Boundedness of the domain of definition is undecidable for polynomial ODEs

Daniel S. Graça
DM/FCT, University of Algarve, C. Gambelas, 8005-139
Faro, Portugal and SQIG/IT Lisbon, Portugal

Jorge Buescu
DM/FCUL, University of Lisbon, Portugal and
CAMGSD, Lisbon, Portugal

Manuel L. Campagnolo
DM/ISA, Technical University of Lisbon, 1349-017
Lisbon, Portugal and SQIG/IT Lisbon, Portugal

July 2, 2007

Abstract

Consider the initial-value problem with computable parameters

$$\begin{cases} \frac{dx}{dt} = p(t,x) \\ x(t_0) = x_0, \end{cases}$$

where $p : \mathbb{R}^{n+1} \to \mathbb{R}^n$ is a vector of polynomials and $(t_0, x_0) \in \mathbb{R}^{n+1}$. We show that the problem of determining whether the maximal interval of definition of this initial-value problem is bounded or not is in general undecidable.

1 Introduction

Differential equations are a powerful tool to model natural phenomena. Their use is transversal to many fields and applications can be found ranging from fields like physics or chemistry up to biology or economics. Among these, much attention is devoted to analytic ordinary differential equations (ODEs) that yield initial-value problems

$$\begin{cases} x' = f(t,x) \\ x(t_0) = x_0 \end{cases} \tag{1}$$

where $f : \mathbb{R}^{n+1} \to \mathbb{R}^n$ is an analytic function. In practical applications, like Lorenz equations in meteorology or Lotka-Volterra equations for predator-prey

1

system [HS74], this IVP often reduces to an IVP of the type

$$\begin{cases} x' = p(t, x) \\ x(t_0) = x_0 \end{cases} \tag{2}$$

where p is a vector of polynomials. Indeed, it can be shown (see Theorem 10) that if each component of f in (1) can be written as the composition of many of the standard function of analysis like polynomials, sin, e^x, etc. (in analytical terms, as an element of the elementary function field), then (1) can be written as an equivalent system of the type (2) of suitably larger degree.

It is well known from the basic existence-uniqueness theory of ODEs [CL55], [Lef65] that if f is analytic, then the IVP (1) has a unique solution $x(t)$ defined on a maximal interval of existence $I = (\alpha, \beta) \subset \mathbb{R}$ that is analytic on I [Arn78]. Actually, f only needs to be continuous and locally Lipschitz in the second argument so that this maximal interval exists. The interval is maximal in the sense that either $\alpha = -\infty$ or $x(t)$ is unbounded as $t \to \alpha^+$ (similar conditions apply to β).

A question of interest is the following: is it possible to design an automated method that, on input (f, t_0, x_0), gives as output the maximal interval of existence for the solution of (1)? In computability theory, e.g. [Sip97], [HMU01], it is well known that some problems cannot be answered by the use of an algorithm (more precisely, by the use of a Turing machine). Such problems are labeled *undecidable* and many examples are known. The most prominent undecidable problem is the Halting Problem: given a universal Turing machine and some input to it, decide whether the machine eventually halts or not. To address this kind of questions for IVPs, we have to resort to notions of computability over the reals. In particular, we use the computable analysis approach [PER89], [Ko91], [Wei00]. The idea underlying this theory is to compute an approximation of the output with arbitrary precision from a suitable approximation of the input. More details can be found in Section 2. Using that approach, it was shown in [GZB07] that given an analytic IVP (1), defined with computable data, its corresponding maximal interval may be noncomputable.

Noncomputability results related to initial-value problems of differential equations are not new. For example, Pour-El and Richards [PER79] showed that if we relax the condition of analyticity in the IVP (1) defined with computable data, it can have noncomputable solutions. In [PER81], [PEZ97] it is shown that there is a three-dimensional wave equation, defined with computable data, such that the unique solution is nowhere computable. However, in these examples, noncomputability is not "genuine" in the sense that the problems under study are ill-posed: either the solution is not unique or it is unstable [WZ02]. In other words, ill-posedness was at the origin of noncomputability in those examples. In contrast, an analytic IVP (1) is classically well-posed and, consequently, the noncomputability results do not seem to reflect computational and well-posedness deficiencies inherited by the problems.

Motivated by the noncomputability result obtained in [GZB07], this latter paper also addresses the following problem: while it is not possible to compute

the maximal interval of (1) is it possible to compute some partial information about it? In particular, is it possible to decide if this maximal interval is bounded or not?

This question has interest on its own for the following reason. In many problems, we implicitly assume that t is defined for "all time". For example, if one wants to compute things like sinks or limit cycles associated with ODEs, this only makes sense if the solution of the ODE is defined for all times $t > t_0$. This is also implicitly assumed in problems like reachability [AD90], [AMP95], [HKPV98], [Bou99], [BT00], etc. For this reason, those problems only make sense when associated with ODEs for which the maximal interval is unbounded. So, it would be interesting to know which are the "maximal" classes of functions f for which the boundedness problem is decidable.

In [GZB07], it was shown that for the general class of analytic functions, the boundedness problem of the maximal interval is undecidable. Here we will deepen this result: we will show that the boundedness problem is still undecidable if we pick the class of polynomial functions of degree 56. This result is shown using different techniques from those employed in [GZB07]. Since the boundedness problem is decidable for linear differential functions this suggests that the boundary between decidability/undecidability lies in the class of polynomials of degree n, for some $2 \leq n \leq 56$.

2   Preliminaries

Let us recall some results from the theory of ODEs. In particular, we review the notion of maximal interval for ODEs.

Definition 1 *Let $E \subseteq \mathbb{R}^l$ be an open set. A function $f : E \to \mathbb{R}^n$ is called locally Lipschitz on E if for every compact set $\Lambda \subseteq E$ there is a constant $K_\Lambda \geq 0$ such that*
$$\|f(x) - f(y)\| \leq K_\Lambda \|x - y\|, \quad \text{for all } x, y \in \Lambda.$$

Here we deal with the case where $E \subseteq \mathbb{R}^{n+1}$. Hence, when considering a function $f : E \to \mathbb{R}^n$ with argument (t, x), we refer to $t \in \mathbb{R}$ as the first argument and $x \in \mathbb{R}^n$ as the second argument of f.

Definition 2 *Let $E \subseteq \mathbb{R}^{n+1}$ be an open set. A function $f : E \to \mathbb{R}^n$ is called locally Lipschitz in the second argument, on E, if for every compact set $\Lambda \subseteq E$ there is a constant $K_\Lambda \geq 0$ such that*

$$\|f(t, x) - f(t, y)\| \leq K_\Lambda \|x - y\|, \quad \text{for all } (t, x), (t, y) \in \Lambda.$$

The following classical lemma [Hal80] asserts that C^1 functions are locally Lipschitz, and hence locally Lipschitz in the second argument.

Lemma 3 *If $f : E \to \mathbb{R}^n$ is of class C^1 over $E \subseteq \mathbb{R}^l$, then f is locally Lipschitz on E.*

The following result introduces the notion of maximal interval for ODEs and follows as an immediate consequence of the fundamental existence-uniqueness theory for the initial-value problem (1), where the analyticity condition is dropped for f [CL55], [Lef65], [Hal80]

Proposition 4 *Let E be an open subset of $\mathbb{R}^{n+1}$ and assume that $f : E \to \mathbb{R}^n$ is continuous on E and locally Lipschitz in the second argument. Then for each $(t_0, x_0) \in E$, the problem (1) has a unique solution $x(t)$ defined on a maximal interval (α, β), on which it is C^1. The maximal interval is open and has the property that, if $\beta < +\infty$ (resp. $\alpha > -\infty$), either $(t, x(t))$ approaches the boundary of E or $x(t)$ is unbounded as $t \to \beta^-$ (resp. $t \to \alpha^+$).*

Note that, as a particular case, when $E = \mathbb{R}^{n+1}$ and $\beta < \infty$, $x(t)$ is unbounded as $t \to \beta^-$. This will be the case under study in this paper.

Next we recall basic notions from computable analysis. The idea underlying computable analysis is to extend the classical computability theory so that it might deal with real quantities. See [Wei00] for an up-to-date monograph on computable analysis from the computability point of view, [Ko91] for a presentation from a complexity point of view, or [PER89] for a good introduction to the subject.

Definition 5 *A sequence $\{r_n\}$ of rational numbers is called a ρ-name of a real number x if there exist three functions a, b, c from $\mathbb{N}$ to $\mathbb{N}$, such that for all $n \in \mathbb{N}$, $r_n = (-1)^{a(n)} \frac{b(n)}{c(n)+1}$ and*

$$|r_n - x| \leq \frac{1}{2^n}. \tag{3}$$

In the conditions of the above definition, we say that the ρ-name $\{r_n\}$ is given as an oracle to an oracle Turing machine, if the oracle to be used is (a, b, c). The notion of the ρ-name can be extended to $\mathbb{R}^l$: a sequence $\{(r_{1n}, r_{2n}, \ldots, r_{ln})\}_{n \in \mathbb{N}}$ of rational vectors is called a ρ-name of $x = (x_1, x_2, \ldots, x_l) \in \mathbb{R}^l$ if $\{r_{jn}\}_{n \in \mathbb{N}}$ is a ρ-name of x_j, $1 \leq j \leq l$.

Definition 6 *A real number x is called computable if a, b, and c in (3) are computable (recursive) functions.*

Note that the above definition implies that every rational number must be computable. We now define the notion of computable function over $\mathbb{R}$.

Definition 7 *A function $f : \mathbb{R}^m \to \mathbb{R}^k$ is computable if there is an oracle Turing machine such that for any input $n \in \mathbb{N}$ (accuracy) and any ρ-name of $x \in E$ given as an oracle, the machine will output a rational vector r satisfying $\|r - f(x)\|_\infty \leq 2^{-n}$, where $\|(y_1, \ldots, y_l)\|_\infty = \max_{1 \leq i \leq l} |y_i|$ for all $(y_1, \ldots, y_l) \in \mathbb{R}^l$.*

In particular, it is not difficult to show that polynomials having computable coefficients are computable functions.

4

3 Simulating Turing machines

Several authors have proved that finite dimensional maps can simulate Turing machines. The general approach is to associate each configuration of a Turing machine to a point of $\mathbb{R}^n$, and to show that there is a dynamical system with state space in $\mathbb{R}^n$ that embeds its evolution. This is done, for example, in [Moo90], [KCG94], [SS95], [KM99]. Here we use the approach presented in [GCB05].

Without loss of generality, consider a Turing machine M using 10 symbols, the blank symbol $B = 0$, and symbols $1, 2, ...9$. Let

$$...B\,B\,B\,a_{-k}\,a_{-k+1}...\,a_{-1}\,a_0\,a_1...\,a_n\,B\,B\,B...$$

represent the tape contents of the Turing machine M. We assume that the head reads one of the symbols, a_0, and $a_i \in \{0, 1, ..., 9\}$ for all i. We also suppose that M has m states, represented by numbers 1 through m. For convenience, we consider that if the machine reaches a halting configuration it moves to the same configuration. We assume that, in each transition, the head either moves to the left, moves to the right, or does not move. Take

$$y_1 = a_0 + a_1 10 + ... + a_n 10^n \qquad y_2 = a_{-1} + a_{-2} 10 + ... + a_{-k} 10^{k-1}$$

and let q be the state associated with the current configuration. Then the triple $(y_1, y_2, q) \in \mathbb{N}^3$ gives the current configuration of M. In [GCB05] it is shown that the transition function of M, $\psi_M : \mathbb{N}^3 \to \mathbb{N}^3$, can be embedded into an analytic function $f : \mathbb{R}^3 \to \mathbb{R}^3$, in an error-robust manner.

However, what we would like to do is to simulate the evolution of a Turing machine with a polynomial ODE, to obtain undecidability results about (2). In other words, we would like to have a system (2), where the input x_0 of the Turing machine is coded in the initial condition, and $y(n)$ gives the configuration of the Turing machine at time $n \in \mathbb{N}$, where y is the solution of (2).

This is achieved with the following result.[1]

Theorem 8 ([GCB05]) *Let $\psi_M : \mathbb{N}^3 \to \mathbb{N}^3$ be the transition function of a Turing machine M, under the encoding described above and let ε satisfy $0 < \varepsilon \leq 1/4$. Then there is a computable analytic function $f_M : \mathbb{R}^7 \to \mathbb{R}^6$ such that the ODE $z' = f_M(t, z)$ simulates M in the following sense: for all $x_0 \in \mathbb{N}^3$, the solution $z(t)$ of the IVP defined by the above ODE plus the initial condition (x_0, x_0), defined for $t_0 = 0$, satisfies*

$$\left\| z_2(k) - \psi_M^{[k]}(x_0) \right\|_\infty \leq \varepsilon,$$

for all $k \in \mathbb{N}$, where $z \equiv (z_1, z_2)$ with $z_1, z_2 \in \mathbb{R}^3$.

Remark 9 *In the conditions of the above theorem:*

---

[1] We take $\|f\|_\infty = \sup_{x \in \mathbb{R}} \|f(x)\|_\infty$, where f is a real function. If $f : A \to A$ is a function, then $f^{[k]}$ denotes the kth iterate of f.

1. f_M is a composition of polynomials, sin, cos, arctan;

2. although Theorem 8 only gives the behavior of z on integer values of t, its behavior in between is essentially monotone. More precisely, on any interval $[k, k+1]$, z_1 varies componentwise monotonically from $\psi_M^{[k]}(x_0)$ to $\psi_M^{[k+1]}(x_0)$, with an error bounded by $\varepsilon + \delta$, where δ can be chosen to be less than $1/4$. The same applies to z_2, which behaves like z_1, but with delay $1/2$. (See [GCB05] and [GCB] for details.)

However, this is not enough for our needs. What we would like to have is that f_M is a vector of polynomials so that we can derive undecidability results about (2). First we remark that sin, cos, and arctan are solutions of IVPs of the type (2). Then the following result, taken from [GCB], becomes useful.

Theorem 10 *Consider the IVP*

$$\begin{cases} x' = f(t, x), \\ x(t_0) = x_0, \end{cases} \qquad (4)$$

where $f : \mathbb{R}^{l+1} \to \mathbb{R}^l$ and each component of f is a composition of polynomials and functions that are solutions of IVP of the type (2). Then there exist $n \geq l$, a polynomial $p : \mathbb{R}^{n+1} \to \mathbb{R}^n$ and a $y_0 \in \mathbb{R}^n$ such that the solution of (4) is given by the first l components of $y = (y_1, ..., y_n)$, where y is the solution of the polynomial IVP

$$\begin{cases} y' = p(t, y), \\ y(t_0) = y_0. \end{cases}$$

The proof of Theorem 10 in [GCB] is constructive, and preserves the computability of the parameters of the IVP. From Theorem 8, the Remark following it, and Theorem 10, we obtain the following result.

Lemma 11 *Let M be a Turing machine with m states, coded as elements of $\{1, \ldots, m\}$, where m is the halting state. Then M can be simulated by an IVP (2) as follows:*

1. *The initial configuration (input) x_0 of the TM is coded in the initial conditions of the IVP: $(x_1(0), \ldots, x_6(0)) = (x_0, x_0)$ and $(x_7(0), \ldots, x_n(0)) = \alpha \in \mathbb{R}^{n-6}$ with $n \geq 6$ and where α is computable;*

2. *Let $q(k) \in \{1, \ldots, m\}$ denote the state of the TM at step k. Then there is one variable x_q of the IVP that approximates $q(k)$ with error bounded by $5/16$ in each time interval $[k, k+1/2]$, where $k \in \mathbb{N}$. This variable updates in a monotone manner in the time interval $[k+1/2, k+1]$.*

3. *The polynomial p in (2) and the value α are computable.*

For point 2, we used Theorem 8 with $\varepsilon = 1/4$ and the Remark following it, where we have set $\delta = 1/16$. If x_q is the variable from z_2 that encodes the state, then we get point 3.

4 The result

We now state and prove the main result of the paper.

Theorem 12 *The following problem is undecidable: "Given $p : \mathbb{R}^{n+1} \to \mathbb{R}^n$ with polynomial components and $(t_0, x_0) \in \mathbb{R} \times \mathbb{R}^n$, where p and (t_0, x_0) are computable, decide whether the maximal interval of the IVP (2) is bounded or not".*

Proof. Let M be a universal Turing machine. Suppose that M has m states, coded as elements of $\{1, \ldots, m\}$, where m is the halting state. From Lemma 11, there is a polynomial IVP (2), with solution x, that satisfies for every $k \in \mathbb{N}$

$$\begin{cases} x_q(t) \leq m - \frac{11}{16} & \text{if } M \text{ has not halted at step } k \text{ and } t \leq k \\ x_q(t) \geq m - \frac{5}{16} & \text{if } M \text{ has already halted at step } k \text{ and } t \geq k. \end{cases} \tag{5}$$

Consider the IVP

$$\begin{cases} z_1' = x_q - (m - 1/2) \\ z_2 = \frac{1}{z_1} \end{cases} \iff \begin{cases} z_1' = x_q - (m - 1/2) \\ z_2' = ((m - 1/2) - x_q)z_2^2 \end{cases} \tag{6}$$

where $z_1(0) = z_2(0) = -1$. Since x_q appears as a component, we assume that this IVP is coupled with the polynomial IVP defined by Proposition 11. It is easy to see that while M hasn't halted, $x_q - (m - 1/2) \leq -3/16$. Thus z_1 keeps decreasing and the IVP is defined in $(0, +\infty)$, i.e. the maximal interval is unbounded, if M never halts.

On the other hand, if M eventually halts, z_1 starts increasing at a rate of at least $3/16$ and will do that forever. So, at some time it will have to assume the value 0. When this happens, a singularity appears for z_2 and the maximal interval is therefore (right-)bounded. For negative values of t just replace t by $(-t)$ in the polynomial IVP (2) and assume t to be positive. It can be shown that the behavior of the system will be similar, and we reach the same conclusions for the left bound of the maximal interval. So M halts iff the maximal interval of the polynomial IVP (6) is bounded, i.e. boundedness is undecidable. ∎

Actually, if we are given the description of a universal Turing machine, we can constructively define a set of polynomial ODEs simulating it and adapt in a constructive manner the proof of Theorem 4. If we use the small universal Turing machine presented in [Rog96], having 4 states and 6 symbols, we obtain the following theorem, whose full proof can be found in [Gra].

Theorem 13 *There is a vector $p : \mathbb{R}^{n+1} \to \mathbb{R}^n$, with $n \geq 1$, constituted by computable polynomials, where each component has degree less than or equal to 56, such that the following problem is undecidable: "Given computable $(t_0, x_0) \in \mathbb{R} \times \mathbb{R}^n$, decide whether the maximal interval of the IVP (2) is bounded or not".*

Let us remark that, while the boundedness problem of the maximal interval for unrestricted polynomial IVPs is in general undecidable, it is not the case for some subclasses of polynomials. For instance, the boundedness problem is

7

decidable for the class of linear differential equations or for the class of one-dimensional autonomous differential equations where f is a polynomial of any degree. It would be interesting to investigate maximal classes with the above property.

Theorem 14 *Consider the IVP (2) with $p(t, x) = A(t)x + h(t)$, where A and h are $m \times m$ and $m \times 1$ matrices, respectively, and each entry $A_{jk} : \mathbb{R} \to \mathbb{R}$, $h_j : \mathbb{R} \to \mathbb{R}$ is a continuous function, for $j, k = 1, \ldots, m$. Then the maximal interval associated with this IVP is $(-\infty, \infty)$. In particular, the boundedness problem is decidable for linear problems.*

Proof. See [Hal80, p. 79]. ■

References

[AD90]   R. Alur and D. L. Dill. Automata for modeling real-time systems. In *Automata, Languages and Programming, 17th International Colloquium*, LNCS 443, pages 322–335. Springer, 1990.

[AMP95]   E. Asarin, O. Maler, and A. Pnueli. Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoret. Comput. Sci.*, 138:35–65, 1995.

[Arn78]   V. I. Arnold. *Ordinary Differential Equations*. MIT Press, 1978.

[Bou99]   Olivier Bournez. Achilles and the Tortoise climbing up the hyper-arithmetical hierarchy. *Theoret. Comput. Sci.*, 210(1):21–71, 1999.

[BT00]   V. D. Blondel and J. N. Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 9(36):1249–1274, 2000.

[CL55]   E. A. Coddington and N. Levinson. *Theory of Ordinary Differential Equations*. McGraw-Hill, 1955.

[GCB]   D. S. Graça, M. L. Campagnolo, and J. Buescu. Computability with polynomial differential equations. *Adv. Appl. Math.* to appear.

[GCB05]   D. S. Graça, M. L. Campagnolo, and J. Buescu. Robust simulations of Turing machines with analytic maps and flows. In S. B. Cooper, B. Löwe, and L. Torenvliet, editors, *CiE 2005: New Computational Paradigms*, LNCS 3526, pages 169–179. Springer, 2005.

[Gra]   D. S. Graça. *Computability with Polynomial Differential Equations.* PhD thesis, Instituto Superior Técnico/Universidade Técnica de Lisboa. submitted.

[GZB07]   D.S. Graça, N. Zhong, and J. Buescu. Computability, noncomputability and undecidability of maximal intervals of IVPs. *Trans. Amer. Math. Soc.*, 2007. To appear.

[Hal80]   J. K. Hale. *Ordinary Differential Equations.* Robert E. Krieger Pub. Co, 2nd edition, 1980.

[HKPV98]   T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? *J. Comput. System Sci.*, 57(1):94–124, 1998.

[HMU01]   J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 2nd edition, 2001.

[HS74]   M. W. Hirsch and S. Smale. *Differential Equations, Dynamical Systems, and Linear Algebra.* Academic Press, 1974.

[KCG94]   P. Koiran, M. Cosnard, and M. Garzon. Computability with low-dimensional dynamical systems. *Theoret. Comput. Sci.*, 132:113–128, 1994.

[KM99]   P. Koiran and C. Moore. Closed-form analytic maps in one and two dimensions can simulate universal Turing machines. *Theoret. Comput. Sci.*, 210(1):217–223, 1999.

[Ko91]   K.-I Ko. *Computational Complexity of Real Functions.* Birkhäuser, 1991.

[Lef65]   S. Lefshetz. *Differential Equations: Geometric Theory.* Interscience, 2nd edition, 1965.

[Moo90]   C. Moore. Unpredictability and undecidability in dynamical systems. *Phys. Rev. Lett.*, 64(20):2354–2357, 1990.

[PER79]   M. B. Pour-El and J. I. Richards. A computable ordinary differential equation which possesses no computable solution. *Ann. Math. Logic*, 17:61–90, 1979.

[PER81]   M. B. Pour-El and J. I. Richards. The wave equation with computable initial data such that its unique solution is not computable. *Adv. Math.*, 39:215–239, 1981.

[PER89]   M. B. Pour-El and J. I. Richards. *Computability in Analysis and Physics.* Springer, 1989.

[PEZ97]   M. B. Pour-El and N. Zhong. The wave equation with computable initial data whose unique solution is nowhere computable. *Math. Log. Quart.*, 43:499–509, 1997.

[Rog96]   Y. Rogozhin. Small universal Turing machines. *Theoret. Comput. Sci.*, 168(2):215–240, 1996.

[Sip97]   M. Sipser. *Introduction to the Theory of Computation*. PWS Publishing Company, 1997.

[SS95]   H. T. Siegelmann and E. D. Sontag. On the computational power of neural networks. *J. Comput. System Sci.*, 50(1):132–150, 1995.

[Wei00]   K. Weihrauch. *Computable Analysis: an Introduction*. Springer, 2000.

[WZ02]   K. Weihrauch and N. Zhong. Is wave propagation computable or can wave computers beat the Turing machine? *Proc. London Math. Soc.*, 85(3):312–332, 2002.