

Computability of ordinary differential equations

Daniel S. Graça

FCT, Universidade do Algarve, C. Gambelas,
8005-139 Faro, Portugal
& Instituto de Telecomunicações, Lisbon, Portugal

Ning Zhong

DMS, University of Cincinnati,
Cincinnati, OH 45221-0025, U.S.A.

October 17, 2018

Abstract

In this paper we provide a brief review of several results about the computability of initial-value problems (IVPs) defined with ordinary differential equations (ODEs). We will consider a variety of settings and analyze how the computability of the IVP will be affected. Computational complexity results will also be presented, as well as computable versions of some classical theorems about the asymptotic behavior of ODEs.

Ordinary differential equations (ODEs) appear in many applications and are used to describe a large variety of phenomena. For that reason much effort has been directed towards solving ODEs. Although one can solve exactly a few classes of ODEs such as linear ODEs, separable first-order ODEs, etc., in practice we need other methods to analyze non-linear ODEs. For example, we can numerically approximate the solution of an initial-value problem (IVP) defined with an ODE and, in some cases, we can also use some qualitative results to better understand the dynamics of the system. For instance the Poincaré-Bendixson theorem rules out chaotic behavior on dynamical systems defined by autonomous ODEs in the plane.

Numerical methods have gained a particular prominence with the advent of the digital computer. Here we will analyze the problem of finding the solution of an IVP defined with an ODE, from a computational perspective.

1 Background

In this paper we consider (autonomous) ODEs with the format

$$y' = f(y) \tag{1}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $y : \mathbb{R} \rightarrow \mathbb{R}^n$ is a solution of the ODE, with $y = (y_1, \dots, y_n)$. We notice that an initial-value problem (IVP) defined with an apparently more general non-autonomous ODE $y' = f(t, y)$ can always be reduced to an initial-value problem defined with an autonomous ODE (1) by introducing

a new component y_{n+1} satisfying $y_{n+1}(0) = 0$ and $y'_{n+1} = 1$, and by replacing t by y_{n+1} in the non-autonomous ODE. Therefore we don't lose any generality by only considering autonomous ODEs.

There are two important classical results on the existence and uniqueness of solutions for initial value problems (IVPs) involving ODEs, i.e. problems of the type

$$\begin{cases} y'(t) = f(y) \\ y(t_0) = y_0 \end{cases} \quad (2)$$

Both results are important when analyzing IVPs (2) from a computability perspective. The first one is Peano's existence theorem which asserts that if f is continuous, then (2) has at least one (local) solution (see e.g. [BR89] for a precise statement of this theorem). While this is a very general theorem (only continuity of f is assumed), it poses some challenges since an IVP (2) can have, under these conditions, an infinite number of solutions. Moreover, it was proved in [PER79] that there exists an IVP (2) defined on the plane, where f is computable and thus continuous — see Section 2 — but which does not admit any computable solution. This result shows that it is not enough to assume continuity of f to ensure computability of a solution of an IVP (2).

For that reason, we need the second classical result, the Picard–Lindelöf theorem, which is an existence *and* uniqueness theorem for IVPs (2). This theorem says that if f is Lipschitz continuous, then (2) has one and only one (local) solution. Moreover, this theorem is constructive and thus can be used to compute the solution of (2) when f is Lipschitz continuous. We recall that a function f is Lipschitz continuous if there is a constant $K > 0$ such that

$$\|f(x) - f(y)\| \leq K \|x - y\| \quad (3)$$

for all x, y in the domain of f .

The idea to prove the Picard–Lindelöf theorem is to first construct a very rough approximation of the solution of (2). By Picard's method one can construct a sequence of approximate solutions which will converge to the solution of (2). Moreover, by using the Lipschitz condition (3), we are able to show that the sequence uniformly converges (locally) to the solution of (2) and that the convergence rate can be expressed in terms of the Lipschitz constant K . From these facts we can conclude that the solution of (2) is locally computable if f is Lipschitz continuous and computable.

However, the previous two results are local. Thus it is natural to look also for global results. When the hypotheses of the Picard–Lindelöf theorem are satisfied, the above local existence and uniqueness can be extended globally. The idea is to apply the (local) Picard–Lindelöf theorem to (2) to show that a local solution exists and is unique in the time interval $[t_{-1}, t_1]$, with $t_0 \in [t_{-1}, t_1]$. By applying again the Picard–Lindelöf theorem to two IVPs (1) associated to the initial conditions $y(t_1) = y_1$ and $y(t_{-1}) = y_{-1}$, where y_1 and y_{-1} are the values of the local solution of (2) at times t_1 and t_{-1} , respectively, we obtain local solutions for these two IVPs. Since those solutions are unique, they must be equal to the previously obtained solution on $[t_{-1}, t_1]$. Thus by “gluing” those solutions with the previous one, we can get a local solution in a new time interval $[t_{-2}, t_2]$, with $t_{-2} < t_{-1}$ and $t_1 < t_2$. By continuing this procedure we get a solution which is defined in a time interval $(t_{-\infty}, t_{\infty})$. It can be shown that this

interval is maximal in the following sense: either t_∞ is $+\infty$ ($t_{-\infty}$ is $-\infty$) or one of the following two cases occurs:

- the solution explodes in finite time;
- the solution leaves the domain of f .

More formally, the following result is obtained (see e.g. [Har82, pp. 12-13]).

Proposition 1 (*Maximal Interval of Existence*) *Let $f : E \rightarrow \mathbb{R}^n$ be continuous on E , where $E \subseteq \mathbb{R}^{n+1}$ is an open set, and let y be some solution of the IVP $y' = f(t, y), y(t_0) = y_0$ in a nonempty interval. Then y can be extended (as a solution of the IVP) over a maximal interval of existence of the form (α, β) , where $\alpha \in [-\infty, \infty)$ and $\beta \in (-\infty, \infty]$. Moreover, $y(t)$ leaves every compact subset of E , i.e. $y(t)$ tends to the boundary ∂E of E , as $t \rightarrow \alpha$ from above and as $t \rightarrow \beta$ from below.*

In the following sections, we will discuss the computability and computational complexity of solving an IVP (2) from local as well as global perspectives. But first some basic notions concerning computability with real numbers are in order.

2 Computable analysis

In this paper, we consider computability questions involving real numbers, which require computations with infinite data. Computable analysis provides a proper setting for studying such problems. In computable analysis, real numbers are represented as infinite sequences of finite objects and computations are performed only on a portion of this input; for example, a real number can be represented by a Cauchy sequence of rational numbers [BC06], [BHW08], [Wei00]. A key feature of this approach to computing with infinite data is that the result can be computed with any guaranteed accuracy, usually by providing more computation time to read more information from the input in exchange of augmented (rigorous) accuracy of results. Computable analysis is gaining an increasing amount of interest due to several factors: (i) it allows computations with more general mathematical objects, such as real numbers, open or closed subsets of \mathbb{R}^n , and functions defined on subsets of \mathbb{R}^n or over more general spaces (function spaces, metric spaces, etc.); (ii) it makes possible to use many ideas and notions from theoretical computer science developed originally for discrete objects, including the study of non-computable problems over \mathbb{R}^n , the classification of problems to different complexity classes, etc.; (iii) it reflects the limitation that computers can only handle a finite number of bits at a time. These features of computable analysis are well suited for our purposes. In the following, we present some basic definitions. More details can be found in [BC06], [BHW08], [Wei00] and references therein.

Informally, an object is computable within the computable analysis framework if it can be approximated by computer-generated approximations with an arbitrarily high precision. To formalize this notion, one needs to encode infinite objects, such as real numbers, as infinite sequences of finite objects, called representations. In the case of real numbers, this can be done with sequences

of rationals converging rapidly to a given real number. This notion can be extended to more general objects like open/closed sets in \mathbb{R}^n , etc., by using representations (see [Wei00] for a complete development).

In general, a represented space is a pair $(X; \delta)$ where X is a set, $\text{dom}(\delta) \subseteq \Sigma^{\mathbb{N}}$, and $\delta : \subseteq \Sigma^{\mathbb{N}} \rightarrow X$ is an onto map (“ $\subseteq \Sigma^{\mathbb{N}}$ ” is used to indicate that the domain of δ may be a subset of $\Sigma^{\mathbb{N}}$). Every $q \in \text{dom}(\delta)$ such that $\delta(q) = x$ is called a δ -name of x (or a name of x when δ is clear from context). For example, a possible representation of a real number is the encoding of a sequence of rationals converging rapidly to it (see more details below). An element $x \in X$ is said to be computable if it has a computable name in $\Sigma^{\mathbb{N}}$ (the notion of computability on $\Sigma^{\mathbb{N}}$ is well established). A map $\Phi : (X; \delta_X) \rightarrow (Y; \delta_Y)$ between two represented spaces is computable if there is a computable map $\phi : \subseteq \Sigma^{\mathbb{N}} \rightarrow \Sigma^{\mathbb{N}}$ such that $\Phi \circ \delta_X = \delta_Y \circ \phi$. Informally speaking, this means that there is a computer program that outputs a name of $\Phi(x)$ when given a name of x as input [BHW08].

Since we are interested in computing the operator Φ which maps (f, t_0, y_0) to a solution y of (2), we need to have representations of real numbers (for the inputs t_0, y_0) and of functions (for the input f and the output y). Here we use the following representations for points in \mathbb{R}^n and for continuous functions defined on $I_1 \times I_2 \times \dots \times I_n \subset \mathbb{R}^n$, where the I_j 's are intervals:

- (1) For a point $x \in \mathbb{R}^n$, a name of x is a sequence $\{r_k\}$ of points with rational coordinates satisfying $|x - r_k| < 2^{-k}$. Thus x is computable if there is a Turing machine (or a computer program or an algorithm) that outputs a rational n -tuple r_k on input k such that $|r_k - x| < 2^{-k}$; for a sequence $\{x_j\}$, $x_j \in \mathbb{R}^n$, a name of $\{x_j\}$ is a double sequence $\{r_{j,k}\}$ of points with rational coordinates satisfying $|x_j - r_{j,k}| < 2^{-k}$.
- (2) For every continuous function f defined on $I_1 \times I_2 \times \dots \times I_n \subseteq \mathbb{R}^n$, where I_j is an interval with endpoints a_j and b_j , a name of f is a double sequence $\{P_{k,l}\}$ of polynomials with rational coefficients satisfying $d_k(P_{k,l}, f) < 2^{-l}$, where $d_k(g, f) = \max\{|g(x) - f(x)| : a_j + 2^{-k} \leq x_j \leq b_j - 2^{-k}, 1 \leq j \leq n\}$ ($d_k(g, f) = 0$ if $[a_j + 2^{-k}, b_j - 2^{-k}] = \emptyset$). Thus, f is computable if there is an (oracle) Turing machine that outputs $P_{k,l}$ (more precisely coefficients of $P_{k,l}$) on input k, l satisfying $d_k(P_{k,l}, f) < 2^{-l}$.
- (3) For every C^m function f defined on $E = I_1 \times I_2 \times \dots \times I_n \subseteq \mathbb{R}^n$, where I_j is an interval with endpoints a_j and b_j , a (C^m) name of f is a double sequence $\{P_{k,l}\}$ of polynomials with rational coefficients satisfying

$$d_k^m(P_{k,l}, f) < 2^{-l},$$

where

$$d_k^m(g, f) = \max_{0 \leq i \leq m} \max\{|D^i g(x) - D^i f(x)| : a_j + 2^{-k} \leq x_j \leq b_j - 2^{-k}\}$$

($d_k^m(g, f) = 0$ if $[a_j + 2^{-k}, b_j - 2^{-k}] = \emptyset$). We observe that a C^m name of f contains information on both f and $Df, D^2f, \dots, D^m f$, in the sense that (P_1, P_2, \dots) is a name of f while $(D^i P_1, D^i P_2, \dots)$ is a name of $D^i f$. See [ZW03] for further details.

Because we also want to characterize the computability of the maximal interval of existence of (2), we need to have computability notions involving sets.

Informally, a planar computable open set can be visualized on a computer screen with an arbitrarily high magnification.

- (4) For an open subset A of \mathbb{R}^n , a name of A consists of a pair of an inner-name and of an outer-name; an inner-name is a sequence of balls $B(a_n, r_n) = \{x \in \mathbb{R}^n : d(a_n, x) < r_n\}$, $a_n \in \mathbb{Q}^n$ and $r_n \in \mathbb{Q}$, exhausting A , i.e., $A = \bigcup_{n=1}^{\infty} B(a_n, r_n)$; an outer-name is a sequence dense in $\mathbb{R}^n - A$. A is said to be r.e. if the sequences $\{a_n\}$ and $\{r_n\}$ are computable; co-r.e. if the sequence (dense in A) is computable; and computable if it is r.e. and co-r.e..

We close this section by noting that computable functions are always continuous (see e.g. [BHW08]).

3 Computability of the solutions of ordinary differential equations

In this section, we analyze the computability of ODEs in several settings. First we note that if f in (2) is C^1 on a compact set $E = I_1 \times I_2 \times \cdots \times I_n \subset \mathbb{R}^n$, where $I_j \subseteq \mathbb{R}$ is a closed interval, then it is Lipschitz continuous there. This fact combined with the constructive nature of the Picard–Lindelöf theorem shows that there is a computable operator which locally computes the solution of an IVP (2) defined by a C^1 function f . The computability of this operator is uniform on f and on the initial data (t_0, y_0) . In particular, if f is computable, then (2) admits a local computable solution. A theorem by Osgood [Bir73] shows that, in the case of an IVP defined over \mathbb{R}^2 , if f is computable, then so is its (local) solution (no C^1 assumption is needed). The idea behind this results is that it is possible to construct two sequences of functions, one converges from above to a solution of (2), while the other converges from below to a solution of (2). Since the solution is assumed to be unique, it follows that both sequences must converge to the solution of (2), thus ensuring its computability.

We notice that if the uniqueness requirement is dropped from (2), then it may happen that (2) has no computable solutions. This was shown in [PER79] and further improved in [Ko83]. In particular, there is a polynomial-time computable function $f : [0, 1] \times [-1, 1] \rightarrow \mathbb{R}$ such that the IVP $y' = f(t, y)$, $y(0) = 0$ does not have a computable solution on $[0, \delta]$ for any $\delta > 0$. Note that the latter IVP must have several solutions, since computability of f implies continuity of f and therefore at least a solution of the IVP must exist by Peano’s theorem. This solution cannot be unique because uniqueness would imply computability of the solution.

Next we consider the computability of IVPs (2) when f is defined on an open subset of \mathbb{R}^n . Are we able to compute the solution of (2) over its maximal interval of existence? The situation here is more complicated compared to the compact case, because the Picard–Lindelöf theorem can no longer be applied directly to the problem. Although, as described at the end of Section 1, it is possible to construct a global solution of (2) from infinitely many local solutions. But one may encounter a problem here - there may not have a master program to compute all local solutions despite the fact that each local solution is computable; in other words, the computations of local solutions are non-uniform

and different algorithms are needed for different local solutions. Nevertheless, since the input function f is the same and the initial point for each new local solution is computable from the previous local solutions, we have almost all the ingredients needed to uniformly compute all local solutions and thus to compute the global solution of (2). The remaining problem is that, to ensure the proper convergence of Picard's method to a local solution, we need a (local) Lipschitz constant; but there is no guarantee that such Lipschitz constants exist if f is merely continuous. However, if f is C^1 , then it is locally Lipschitz and its local Lipschitz constants are computable from its derivative. Hence, we can compute the solution of (2) globally from a C^1 -name of f . This is shown in [GZB09].

Theorem 2 *Assume E is a r.e. open subset of \mathbb{R}^n . Consider the initial-value problem (2) where f is C^1 on E . Let (α, β) be the maximal interval of existence of the solution $y(\cdot)$ of (2) on E . Then:*

1. *The operator $(f, t_0, y_0) \mapsto y(\cdot)$ is computable;*
2. *The operator $(f, t_0, y_0) \mapsto (\alpha, \beta)$ is semi-computable (i.e. an inner name of (α, β) can be computed from f, t_0, y_0).*

In [GZB09] it was also proved that the maximal interval can be non-computable.

Theorem 3 *There is an analytic and computable function $f : \mathbb{R} \rightarrow \mathbb{R}$ such that the unique solution of the problem*

$$\begin{cases} y' = f(y) \\ y(0) = 0 \end{cases} \quad (4)$$

is defined on a non-computable maximal interval of existence.

Since the function f in the above theorem is analytic, we conclude that the computability of the maximal interval (α, β) of existence is not tied up to the smoothness of f . The function f is constructed by creating a computable odd and bijective function $\varphi : (-\alpha, \alpha) \rightarrow \mathbb{R}$, where α is a non-computable real number, such that φ satisfies the following conditions: (i) φ is the solution of (4) for an analytic and computable function f ; and (ii) $\varphi(x) \rightarrow \pm\infty$ as $x \rightarrow \pm\alpha^\mp$.

The previous result can be generalized in the following manner. Let $E = \mathbb{R}^n$ and suppose that f is defined on \mathbb{R}^n . If a solution $y(t)$ of (2) is defined for all $t \geq t_0$, it is called a (positively) global solution that does not blow up in finite time. In general, it is difficult to predict whether or not a solution will blow up in finite time from a given initial condition (t_0, y_0) , since it can be proved that this problem is undecidable, even if $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ has only polynomials as components [GBC07]. Nevertheless, certain computational insights on the blow up problem are obtainable from f , as the following theorem in [RWZ09] suggests.

Theorem 4 *Consider the IVP (2), where f is locally Lipschitz. Let \mathcal{Z} be the set of all initial values (t_0, y_0) for which the corresponding unique solution of (2) is (positively) global. Then \mathcal{Z} is a G_δ -set (i.e. a countable intersection of open sets) and there is a computable operator determining \mathcal{Z} from f .*

We note that in Theorem 3 we have assumed f is C^1 over its open domain $E \subseteq \mathbb{R}^n$. However, the requirement for f to be C^1 is nevertheless not necessary for ensuring the existence of a unique solution of (2). It is then natural to ask whether it is possible to compute the unique solution of the IVP (2) under more general conditions. A possibility is to consider continuous functions f and to assume uniqueness of solutions of (2). This problem is studied in [CG09] and the following result, which strengthens Theorem 2, is presented there.

Theorem 5 *Consider the initial value problem (2) where f is continuous on the open set E . Suppose that (2) has a unique solution $y(\cdot)$ on E , defined on the maximal interval of existence (α, β) . Then*

1. *The operator $(f, t_0, y_0) \mapsto y(\cdot)$ is computable;*
2. *The operator $(f, t_0, y_0) \mapsto (\alpha, \beta)$ is semi-computable.*

In particular, if f is a computable function and t_0, y_0 are computable points, then (α, β) is a recursively enumerable open set and the solution $y(\cdot)$ is a computable function.

The proof of this theorem uses a quite different approach; the idea underlying the approach is to try to cover the solution with rational boxes and to test the following conditions, in an algorithmic way: (i) whether a given set of rational boxes is an actual covering of the solution of (2), and (ii) whether the “diameter of the covering” is sufficiently small so that the rational boxes provide an approximation of the solution with the desired accuracy, whenever (i) is satisfied. We can enumerate all possible families of rational boxes and apply the tests (i) and (ii) to each family generated, in a computable manner, therefore obtaining better and better approximations of the unique solution of (2), and thus proving that the solution is computable. This procedure can also be applied to study the computability of differential inclusions (see [CG09] for more details).

Due to the limited length of this short survey, we will only briefly discuss several results concerning computational complexity for solving ODEs. Let us begin by considering the case where f is defined on a compact set. In this case, it has been shown that the solution of (2), while computable, can have arbitrarily high complexity. The following theorem can be found in [Ko91, p. 219].

Theorem 6 *Let a be an arbitrary computable real number in $[0, 1]$. Then there is a polynomial-time computable function f defined on $[0, 1] \times [-1, 1]$ such that $y(t) = at^2$ is the unique solution of the initial-value problem defined by*

$$y' = f(t, y) \text{ and } y(0) = 0. \quad (5)$$

One solution to this problem is to ensure that f is Lipschitz continuous. In this case, there are various techniques available for computing the solution of (2); for example, Euler’s method. The theorem below was proved in [Ko91, p. 221]) by a careful analysis of the computational resources needed when Euler’s method is applied.

Theorem 7 *Let $f : [0, 1] \times B(0, 1) \rightarrow \mathbb{R}^n$, with $B(0, 1) = \{x \in \mathbb{R}^n : \|x\| \leq 1\} \subseteq \mathbb{R}^n$, be a polynomial-time computable Lipschitz function and assume that $y : [0, 1] \rightarrow B(0, 1)$ is the solution of (5). Then y is polynomial-space computable.*

It was shown in [Kaw10, Corollary 3.3] that this bound is sharp in the sense that there is a polynomial-time computable Lipschitz function f such that the solution of (5) is PSPACE-hard. Note that the above results are non-uniform (they are only valid when f is polynomial-time computable). A uniform version of this result was proved in [KC12, Theorem 4.10].

Another related question is whether the smoothness of f helps to reduce the computational complexity of solving an IVP (5). This problem is analyzed in [KORZ14] and PSPACE-hardness results are also shown for C^1 functions. There it is also shown that if f is more than once differentiable, then the unique solution can be CH -hard, where $CH \subseteq PSPACE$ is the counting hierarchy.

When f is analytic, the solution of (5) is polynomial-time computable on a compact set. This follows from results of Müller [Mül87] and Ko and Friedman [KF88]. As remarked in [KORZ14, last paragraph of Section 5.2], these results also show uniform polynomial computability of the IVP solving operator for analytic functions.

When we consider (5) and (2) for functions f defined over an open domain $E = I_1 \times I_2 \times \dots \times I_n \subset \mathbb{R}^n$, where $I_j \subseteq \mathbb{R}$ is an open interval, the situation is more complicated and very few results exist for that case. The problem is that the above complexity results rely on the existence of a Lipschitz constant and depend on its value. In a compact set, the value of the Lipschitz constant is fixed, and thus its value can be ignored. However, in the non-compact case we have to use local Lipschitz constants. Since the value of local Lipschitz constants can vary in non-trivial ways (it depends on the solution we are trying to compute), affecting hugely the local complexity, it is very hard to obtain a global complexity result. A possible way to solve this problem is to use a bound on the growth of the solution y of (5) as a parameter on the function used to measure the complexity, since the problem of knowing how quickly y can grow is not generally well-understood, even when f is constituted by polynomials. This approach is taken, for instance, in [BGP12], [PG16] for the case of polynomial IVPs and in [BGP11] for the case of analytic IVPs. In particular, in [PG16] it is shown that the complexity of solving a polynomial IVP over its maximal interval of definition is polynomial in the length of the solution curve y . The idea of using the length of the solution curve solves several problems and provides a robust notion of complexity for the non-compact case. See [PG16] for more details.

Finally, we note that there are several qualitative results about ODEs. In general, the problem of determining the long-term behavior of a system defined with an ODE is a very complicated one. For that reason, qualitative results have been obtained in dynamical systems theory and computable version of several of these results exist. For example, in [GZB12] a computable version of the stable manifold is given, while [GZD12] provides a computable version of the Hartman-Grobman theorem. It is also shown in [GZ15] that the domain of attraction of an hyperbolic equilibrium point x_0 (i.e. a zero of f) is in general non computable, i.e. one cannot decide whether the trajectory starting from some point will ultimately converge to x_0 .

4 Acknowledgments

Daniel Graça was partially supported by *Fundação para a Ciência e a Tecnologia* and EU FEDER POCTI/POCI via SQIG - Instituto de Telecomunicações through the FCT project UID/EEA/50008/2013.



This project has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 731143.

EU disclaimer: the results presented in this paper reflect only the author's view and the EU is not responsible for any use that may be made of the information it contains.

References

- [BC06] M. Braverman and S. Cook. Computing over the reals: foundations for scientific computing. *Notices of the American Mathematical Society*, 53(3):318–329, 2006.
- [BGP11] O. Bournez, D. S. Graça, and A. Pouly. Solving analytic differential equations in polynomial time over unbounded domains. In F. Murlak and P. Sankowski, editors, *Proc. 36th International Symposium on Mathematical Foundations of Computer Science (MFCS 2011)*, volume 6907 of *Lecture Notes in Computer Science ARCoSS (Advanced Research in Computing and Software Science)*, pages 170–181. Springer, 2011.
- [BGP12] O. Bournez, D. S. Graça, and A. Pouly. On the complexity of solving initial value problems. In *Proc. 37th International Symposium on Symbolic and Algebraic Computation (ISSAC 2012)*, volume abs/1202.4407, 2012.
- [BHW08] V. Brattka, P. Hertling, and K. Weihrauch. A tutorial on computable analysis. In S. B. Cooper, B. Löwe, and A. Sorbi, editors, *New Computational Paradigms: Changing Conceptions of What is Computable*, pages 425–491. Springer, 2008.
- [Bir73] G. Birkhoff, editor. *A Source Book in Classical Analysis*, chapter Osgood's Existence Theorem, pages 251–258. Harvard University Press, 1973.
- [BR89] G. Birkhoff and G.-C. Rota. *Ordinary Differential Equations*. John Wiley & Sons, 4th edition, 1989.
- [CG09] P. Collins and D. S. Graça. Effective computability of solutions of differential inclusions — the ten thousand monkeys approach. *Journal of Universal Computer Science*, 15(6):1162–1185, 2009.
- [GBC07] D. S. Graça, J. Buescu, and M. L. Campagnolo. Boundedness of the domain of definition is undecidable for polynomial ODEs. In R. Dillhage, T. Grubba, A. Sorbi, K. Weihrauch, and N. Zhong,

editors, *Proc. 4th International Conference on Computability and Complexity in Analysis (CCA 2007)*, volume 202 of *Electronic Notes in Theoretical Computer Science*, pages 49–57. Elsevier, 2007.

- [GZ15] D. S. Graça and N. Zhong. An analytic system with a computable hyperbolic sink whose basin of attraction is non-computable. *Theory of Computing Systems*, 57:478–520, 2015.
- [GZB09] D. S. Graça, N. Zhong, and J. Buescu. Computability, noncomputability and undecidability of maximal intervals of IVPs. *Transactions of the American Mathematical Society*, 361(6):2913–2927, 2009.
- [GZB12] D. S. Graça, N. Zhong, and J. Buescu. Computability, noncomputability, and hyperbolic systems. *Applied Mathematics and Computation*, 219(6):3039–3054, 2012.
- [GZD12] D. S. Graça, N. Zhong, and H. S. Dumas. The connection between computability of a nonlinear problem and its linearization: the hartman-grobman theorem revisited. *Theoretical Computer Science*, 457(26):101–110, 2012.
- [Har82] P. Hartman. *Ordinary Differential Equations*. Birkhäuser, 2nd edition, 1982.
- [Kaw10] A. Kawamura. Lipschitz continuous ordinary differential equations are polynomial-space complete. *Computational Complexity*, 19(2):305–332, 2010.
- [KC12] A. Kawamura and S. Cook. Complexity theory for operators in analysis. *ACM Transactions on Computation Theory*, 42(2):Article No. 5, 2012.
- [KF88] K.-I Ko and H. Friedman. Computing power series in polynomial time. *Advances in Applied Mathematics*, 9(1):40–50, 1988.
- [Ko83] K.-I Ko. On the computational complexity of ordinary differential equations. *Information and control*, 58:157–194, 1983.
- [Ko91] K.-I Ko. *Complexity Theory of Real Functions*. Birkhäuser, 1991.
- [KORZ14] A. Kawamura, H. Ota, C. Rösnick, and M. Ziegler. Computational complexity of smooth differential equations. *Logical Methods in Computer Science*, 10(1:6):1–15, 2014.
- [Mül87] N. Th. Müller. Uniform computational complexity of taylor series. In T. Ottmann, editor, *Proc. 14th International Colloquium on Automata, Languages and Programming (ICALP 1987)*, volume 267 of *Lecture Notes in Computer Science*, pages 435–444. Springer, 1987.
- [PER79] M. B. Pour-El and J. I. Richards. A computable ordinary differential equation which possesses no computable solution. *Annals of Mathematical Logic*, 17:61–90, 1979.

- [PG16] Amaury Pouly and Daniel S. Graça. Computational complexity of solving polynomial differential equations over unbounded domains. *Theoretical Computer Science*, 626(2):67–82, 2016.
- [RWZ09] R. Rettinger, K. Weihrauch, and N. Zhong. Topological complexity of blowup problems. *Journal of Universal Computer Science*, 15(6):1301–1316, 2009.
- [Wei00] K. Weihrauch. *Computable Analysis: an Introduction*. Springer, 2000.
- [ZW03] N. Zhong and K. Weihrauch. Computability theory of generalized functions. *Journal of the ACM*, 50(4):469–505, 2003.