

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§

平成 24 年 1 月 31 日

概要

常微分方程式 $h(0) = 0$, $h'(t) = g(t, h(t))$ の解 h の計算量と、関数 g の計算量及び制限の関係は、常微分方程式を数值的に解くことの本質的な難しさを表しているとして調べられている。本稿では河村が 2010 年の論文の中で Lipschitz 条件を満たす多項式時間計算可能な関数 g の常微分方程式の解が PSPACE 困難たりうるという結果をしめすために用いた手法を、微分可能な g に拡張する。これにより多項式時間計算可能で 1 回連続微分可能な関数の常微分方程式が、PSPACE 困難な解を持ちうることがわかる。また任意の k について、多項式時間計算可能で k 回微分可能な関数の常微分方程式は、本稿で定義される計算量 DIVP(log) について困難な解を持ちうことをしめす。DIVP(log) = PSPACE であるかどうかは未解決である。

1 導入

1.1 計算可能解析

計算可能解析 (Computable Analysis) では計算可能性理論や計算量理論の視点から解析学を扱う。「計算可能な実数」や「多項式時間計算可能な実関数」といった概念を定義し、解析学に現れる様々な実数や実関数の本質的な難しさを分析する。

離散的な対象においては「計算できる関数」はモデルによらず、すべてチューリング機械で計算できるものと同値であることが知られているが、実数計算においては、計算できる関数が互いに異なる、いくつかのモデルが提唱されている。本稿では、「機械が実関数を計算する」ことを次のように定義する。

実関数 $f: \mathbb{R} \rightarrow \mathbb{R}$ を計算するといっても、実数は有限の文字列で表されないため、機械がその完全な値を読み書きすることはできない。

* 東京大学

† 第 1 著者に同じ

‡ ダルムシュタット工科大学

§ 第 3 著者に同じ

表 1.1 多項式時間計算可能関数 g の常微分方程式 (1.1) の解 h の計算量

制限	上界	下界
—	—	計算不可能たりうる [9]
h が g の唯一解	計算可能 [1]	任意の時間がかかりうる [5] [8]
Lipschitz 条件を満たす	PSPACE	PSPACE 困難になりうる [4]
$\mathcal{D}^{(0,1)}g$ が連続	PSPACE	PSPACE 困難たりうる [本稿]
$\mathcal{D}^{(0,i)}g$ が連続	PSPACE	DIVP(log) 困難たりうる [本稿]
g が解析的	多項式時間 [7] [3]	—

そこで実数を近似値の列で表現する. 有理数の列 $(r_n)_n$ が x を表現するとは, $(r_n)_n$ が x へ速く収束すること, すなわち $|r_n - x| \leq 2^{-n}$ を満たすこととする. 数列は $n \in \mathbb{N}$ を $r_n \in \mathbb{Q}$ へ移す関数と考えることもでき, そのような関数または数列を実数の名と呼ぶ.

実関数を計算するモデルとしては神託チューリング機械を用いる. ある機械が関数 $f: [0, 1] \rightarrow \mathbb{R}$ を計算するとは, 入力となる実数 x の名を神託として与えられ, 求める精度 n を入力として与えられたとき, 有理数 s_n で $|s_n - f(x)| \leq 2^{-n}$ を満たすものを出力することとする.

この神託機械の資源を制限することで, 多項式時間 (P) や多項式領域 (PSPACE) に対応する実関数のクラスを定義できる (2 節).

1.2 問題と関連研究

連続実関数 $g: [0, 1] \times \mathbb{R} \rightarrow \mathbb{R}$ にたいして次の常微分方程式を考える.

$$h(0) = 0, \quad h'(t) = g(t, h(t)) \quad (t \in [0, 1]) \quad (1.1)$$

本稿では g が多項式時間計算可能であるとき, 解 h がどれほど複雑な実関数になりうるかを考える.

g に他に何の制限も設けない場合, 解 h (一般に一意でない) は計算不能たりうるため, 様々な制限のもと常微分方程式の解の計算量が研究されている [表 1.1]. Lipschitz 条件とは常微分方程式の解の一意性を保証する十分条件である. Lipschitz 条件を満たすとき, 解は多項式領域計算可能であり, PSPACE 困難 (2 節で定義) でありうる. つまり上界と下界が多項式領域で一致しており, 多項式領域完全といえる. 一方で g が解析的であるとき, 常微分方程式の解も解析的となり, 多項式時間計算可能である.

そこで我々はこの隔たりを埋めることを目的として, 滑らかな関数, つまり微分可能な g について h の計算量がどれほどになりうるかを調べ, 以下の結果を得た.

定理 1.1. 多項式時間実関数 $g: [0, 1] \times [-1, 1] \rightarrow \mathbf{R}$ で, $(\infty, 1)$ 回連続的微分可能であり, g の常微分方程式 (1.1) の解 h が PSPACE 困難であるものが存在する.

定理 1.2. 任意の自然数 $k \geq 2$ にたいして, 多項式時間実関数 $g: [0, 1] \times [-1, 1] \rightarrow \mathbf{R}$ で, (∞, k) 回連続的微分可能であり, g の常微分方程式 (1.1) の解 h が DIVP(log) 困難であるものが存在する.

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区間のときについてのみ定義するため, 一般の常微分方程式の関数 $g: [0, 1] \times \mathbf{R} \rightarrow \mathbf{R}$ について考えることができない. しかし 唯一解 h が g の定義域をはみ出さない, つまり $t \in [0, 1]$ について $-1 \leq h(t) \leq 1$ をみたすような g を構成する.

DIVP(log) とは本稿において定義される計算量クラスであり, $\text{DIVP}(\log) \subseteq \text{PSPACE}$ であるが, $\text{DIVP}(\log) = \text{PSPACE}$ は未解決であるため, (∞, k) 階連続的微分可能関数の常微分方程式の解が PSPACE 困難になりうるかは未解決である. 厳密な定義は次節で導入する.

二変数関数 g が (i, j) 階連続的微分可能であるとは, 第一変数について i 回, 第二変数について j 回微分可能であり, その導関数が連続であることと定義する. この定義は一般的な多変数関数における k 階連続的微分可能の定義 (任意の k 階導関数が存在し, それらがすべて連続) とは異なる. $i \geq j$ とすると (i, j) 回連続的微分可能であれば j 回連続的微分可能であるため, 定理 1.1, 1.2 はそれぞれ 1 回連続的微分可能, k 階連続的微分可能と置き換えても成り立つ.

また定理 1.2 において任意の k に対して (∞, k) 階微分可能な関数を考えているが, 一つ関数が任意の k にたいして k 階微分ではない. つまり g が無限回微分可能であると制限しているわけではない. 無限回微分可能な関数に対する常微分方程式の計算量は今後の課題である.

1.3 差分方程式

定理 1.1 と定理 1.2 の証明の中では, まず滑らかな実関数の常微分方程式で模倣可能であるような離散版常微分方程式を考え, その「離散版常微分方程式」がある計算量クラスについて困難であることを示している. この節ではその離散版常微分方程式である, 差分方程式と差分方程式に対応する計算量クラスについて定義する.

$[n] = \{0, \dots, n-1\}$ と表記する. 関数 $G: [P] \times [Q] \times [R] \rightarrow \{-1, 0, 1\}$ にたいして, 関数 $H: [P+1] \times [Q+1] \rightarrow [R]$ が任意の $i \in [P]$, $T \in [Q]$ について以下を満たすとき, H を G の差分方程式の解と呼ぶ.

$$H(i, 0) = H(0, T) = 0 \quad (1.2)$$

$$H(i+1, T+1) = H(i+1, T) + G(i, T, H(i, T)) \quad (1.3)$$

P, Q, R のことをそれぞれ行数, 列数, 欄の大きさと呼ぶ. G と H が常微分方程式の g と h に対応し, $H(i, 0) = 0$ という条件が $h(0) = 0$, 式 (1.3) と同値である $H(i+1, T+1) - H(i+1, T) = G(i, T, H(i, T))$ という条件が $h'(t) = g(t, h(t))$ と対応する.



図 1.1 差分方程式

言語 L にたいして,

$$L(u) = \begin{cases} 1 & u \in L \\ 0 & u \notin L \end{cases} \quad (1.4)$$

と表記する. 言語 L が $(G_u)_u$ の差分方程式によって認識されるとは, 各 u にたいして G_u の段数と列数, 解をそれぞれ P_u, Q_u, H_u としたとき, $H_u(P_u, Q_u) = L(u)$ をみたすこととする [表 1.3].

段数が $|u|$ の多項式で抑えられ, G_u が多項式時間計算可能な差分方程式について考えたい. $(G_u)_u$ が多項式段一様であるとは, 任意の u について, G_u の段数が $|u|$ の多項式で抑えられ, u を入力として見做した関数 $G(u, i, T, Y) = G_u(i, T, Y)$ が $|u|$ の多項式時間計算可能であることと定義する. $(G_u)_u$ が多項式段一様であるとき, G_u は $|u|$ の多項式時間計算可能であることから, 列数及び欄の大きさは $|u|$ の多項式の指数 ($2^{\text{poly}(|u|)}$) で抑えられる.

河村の論文において多項式段一様な関数族の差分方程式族によって認識される言語のクラスは PSPACE であることが示されている.

補題 1.3 (補題 4.7. [4]). 任意の言語 L について以下は同値.

- $L \in \text{PSPACE}$
- 多項式段一様関数族 $(G_u)_u$ で, 差分方程式が L を認識するものが存在する.

さらに段数が対数によって制限された差分方程式を考えたい. $(G_u)_u$ が対数段一様であるとは, G_u の段数が $|u|$ の対数で抑えられ, u を入力として見做した関数 $G(u, i, T, Y) = G_u(i, T, Y)$ が $|u|$ の多項式時間計算可能であることと定義する. 対数段一様関数族 $(G_u)_u$ の差分方程式によって認識される言語のクラスを DIVP(log) と名付ける.

段数が多項式である差分方程式が認識する言語は PSPACE であったため, $\text{DIVP}(\log) \subseteq \text{PSPACE}$ であるが, $\text{DIVP}(\log) = \text{PSPACE}$ か否かは未解決である. 段数が 2 段の差分方程式を考えると, H_u の最後の欄の値は $|u|$ の多項式サイズの 各 $t \in \{0, 1\}^*$ について $G_u(0, t, 0) \in \{-1, 0, 1\}$ の和になっている. つまり $\#P$ と同程度の能力を持つと言える. よって DIVP(log) は $\#P$ よりも強いクラスであると考えられる.

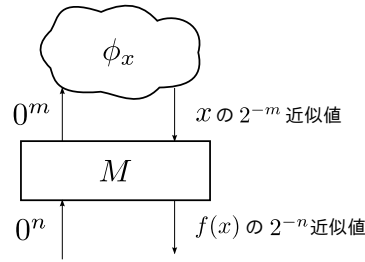


図 2.1 実関数を計算する機械

2 準備

2.1 表記

自然数の集合を \mathbf{N} , 整数の集合を \mathbf{Z} , 実数の集合を \mathbf{R} , 有理数の集合を \mathbf{Q} , $\{0\}^* = \{0^n \mid n \in \mathbf{N}\}$ で表す.

$A \subset \mathbf{R}$ とする. 一変数関数 $f: A \rightarrow \mathbf{R}$ が i 回微分可能であるとき, その i 階導関数を $\mathcal{D}^{(i)}f$ と表記する.

二変数関数 $g: A \times B \rightarrow \mathbf{R}$ が (i, j) 回連続微分可能であるとき, 第一変数について i 階, 第二変数について j 階の導関数はその微分の順序によらず等しい [10]. その導関数を $\mathcal{D}^{(i,j)}g$ で表す.

実関数 $f: A \rightarrow \mathbf{R}$ にたいして $|f| = \sup_{x \in A} f(x)$ と書く.

2.2 実数の名

実数は有限な文字列に符号化できない. そこで文字列から文字列への関数に符号化する.

定義 2.1 (実数の名). 関数 $\phi: \{0\}^* \rightarrow \mathbf{Z}$ が実数 $x \in [0, 1]$ の名であるとは, $\phi(0^n) = \lfloor x \cdot 2^n \rfloor$ または $\phi(0^n) = \lceil x \cdot 2^n \rceil$ を満たすこと.

ここで $\lfloor \cdot \rfloor, \lceil \cdot \rceil$ とはそれぞれ整数への切り捨て関数と切り上げ関数である. つまり実質的には実数 x の名は, サイズ n の入力を受け取ると, 精度 n 桁の x の近似値を返す. 以下では ϕ の値を二進数で表すことにし, ϕ を文字列から文字列への関数として扱う.

2.3 計算可能実関数, 多項式時間実関数

実数を受け取り実数を返す関数を機械が計算するとはどういうことが定義しよう. 実数自体が関数として符号化されているため, それを読み書きする機構として, 神託チューリング機械 (以下単に機械という) を使う [図 2.3].

機械 M に, 文字列から文字列への関数 ϕ を神託として与え, 文字列 0^n を入力として与えたとき,

出力される文字列を $M^\phi(0^n)$ で表す. つまり M^ϕ をやはり文字列から文字列への関数とみる.

定義 2.2. A を \mathbf{R} の有界閉区間とする. 機械 M が実関数 $f: A \rightarrow \mathbf{R}$ を計算するとは, 任意の実数 $x \in A$, 任意の x の名 ϕ_x にたいして, M^{ϕ_x} が $f(x)$ の名であること.

A が \mathbf{R}^2 のコンパクトな部分集合であるときにも, 神託を二つ取る機械を考えて同様に定義する. 計算可能な実関数は Grzegorzczyk によって初めて形式的に定義され, [2].

ある実関数が計算可能であるとは, その関数を計算する神託機械が存在することである. 同様に, ある実関数が多項式時間計算可能であるとは, その関数を計算する多項式時間神託機械が存在することである.

文字列 u で添字づけられた実関数 $f_u: A \rightarrow \mathbf{R}$ の族 $(f_u)_u$ を機械 M が計算するとは, 任意の実数 $x \in A$, 任意の x の名 ϕ_x にたいして, 文字列 v を $M^{\phi_x}(u, v)$ へ移す関数が, $f_u(x)$ の名であることをいう. 実関数族が多項式時間計算可能であるとは, その実関数族を計算する多項式時間神託機械が存在することである.

神託機械 M で f を計算するとき, 求める精度 n にたいして, x の近似値に必要な精度 m が定まるため, 計算可能な関数は連続である. また n と m の対応関係と有理数における近似値を与えることで, 計算可能実関数や多項式時間計算可能実関数にたいして, 神託機械を用いない同値な特徴付けが可能である.

補題 2.3. 実関数 $f: [0, 1] \rightarrow \mathbf{R}$ にたいして, $\phi_f: (\mathbf{Q} \cap [0, 1]) \times \{0\}^* \rightarrow \mathbf{Q}$, $m_f: \mathbf{N} \rightarrow \mathbf{N}$ は

$$|\phi_f(d, 0^n) - f(d)| \leq 2^{-n} \quad (d \in (\mathbf{Q} \cap [0, 1]), \quad n \in \mathbf{N}) \quad (2.1)$$

$$|x - y| \leq 2^{-p_f(m)} \Rightarrow |f(x) - f(y)| \leq 2^{-m} \quad (x, y \in [0, 1], \quad m \in \mathbf{N}) \quad (2.2)$$

をみたす関数とする.

- f が計算可能であることは, 計算可能な ϕ_f, m_f が存在することと同値である.
- f が多項式時間計算可能であることは, 多項式時間計算可能な ϕ_f , 多項式 m_f が存在することと同値である.

2.4 困難性

実関数の計算量の下界を述べるために, 困難性を定義する.

まず実関数に言語が還元することを定義する. 言語 $L: \{0, 1\}^* \rightarrow \{0, 1\}$ が実関数 $f: [0, 1] \rightarrow \mathbf{R}$ に多項式時間還元可能であるとは, f を計算する機械を使って $L(u)$ を多項式時間で計算可能であることである. つまり f を計算する機械があるとしたとき, 入力 u にたいして, 精度 n をこの機械に与え, ある実数 x_u の神託を模倣し, $f(x_u)$ の n 桁近似値から, u が L に属するか否かを多項式時間で計算可能であることである [図 2.4]. 厳密には以下のように定義する.

定義 2.4 (多項式時間還元可能). 言語 L が実関数 $f: [0, 1] \rightarrow \mathbf{R}$ に多項式時間還元可能であると



図 2.2 言語 L から関数 f への還元

は, 多項式時間計算可能な関数 R, S, T が存在し, 任意の文字列 u にたいして以下を満たすことをいう.

- $S(u, \cdot)$ はある実数 x_u の名
- $f(x_u)$ の任意の名 ϕ にたいして

$$L(u) = R(u, \phi(T(u))).$$

以下単に言語が実関数に還元可能といった場合, 多項式時間還元可能を指す. 計算量 C にたいして, 関数 f が C 困難であるとは, C に属する任意の言語が f に還元可能であることと定義する.

3 微分可能関数と常微分方程式

この節では定理 1.1, つまり $(\infty, 1)$ 階連続微分可能な関数の常微分方程式の解は PSPACE 困難でありうることを証明する. 証明の流れとして, まず補題 1.3 から補題 3.1 をしめしたあと, 補題 3.1 から定理 1.1 をしめす. 補題 3.1 では補題 1.3 の関数族 $(G_u)_u, (H_u)_u$ を模倣する実関数族 $(g_u)_u, (h_u)_u$ の存在を述べている. 定理 1.1 の証明では $(g_u)_u, (h_u)_u$ から求める g, h を構成する.

3.1 多項式段一様関数族の差分方程式と PSPACE

任意の PSPACE に含まれる言語 L について, その差分方程式が L を認識する多項式段一様な関数族 $(G_u)_u$ が存在すること (補題 1.3) は河村の論文のなかで示されているが, その証明の概略をしめす.

任意の L ではなく PSPACE 完全な言語である QBF を認識する $(G_u)_u$ を構成することにより, 任意の L を認識する多項式段一様な関数族 $(G_u)_u$ が存在することをしめす. ここで QBF とは, 文字列 u を $\psi = Q_1 x_1 \cdots Q_n x_n \phi(x_1, \dots, x_n)$ と解釈したとき $u \in \text{QBF} \Leftrightarrow \psi = T$ によって定義される言語である. ただし $Q_i \in \{\exists, \forall\}$, $\phi(x_1, \dots, x_n)$ は x_i 以外の変数を含まない論理式.

論理式 $\psi = Q_1 x_1 \cdots Q_n x_n \phi(x_1, \dots, x_n)$ の値を \vee, \wedge によってラベル付された二分木によって計算することを考える. 量子化子 Q_1 を除き x_1 を T と F に置き換えた式をそれぞれ $\psi_T = Q_2 x_2 \cdots Q_n x_n \phi(T, x_2, \dots, x_n)$, $\psi_F = Q_2 x_2 \cdots Q_n x_n \phi(F, x_2, \dots, x_n)$ と置くと $Q_1 = \forall$ ならば $\psi = \psi_T \wedge \psi_F$, $Q_1 = \exists$ ならば $\psi = \psi_T \vee \psi_F$. つまり変数の 1 つ少ない 2 つの論理式と量子化

子によってもとの論理式の値も決まる。再帰的に繰り返すことで計算可能であり、それは深さ n の二分木を葉からたどっていくことと同じである。この過程は二分木の深さが段数に、幅が列数に対応する形で多項式段一様な関数による差分方程式で模倣可能であるため、QBF を認識する多項式段一様な関数の差分方程式が存在する。

3.2 離散初期値問題を模倣する関数族

証明の基本的な流れとして、まず任意の言語 $L \in \text{PSPACE}$ にたいし、補題 1.3 を用いて L を認識する差分方程式 $(G_u)_u$ 及びその解 $(H_u)_u$ を得る。そして各 G_u, H_u から、その差分方程式の計算過程を模倣する $(\infty, 1)$ 階連続微分可能な $g_u: [0, 1] \times [-1, 1] \rightarrow \mathbf{R}$ とその常微分方程式の解 $h_u: [0, 1] \rightarrow \mathbf{R}$ を構成する。 $(G_u)_u$ の一様性から $(g_u)_u$ の多項式時間計算可能性をしめす。

ここで実関数族 $(g_u)_u$ ($g_u: [0, 1] \times [-1, 1] \rightarrow \mathbf{R}$) が多項式時間計算可能であることを、実関数の多項式時間計算可能性から自然に定義する。つまりある神託機械 M が実関数族 $(g_u)_u$ を計算するとは、実数 t, y の名 ϕ, ψ を神託として受けとり、文字列 u , 求める精度 n を入力として受けとったとき、 $|M^{\phi, \psi}(u, 0^n) - g_u(t, y)| \leq 2^{-n}$ を満たすことである。そして実関数族が多項式時間計算可能であるとは、実関数族を計算し、 $|u|$ と n の多項式時間で動作する神託機械が存在することと定義する。

そうして得た $(g_u)_u$ と $(h_u)_u$ から $(\infty, 1)$ 階連続微分可能な $g: [0, 1] \times [-1, 1] \rightarrow \mathbf{R}$ とその常微分方程式の解で PSPACE 困難な $h: [0, 1] \rightarrow \mathbf{R}$ を構成する。各 $h_u(1)$ には $L(u)$ の情報が含まれるため、すべての g_u, h_u をそれぞれ一つの関数 g, h に埋め込みたい、そこで $[0, 1]$ を無限の区間に分割し、各 u に対応する区間に g_u, h_u を縮小して埋め込む。また次の文字列 u' の計算に影響を与えないために、 h_u を t 方向 (第二変数方向) について反転したものも埋め込むことで、影響を相殺する。

ここまで説明した証明の流れは基本的に、河村によるリブシッツ連続な関数の常微分方程式の解が PSPACE 困難たりうることの証明とかわらない。

違いの一つ目は g_u が単に $(\infty, 1)$ 階連続微分可能であるだけでは、縮小して埋め込まれる g が $(\infty, 1)$ 階連続微分可能であるとは限らないため、より特殊な条件をつけたことである (補題 3.1 (v))。

補題 3.1. 任意の言語 $L \in \text{PSPACE}$ にたいして、係数のみに i を含む多項式 μ_i が存在して、任意の多項式 γ にたいして、多項式 ρ , 関数族 $(g_u)_u, (h_u)_u$ で、 $(g_u)_u$ は多項式時間計算可能であり、各二進文字列 u にたいして以下を満たすものが存在する。

- (i) $g_u: [0, 1] \times [-1, 1] \rightarrow \mathbf{R}, \quad h_u: [0, 1] \rightarrow [-1, 1];$
- (ii) h_u は g_u の常微分方程式 (1.1) の解;
- (iii) g_u は $(\infty, 1)$ 階連続微分可能;
- (iv) 任意の $i \in \mathbf{N}, y \in [-1, 1]$ にたいして

$$\mathcal{D}^{(i,0)} g_u(0, y) = \mathcal{D}^{(i,0)} g_u(1, y) = 0$$

(v) 任意の $i \in \mathbf{N}$ にたいして

$$|\mathcal{D}^{(i,1)}g_u| \leq 2^{\mu_i(|u|)-\gamma(|u|)}, \quad |\mathcal{D}^{(i,0)}g_u| \leq 2^{\mu_i(|u|)-\gamma(|u|)}$$

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

また本の論文では連続なだけの関数によって構成していた g_u, h_u を滑らかな多項式時間実関数 $f: [0, 1] \rightarrow \mathbf{R}$ を用いて構成することで、滑らかにしている.

補題 3.2 (補題 3.6. [6]). 以下を満たす多項式時間無限回微分可能実関数 $f: [0, 1] \rightarrow \mathbf{R}$ が存在する.

- (i) $f(0) = 0, \quad f(1) = 1;$
- (ii) 任意の $n \geq 1$ で $f^{(n)}(0) = f^{(n)}(1) = 0;$
- (iii) f は $[0, 1]$ で単調増加;
- (iv) 任意の $n \geq 1$ で $f^{(n)}$ は多項式時間実関数.

3.3 定理 1.1 の証明

4 任意回微分可能関数と常微分方程式

この節では定理 1.2, つまり (∞, k) 階連続微分可能な関数の常微分方程式の解は $\text{DIVP}(\log)$ 困難でありうることを証明する. 証明の流れは節 3 とほぼ同じく, まず補題 3.1 をしめしたあと, 補題 4.1 から定理 1.2 をしめす. 補題 4.1 は節 3 の補題 3.1 に相当するもので, $\text{DIVP}(\log)$ 困難な関数族 $(G_u)_u, (H_u)_u$ を模倣する実関数族 $(g_u)_u, (h_u)_u$ の存在を述べている. 定理 1.2 の証明では $(g_u)_u, (h_u)_u$ から求める g, h を構成する.

4.1 対数深さ離散初期値問題を模倣する関数族

証明の流れは $(\infty, 1)$ 階連続的微分可能の時と変わらない. 任意の言語 $L \in \text{DIVP}(\log)$, 文字列 u にたいして, 上記の対数深さ離散初期値問題を模倣し $L(u)$ を計算する任意回微分可能な実関数 g_u を構成する.

補題 4.1. 任意の自然数 $k \geq 2$, 任意の言語 $L \in \text{DIVP}(\log)$ にたいして, 係数のみに i を含む多項式 μ_i が存在して, 任意の多項式 γ にたいして, 関数 $\rho: \mathbf{N} \rightarrow \mathbf{N}$, 関数族 g_u, h_u で, $\rho, (g_u)_u$ は多項式時間計算可能であり, 各二進文字列 u にたいして以下を満たすものが存在する.

- (i) $g_u: [0, 1] \times [-1, 1] \rightarrow \mathbf{R}, \quad h_u: [0, 1] \rightarrow [-1, 1];$
- (ii) h_u は g_u の常微分方程式 (1.1) の解;
- (iii) g_u は (∞, k) 階連続微分可能;

(iv) 任意の $i \in \mathbf{N}$, $y \in [-1, 1]$ にたいして

$$\mathcal{D}^{(i,0)}g_u(0, y) = \mathcal{D}^{(i,0)}g_u(1, y) = 0$$

(v) 任意の $i \in \mathbf{N}$, $j \in \{0, \dots, k\}$ にたいして

$$\left| \mathcal{D}^{(i,j)}g_u(t, y) \right| \leq 2^{\mu_i(|u|) - \gamma(|u|)}$$

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

4.2 定理 1.2 の証明

補題 3.1 とくらべて補題 4.1 は PSPACE が DIVP(log) に置き換わり, $(\infty, 1)$ 回連続微分可能が (∞, k) 回連続微分可能に一般化されていること以外は変わらない. この関係は定理 1.1 と定理 1.2 との間の関係と等しい. よって証明も PSPACE を DIVP(log) で置き換え, $(\infty, 1)$ 回連続微分可能を (∞, k) 回連続微分可能に一般化させることで, 定理 1.1 の証明から定理 1.2 の証明が構成できる. ^{*1}

5 謝辞

参考文献

- [1] E.A. Coddington and N. Levinson. *Theory of ordinary differential equations*. McGraw-Hill, 1955.
- [2] A. Grzegorzczk. Computable functionals. *Fund. Math*, 42(19553):168–202, 1955.
- [3] A. Kawamura. Complexity of initial value problems, 2010.
- [4] A. Kawamura. Lipschitz continuous ordinary differential equations are polynomial-space complete. *Computational Complexity*, 19(2):305–332, 2010.
- [5] K.I. Ko. On the computational complexity of ordinary differential equations*. *Information and control*, 58(1-3):157–194, 1983.
- [6] K.I. Ko. *Complexity theory of real functions*. Birkhauser Boston Inc., 1991.
- [7] K.I. Ko and H. Friedman. Computing power series in polynomial time. *Advances in Applied Mathematics*, 9(1):40–50, 1988.
- [8] W. Miller. Recursive function theory and numerical analysis*. *Journal of Computer and System Sciences*, 4(5):465–472, 1970.
- [9] 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.

^{*1} なのでこの章まるまる必要ない. しかしこちらの証明のほうが 1 回微分可能のときの証明より一般化されているため, どちらを採用するか要検討

[10] 高木貞治. 解析概論. 岩波書店, 1968.