,j)} g\left(l_u^{\mp} \pm \frac{t}{\Lambda_u}, \frac{y}{\Lambda_u}\right) = \begin{cases} \pm\Lambda_u^{(i,j)} \sum_{l=j}^{k} \frac{D^{(i,l)} g_u(t,1)}{(l-j)!}(y-1)^l & (1 < y) \\ \pm\Lambda_u^{(i,j)} D^{(i,j)} g_u(t,y) & (-1 < y < 1) \\ \pm\Lambda_u^{(i,j)} \sum_{l=j}^{k} \frac{D^{(i,l)} g_u(t,-1)}{(l-j)!}(y+1)^l & (1 < y). \end{cases} \tag{3.33}$$

for $t \in (0,1)$ and $y \in [-\Lambda_u, \Lambda_u]$ where $y \neq -1, 1$. It is continuous where $t \in (0,1)$ and $y \neq -1, 1$ since each $D^{(i,j)} g_u$ is continuous. We can verify that it is continuous on the boundary($t = 0,1$ or $y = -1, 1$) by the definition and Lemma 3.4 (iv). Finally we show that it is continuous at 1 for the first variable. By Lemma 3.4 (v),

$$\begin{aligned} \left| D^{(i,j)} g\left(l_u^{\mp} \pm \frac{t}{\Lambda_u}, \frac{y}{\Lambda_u}\right)\right| &\leq \Lambda_u^{i+j} \sum_{l=j}^{k} |D^{(i,l)} g_u| (\Lambda_u + 1)^l \\ &\leq \Lambda_u^{i+j+k} 2^{k+\mu(i,|u|)-\gamma(|u|)} = 2^{(i+j+k)\lambda(|u|)+k+\mu(i,|u|)-\gamma(|u|)}. \end{aligned} \tag{3.34}$$

While $|u| \to \infty$, (3.34) correspondingly approaches to 0 by our choice of γ. Hence $\lim_{t \to 1-0} D^{(i,j)} g(t,y) = g(1,y) = 0$. and we complete to show that g is (∞,k)-times continuously differentiable. □

4 演算子の計算量

定理 1.1, 1.2 はいづれも関数 g を多項式時間計算可能と仮定した上で解 h の計算量について述べている. しかし微分方程式を「解く」困難さ, すなわち与えられた g から h を求める演算子の計算量は如何であろうか. この問に答えるにはまず実関数を実関数へ写す演算子の計算量を定義することを要する.

実数を入出力する関数の計算量を論ずるには, 実数を文字列関数で表した. 即ち $\mathbf{R}$ の各元の名として文字列関数を使ったのであり, その対応を $\mathbf{R}$ の表現という. 同じように実関数を入出力する演算子の計算量を論ずるには, 実関数を文字列関数で表す. つまり連続な実関数 $h\colon [0,1] \to \mathbf{R}$ の空間 $\mathrm{C}[0,1]$ や, Lipschitz 連続な実関数 $g\colon [0,1] \times [-1,1] \to \mathbf{R}$ の空間 $\mathrm{C_L}[[0,1] \times [-1,1]]$ について, 表現を指定すればよい. 演算子の計算可能性や計算量はその表現に依ることになるが, ここでは [5] に従い, $\mathrm{C}[0,1]$ の表現として $\delta_\square$ を, $\mathrm{C_L}[[0,1] \times [-1,1]]$ の表現として $\delta_{\square\mathrm{L}}$ をそれぞれ用いる. $\delta_\square$ は関数空間 $\mathrm{C}[0,1]$ の表現として或る意味で自然な唯一のものであることが判っており [4], また $\delta_{\square\mathrm{L}}$ は $\delta_\square$ に Lipschitz 定数の情報を附加した表現である.

これらの表現では使われる文字列関数の長さが有界でないため, 神託機械の時間・空間を測る方法を二階多項式を使って拡張し, これに基いて多項式空間 **FPSPACE** などの計算量クラスや, 多項

式時間 Weihrauch 帰着 $\leq_{\mathrm{W}}$ などの帰着, その下での困難性を定義する [5]．この枠組を上述の実関数の表現と組合せることで, 本稿の結果も以下の如く構成的な形で述べることができる．

実関数 $g \in \mathrm{C_L}[[0,1] \times [-1,1]]$ を, (1.1) の解 $h \in \mathrm{C}[0,1]$ に対応させる演算子 ODE を考える．ODE は $\mathrm{C_L}[[0,1] \times [-1,1]]$ から $\mathrm{C}[0,1]$ への部分写像である．[5, 定理 4.9] では表 1.1 第三行の証明を構成的に書き直すことで, ODE が $(\delta_{\square\mathrm{L}}, \delta_{\square})$-**FPSPACE**-$\leq_{\mathrm{W}}$ 完全であることが示された．本稿の定理 1.1 も同じように構成的に書き直すことができる．即ち ODE を (∞, k) 回連続微分可能な入力に制限したものを ODE_k と書くと,

Theorem 4.1. ODE_1 は $(\delta_{\square\mathrm{L}}, \delta_{\square})$-**FPSPACE**-$\leq_{\mathrm{W}}$ 完全.

これを示すには, 定理 1.1 の証明において関数の構成に使われた情報が入力から容易に得られることを確かめればよく, 新たな技巧を要しないから詳細は省略する．この構成的な主張は非構成的な主張よりも強いものであり [5, 補題 3.7, 3.8], 定理 1.1 は定理 4.1 の系として従う．

なお定理 1.2 も同様に構成的な形で成立ち, 各 $k \in \mathbf{N}$ について ODE_k は $(\delta_{\square\mathrm{L}}, \delta_{\square})$-**CH**-$\leq_{\mathrm{W}}$ 困難であるが, この [5] の枠組における **CH** を定義するには相対化の扱いについて今少しの議論を要するので別稿で扱う．

謝辞

本研究を遂行し発表するにあたり河村は科学研究費補助金若手研究 (B) 23700009 による援助を受けた．記して謝意を表する．

参考文献

[1] E.A. Coddington and N. Levinson. *Theory of Ordinary Differential Equations*. McGraw-Hill, 1955.

[2] A. Kawamura. Complexity of initial value problems, 2010. To appear in *Fields Institute Communications*.

[3] A. Kawamura. Lipschitz continuous ordinary differential equations are polynomial-space complete. *Computational Complexity*, 19(2):305–332, 2010.

[4] A. Kawamura. On function space representations and polynomial-time computability. Dagstuhl Seminar 11411: Computing with Infinite Data, 2011. http://www-imai.is.s.u-tokyo.ac.jp/~kawamura/dagstuhl.pdf.

[5] A. Kawamura and S. Cook. Complexity theory for operators in analysis. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pages 495–502. ACM, 2010.

[6] K.I. Ko. On the computational complexity of ordinary differential equations. *Information and Control*, 58(1-3):157–194, 1983.

[7] K.I. Ko. *Complexity Theory of Real Functions*. Birkhäuser Boston, 1991.

[8] K.I. Ko and H. Friedman. Computing power series in polynomial time. *Advances in Applied Mathematics*, 9(1):40–50, 1988.

[9] W. Miller. Recursive function theory and numerical analysis. *Journal of Computer and System Sciences*, 4(5):465–472, 1970.

[10] N.T. Müller. Uniform computational complexity of Taylor series. *Automata, Languages and Programming*, pages 435–444, 1987.

[11] M.B. Pour-el and I. Richards. A computable ordinary differential equation which possesses no computable solution. *Annals of Mathematical Logic*, 17(1-2):61–90, 1979.

[12] J. Torán. Complexity classes defined by counting quantifiers. *Journal of the ACM (JACM)*, 38(3):752–773, 1991.

[13] K.W. Wagner. The complexity of combinatorial problems with succinct input representation. *Acta Informatica*, 23(3):325–356, 1986.

[14] Klaus Weihrauch. *Computable Analysis: An Introduction.* Texts in Theoretical Computer Science. Springer, 2000.