Chapter 3
Computability of Differential Equations

Daniel S. Graça and Ning Zhong

Abstract In this chapter, we provide a survey of results concerning the computability and computational complexity of differential equations. In particular, we study the conditions which ensure computability of the solution to an initial value problem for an ordinary differential equation (ODE) and analyze the computational complexity of a computable solution. We also present computability results concerning the asymptotic behaviors of ODEs as well as several classically important partial differential equations.

3.1 Introduction

A differential equation is an equation that relates a function with its (partial) derivatives of various orders. Differential equations are widely used in many fields and have been extensively studied since the time when the calculus was invented. Given a differential equation, the goal is to find its solutions. Unfortunately, most differential equations do not possess solutions in explicit forms. In order to obtain useful information about the solutions without having an explicit representation formula for them, various methods and tools have been developed. Classically, the following methods are of most significance:

- *Numerical methods*: where solutions are approximated numerically.
- *Qualitative methods*: where the behavior of a differential equation and of its solutions is analyzed from a qualitative perspective.

Daniel S. Graça

Universidade do Algarve, Faro, Portugal and Instituto de Telecomunicações, Portugal
e-mail: dgraca@ualg.pt

Ning Zhong

Department of Mathematical Sciences, University of Cincinnati, Cincinnati, Ohio, USA
e-mail: ning.zhong@uc.edu

Numerical methods generate numerical algorithms which calculate approximations to the exact solution that cannot be found explicitly, and the approximations are expected to converge to the exact solution. The convergence is a fundamental property for a numerical algorithm to be of any use. In numerical analysis, the accuracy of approximation (the convergence rate) is usually not specified – sometimes it is impossible to specify (see [81] for an example) – as an input to the algorithm that computes the approximations, but rather it is provided externally and implicitly by finding certain analytical forms of approximation errors, which frequently depend on additional assumptions; for example, the twice differentiability of the exact solution. This requirement is often crucial, yet not applicable to many initial value problems for the first-order ODEs whose solutions may not be C^2 but nevertheless are approximable by convergent numerical solutions.

The qualitative method is another key tool, often complementing numerical methods, in the study of ordinary differential equations; in particular, in describing the asymptotic behaviors of solutions of ODEs. For example, it is used to describe the limit set of a typical solution as a spatial object as time tends to infinity.

In addition to the classical methods, there is also a relatively new approach based on the Turing machine model over the real field (see [67, 56, 87] and references therein):

- *Computable analysis methods*: where computability and computational complexity of problems related to differential equations are studied.

In computable analysis, a problem is called computable if it can in principle be solved or approximated with arbitrary precision by a computing device. The computable analysis method provides useful insights on many problems related to a given ODE from the perspective of computation in the sense that (1) the method can be used to identify whether a problem is computable; (2) if a problem is proved to be computable using the computable analysis method, often the proof generates a (Turing) algorithm that computes approximations with arbitrary precision and, moreover, with the accuracy specified as a part of the input (computability); and (3) the method has a mechanism for assessing the resources needed for a computation (computational complexity).

The computable analysis method has been used by many authors to explore the computational aspects of differential equations; in particular, to address questions related to computing solutions of differential equations. In this chapter, we present a brief survey of some existing results.

The chapter is organized as follows. Section 3.2 addresses computability of solutions of first-order ODEs, while Section 3.3 analyzes the computational complexity of the computable solutions. Section 3.4 discusses computability of asymptotic behaviors of ODEs. Finally, Section 3.5 presents some computability results of several classically important partial differential equations.

3.2 Computability of the Solutions of Ordinary Differential Equations

In this section, we consider ordinary differential equations (ODEs) of the form

$$y' = f(y) \tag{3.1}$$

and associated initial value problems (IVPs)

$$\begin{cases} y' = f(y) \\ y(t_0) = y_0 \end{cases} \tag{3.2}$$

where $f : E \to \mathbb{R}^n$ is continuous over $E \subseteq \mathbb{R}^n$. Since an ODE of the more general form $y' = g(t, y)$ can be reduced to (3.1) if one replaces the independent variable t by a new variable y_{n+1} satisfying $y'_{n+1} = 1$, $y_{n+1}(t_0) = t_0$, there is no loss of generality in studying only ODEs in the form of (3.1).

3.2.1 Computability over Compact Sets

Assume that E is a compact subset of $\mathbb{R}^n$. Classically, it is well known that continuity of f ensures the existence of solution(s) to the IVP (3.2) (Peano's existence theorem, see, e.g., [11, pp. 191-193]), and if f is Lipschitz continuous then the solution is unique and C^1 smooth (Picard-Lindelöf theorem, see, e.g., [43, pp. 8-10]). The compactness of E is essential for the proof of Peano's existence theorem, in which a sequence of functions is constructed and then the Arzelà-Ascoli theorem is employed to obtain a subsequence that converges toward a solution of (3.2). The proof is non-constructive. On the other hand, the classical proof of existence and uniqueness for Lipschitz continuous f is constructive, such as Picard's scheme of successive approximations; such constructive schemes lead naturally to some computer programs for computing the unique solution, uniformly in f. Recall that f is said to be Lipschitz continuous if it satisfies a *Lipschitz condition* on E; i.e., there exists some positive constant K (a *Lipschitz constant*) such that, for all $x, y \in E$

$$\|f(x) - f(y)\| \le K \|x - y\|.$$

Thus the question remaining to be investigated is whether solutions are still computable when the vector field f is continuous on E but fails to be Lipschitz continuous. In this case, whether the IVP (3.2) has a unique solution plays a pivotal role in determining whether (3.2) admits a computable solution. A theorem by Osgood [10] shows implicitly that the initial value problem $y' = f(t, y)$, $y(0) = y_0$ does admit a computable solution on $[0, \delta]$ for some $\delta > 0$, provided that f is computable on $E = [0, 1] \times [-1, 1]$ and the solution of the IVP is unique. The idea used for the proof is to compute sequences of functions which converge to a maximal solution y_{max} from above and a minimal solution y_{min} from below. If the solution is unique,

then the convergence is effective and thus results in a computable solution. However, this technique only works in the plane, since in higher dimensions there is no similar notion of a maximal or a minimal solution. In [71] Ruohonen generalizes this result to higher dimensions, using approximation funnels to compute the solution of (3.2), obtaining the following result.

Theorem 3.2.1. *Suppose that f is computable and not identically zero on the closed ball $B(y_0, b) = \{x \in \mathbb{R}^n : |y_0 - x| \leq b\}$, where $b > 0, t_0, y_0$ are computable. Then the solution of (3.2) is computable on $[t_0, t_0 + \delta]$ for $\delta = \frac{b}{max_{x \in E} \|f(x)\|} > 0$, provided that the solution of (3.2) is unique on that interval.*

Another result that relies on Picard's construction to compute the solution of (3.2) is presented in [28]. There a domain-theoretic version of Picard's operator is presented to compute the solution of the IVP (3.2), provided that the function defining the IVP is Lipschitz continuous.

It is also shown in [19] that, under certain conditions, solutions of the class of implicit ODEs in the form of $A(y) \cdot y' = f(y, \lambda)$ are computable, where $A(y)$ is an $n \times n$ matrix formed by polynomials in $y_1, \ldots, y_n$, with $y = (y_1, \ldots, y_n)$, and λ is a parameter.

The scenario changes completely if (3.2) admits multiple solutions. In this case, it is possible that none of the solutions is computable. The following result was first proved by Aberth [2, p. 152] for the case where f is assumed just to be computable; later a different proof was presented by Pour-El and Richards [65]; then the result was extended in [55] to include the case where f is polynomial-time computable.

Theorem 3.2.2. *There exists a polynomial-time computable function $f : [0, 1] \times [-1, 1] \to \mathbb{R}$ such that*

$$\begin{cases} y' = f(t, y) \\ y(0) = 0 \end{cases} \tag{3.3}$$

does not have a computable solution y on $[0, \delta]$ for any $\delta > 0$.

The idea for the proof is to construct a function f that embeds a non-computable problem $\mathscr{P}$ and show that if a computable solution for (3.3) exists, then $\mathscr{P}$ can be solved computationally, a contradiction. For details of the proof, we refer the reader to the original papers [65, 55] or to [56, pp. 216-219].

3.2.2 Computability over Non-compact Sets

In this subsection, we assume that E is an open subset of $\mathbb{R}^n$. The Peano existence theorem indicates that the IVP (3.2) has a solution existing on some time interval containing t_0 in its interior, provided that f is continuous on E. An interval is called an interval of existence if it contains t_0 and if a solution of (3.2) exists on it. The following classical result indicates how large an interval of existence may be (see, e.g., [43, pp. 12-13]).

Proposition 3.2.3. *(Maximal Interval of Existence) Let $f : E \to \mathbb{R}^n$ be continuous on E. Then the IVP (3.2) has a maximal interval of existence, and it is of the form (α, β), with $\alpha \in [-\infty, \infty)$ and $\beta \in (-\infty, \infty]$. There is a solution $y(t)$ of (3.2) on (α, β) and $y(t)$ leaves every compact subset of E, i.e., $y(t)$ tends to the boundary ∂E of E, as $t \to \alpha$ from above and as $t \to \beta$ from below.*

The proposition is a simple classical existence statement "the IVP (3.2) has a maximal interval of existence and a solution exists on it." It turns out that this simple statement gives rise to a variety of questions about computability. Is the maximal interval (α, β) of existence computable from the data (f, t_0, y_0) that defines the IVP (3.2)? Does (3.2) admit a computable solution on (α, β)? If $E = \mathbb{R}^n$, then a solution of (3.2) is said to blow up in finite time if α and β are two real numbers. Is the problem whether (3.2) admits a finite blowup solution decidable? We will discuss these questions in this subsection.

It is known from Subsection 3.2.1 that if the IVP (3.2) has multiple solutions then it may occur that none of the solutions is computable. Hence, in this subsection, we consider only those IVPs to which the solution is unique.

We begin by examining the proof of the proposition. The proposition can be proved by using the Peano existence theorem repeatedly: The theorem first produces a solution $y(t)$ of the IVP (3.2) on an interval $[a_0, b_0]$ containing t_0; then it extends $y(t)$ to $[a_1, a_0]$ and to $[b_0, b_1]$, $a_1 < a_0$ and $b_1 > b_0$, by yielding a solution to the IVP $v' = f(v)$, $v(b_0) = y(b_0)$, and to the IVP $v' = f(v)$, $v(a_0) = y(a_0)$; and the process is repeated infinitely many times to reach the maximality: $(\alpha, \beta) = \bigcup_{j+1}^{\infty} [a_j, b_j]$. Thus, if the IVP (3.2) has a unique solution $y(t)$ over (α, β), then it follows from Theorem 3.2.1 that a_j, b_j, and $y(t)$ on interval $[a_j, b_j]$ are computable from j, f, t_0, and y_0. But the argument is insufficient to show that $y(t)$ is computable on (α, β) because it is possible that there does not exist a "master" algorithm computing $y(t)$, independent of j. One possible way of dealing with this shortcoming is to strengthen f to be a smoother function, which is actually an approach frequently used in classical analysis of ODEs: the smoother the vector field f is, the better, possibly, the solution y behaves. This is the approach used in [36]. By strengthening f to be a C^1 function on E, the following result is shown in [36].

Theorem 3.2.4. *Assume that E is an r.e. open subset of $\mathbb{R}^n$. Consider the initial value problem (3.2) where f is C^1 on E. Let (α, β) be the maximal interval of existence of the solution $y(\cdot)$ of (3.2) 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., α can be computed from above and β can be computed from below).*

Recall that E is an r.e. open subset of $\mathbb{R}^n$ if it can be filled up by a computer-generated sequence $\{B_j\}_{j=1}^{\infty}$ of rational balls (balls having rational centers and rational radii), i.e., $E = \bigcup_{j=1}^{\infty} B_j$. The main role played by the C^1 smoothness in the proof is that C^1 smoothness implies local Lipschitz continuity (and the converse is false); that is, if f is C^1 on E, then f is Lipschitz continuous on every compact subset of E. An algorithmic version of this fact is proved in [36]. Thus, when giving a

C^1-name of f, it becomes possible to compute a Lipschitz constant K_j for each E_j, where E_j is the closure of $\bigcup_{i=1}^{j} B_i$, then compute the solution of (3.2) on E_j, and thus compute the solution over the maximal interval (α, β) of existence, because the E_j are getting ever larger and eventually fill up E.

Although for many important ODEs their vector fields f are indeed C^1, the requirement for f being C^1 is nevertheless not necessary for ensuring the existence of a unique solution. It is then natural to ask whether it is possible to compute the unique solution of the IVP (3.2), whenever it exists, over the maximal interval (α, β); in other words, whether Theorem 3.2.4 holds true over (α, β) when E is an open subset of $\mathbb{R}^n$. The problem is studied in [26] and a positive answer is presented there. The proof uses a quite different approach; the idea underlying the approach is to try to cover the solution with rational boxes and to test (i) and (ii), in an algorithmic way: (i) whether a given set of rational boxes is an actual covering of the solution of (3.2), 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. The process of enumerating all possible families of rational boxes and applying the tests (i) and (ii) to each family generates, effectively, better and better approximations to the unique solution of (3.2), and thus proves that the solution is computable. This procedure can also be applied to study the computability of differential inclusions (see [26] for more details). The following theorem proved in [26] (see also [25]) strengthens Theorem 3.2.4.

Theorem 3.2.5. *Consider the initial value problem (3.2) where f is continuous on the open set E. Suppose that (3.2) 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 (i.e., α can be computed from above and β can be computed from below).*

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.

A question that remains is whether the maximal interval of existence is computable. The answer is negative, as shown in [36].

Theorem 3.2.6. *There is an analytic and computable function $f : \mathbb{R} \to \mathbb{R}$ such that the unique solution of the problem*

$$\begin{cases} y' = f(y) \\ y(0) = 0 \end{cases} \tag{3.4}$$

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

The theorem indicates that the computability of the maximal interval (α, β) of existence is not tied up with the smoothness of f; rather it has more to do with

the "globalness" of (α, β). Later, in Section 3.4, we will see again that a "global property" is usually difficult to compute. The counterexample is constructed by creating a computable odd and bijective function $\varphi : (-\alpha, \alpha) \to \mathbb{R}$, where α is a non-computable real number, such that φ satisfies the following conditions: (i) φ is the solution of (3.4) for an analytic and computable function f; and (ii) $\varphi(x) \to \pm\infty$ as $x \to \pm\alpha^{\mp}$.

When E is the whole space $\mathbb{R}^n$ and f is defined on $\mathbb{R}^n$, if a solution $y(t)$ of (3.2) 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 datum (t_0, y_0), as Theorem 3.2.6 already indicates, because it often requires extra knowledge on some quantitative estimates and asymptotics of the solution over a long period of time. Actually the problem of determining whether the solution of (3.2) blows up in finite time is undecidable even if $f : \mathbb{R}^n \to \mathbb{R}^n$ has only polynomials as components [39]. Nevertheless, certain computational insight on the blowup problem is obtainable from f, as the following theorem in [69] suggests.

Theorem 3.2.7. *Consider the IVP (3.2), where f is locally Lipschitz. Let $\mathscr{Z}$ be the set of all initial values (t_0, y_0) for which the corresponding unique solution of (3.2) is (positively) global. Then $\mathscr{Z}$ is a G_δ-set and there is a computable operator determining $\mathscr{Z}$ from f.*

3.3 Computational Complexity of the Solutions of Ordinary Differential Equations

In this section we analyze the computational complexity of solving an initial value problem (3.2).

We begin by noting that, as mentioned before, any C^1 function defined on a compact set $E \subseteq \mathbb{R}^n$ satisfies a Lipschitz condition. The analysis of any standard algorithm (e.g., Euler's method, or the algorithm implicit in Picard's iteration method) to solve an initial value problem (3.2) applied to a continuous function which satisfies a Lipschitz condition over a compact set, with a Lipschitz constant $K > 0$, shows that the value of the Lipschitz constant affects the number of iterations needed to compute the solution of (3.2) with a given accuracy. However, since the Lipschitz constant is constant in a compact set, it can be hidden in the big O notation. When we solve (3.2) over a non-compact set, we typically have infinitely many distinct local Lipschitz constants, and the complexity results need to take into account this effect. This is the reason the two cases are analyzed in separate sections, as for the case of computability.

3.3.1 Results for Compact Sets

We have just discussed above the importance that a Lipschitz condition seems to have when computing the solution of an initial value problem (3.2). First it makes sense to ask how the presence of a Lipschitz constant affects the computational complexity of the solution of (3.2). We already know from Theorem 3.2.1 that if (3.2) has a unique solution, then it must be computable. So what is the complexity of the solution of (3.2) if we assume that f is "easy to compute" (polynomial-time computable)? The following result from [56, p. 219], based on another result of Miller [58, p. 469], shows that in that case the solution of (3.2), while computable, can have arbitrarily high complexity.

Theorem 3.3.1. *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*

$$y' = f(t,y) \text{ and } y(0) = 0. \qquad (3.5)$$

Note that in (3.5) we have used the independent variable t explicitly, i.e., we considered a non-autonomous ODE instead of an autonomous ODE as in (3.2). Technically speaking, that is not needed as we have seen in Section 3.2, since a non-autonomous IVP can be converted into an autonomous IVP. However, in the remainder of this section we will explicitly consider the independent variable t since, as we will see later in this section, this will allow us to do a more refined analysis on what happens at a computational complexity level by analyzing how the smoothness of f relative to t or to y can affect the complexity of the solution of (3.5). In this section we also suppose that the initial condition for (3.5) is $y(0) = 0$, although WLOG any initial condition $y(t_0) = y_0$, where t_0, y_0 are polynomial-time computable, could be used to obtain the same results. We note that for non-autonomous IVPs (3.5) the Lipschitz condition is not needed for the variable t. Therefore, in that case, we say that $f : [0,1] \times B(0,1) \to \mathbb{R}^n$, with $B(0,1) \subseteq \mathbb{R}^n$, is (right-)Lipschitz if it satisfies the following condition

$$\|f(t,y_1) - f(t,y_2)\| \leq L \|y_1 - y_2\|$$

for all $t \in [0,1]$ and all $y_1, y_2 \in B(0,1) \subseteq \mathbb{R}^n$. Remark that if f in (3.5) is (right-)Lipschitz, then when we convert the ODE of the initial value problem (3.5) into an autonomous ODE (3.1), then f in (3.1) is Lipschitz continuous. Therefore, using non-autonomous ODEs with (right-)Lipschitz ODEs or autonomous Lipschitz continuous ODEs are equivalent approaches, although the non-autonomous case (3.5) has the advantage mentioned above.

In (3.5) we also assume that the function f is defined on $[0,1] \times B(0,1)$. However, those results can easily be extended to functions defined over $[0,T] \times B(0,M)$ for $T,M > 0$ via rescaling. More concretely, let $f : [0,T] \times B(0,M) \to \mathbb{R}^n$ and let $y : [0,T] \to B(0,M)$ be the solution of (3.5). Note that we suppose that $y(t)$ is always defined for all $t \in [0,T]$ (and hence is always inside $B(0,M)$). Define the function

$y_* : [0,1] \to \mathbb{R}^n$ as $y_*(t) = y(tT)$. In other words $y_*(1) = y(T)$. It is not difficult to see that y_* is the solution of the IVP

$$y'_* = Tf(tT, y_*) \text{ and } y_*(0) = 0.$$

Similarly one can define the function $y_{**} : [0,1] \to \mathbb{R}^n$ as $y_{**}(t) = y_*(t)/M = y(tT)/M$. In other words $y_{**}(t)$ is just like y_*, but scaled down by a factor of M so that $y_{**}(t) \in B(0,1)$ for all $t \in [0,1]$. It is also not difficult to check that y_{**} is the solution of the IVP

$$y'_{**} = \frac{T}{M} f(tT, My_{**}) \text{ and } y_{**}(0) = 0.$$

Note that, for a fixed compact set $[0,T] \times B(0,M)$, T and M are just constants. Therefore the complexity of $f : [0,T] \times B(0,M) \to \mathbb{R}^n$ is essentially the same as that of $f_{**} : [0,1] \times B(0,1) \to \mathbb{R}^n$ given by $f_{**}(t,y) = \frac{T}{M} f(tT, My)$ and thus the complexity of finding the solution of (3.5) when $f : [0,T] \times B(0,M) \to \mathbb{R}^n$ is essentially the same as that of finding the solution of (3.5) when $f : [0,1] \times B(0,1) \to \mathbb{R}^n$.

In [23, p. 450] it is shown that if f of (3.5) satisfies a Lipschitz condition on $[0,1] \times [-1,1]$ and f belongs to the nth level of the Grzegorczyk hierarchy $\mathscr{E}^{(n)}$, for $n \geq 3$, then so will its solution y. This result is refined for the case of polynomial-time computable functions f in [55, p. 159] (or [56, p. 221]). There it is mentioned that if f of (3.5) is polynomial-time computable and satisfies a Lipschitz condition on $[0,1] \times [-1,1]$, then the solution of (3.4) can be computed in polynomial space. This result can be extended to the case where $y : [0,1] \to B(0,1)$ where $B(0,1) \subseteq \mathbb{R}^n$ and $n \geq 1$.

Theorem 3.3.2. *Let $f : [0,1] \times B(0,1) \to \mathbb{R}^n$, with $B(0,1) \subseteq \mathbb{R}^n$, be a polynomial-time computable (right-)Lipschitz function and assume that $y : [0,1] \to B(0,1)$ is the solution of (3.5). Then y is polynomial-space computable.*

As mentioned by Ko, the proof of this result follows from a careful analysis of Euler's method. For example, it is shown in [7, pp. 349-350] that when $n = 1$ and we use Euler's method to compute the solution y of (3.4) over the time interval $[0,1]$ with time step bounded by $h > 0$ and rounding error bounded by $\rho > 0$, the error e_n in step n is bounded by

$$|e_n| \leq A\rho + B(hC + \rho/h) \tag{3.6}$$

where $A, B, C > 0$ are some constants (the original formula (6.2.32) [7] explicitly tells how A, B, C depend on the Lipschitz constant, etc., but that level of detail is not needed for our analysis). So to compute the solution of (3.4) over the time interval $[0,1]$ with accuracy 2^{-n} we could use Euler's method with $h = a2^{-n}$ and $\rho = b(2^{-n})^2$ for some appropriate constants $a, b > 0$ which are independent of n. Euler's method would then run in space polynomial (indeed quadratic) in n, which shows the result. The case $n > 1$ can be obtained by considering the Euler method for dimensions $n > 1$ as explained in [11, Section 7.2]. The analysis of the error (done in [11, Section 7.3], except that the rounding error is not considered, but this can also be taken into account using a procedure similar to that of [7, p. 350]) gives a

formula (3.6), but with constants $A, B, C > 0$ which have different values than from the previous case, and therefore the conclusions remain valid for the case where $n > 1$.

In [56, Section 7.4] (also implicitly in [55, p. 159, Question D],) Ko asks whether polynomial space is also a lower bound for the complexity of the solution y of (3.5), assuming the conditions of the previous theorem for $n = 1$. Kawamura provides an affirmative answer in [49] (see also [48]) using an appropriate notion of reduction (see [49, Section 2.2]) which allows us to define $\mathscr{C}$-hard and $\mathscr{C}$-complete functions for a complexity class $\mathscr{C}$. Namely Kawamura shows the following theorem.

Theorem 3.3.3. *There exists a polynomial-time computable function $f : [0, 1] \times [-1, 1] \to \mathbb{R}$ which satisfies a (right-)Lipschitz condition, such that the unique solution $y : [0, 1] \to \mathbb{R}$ of (3.5) takes values in $[-1, 1]$ and is PSPACE-complete.*

As a corollary [49, Corollary 3.3] one has the following result.

Corollary 3.3.4. *$P = PSPACE$ if and only if any solution $y : [0, 1] \to B(0, 1)$ of (3.5) is polynomial-time computable, where $f : [0, 1] \times B(0, 1) \to \mathbb{R}^n$, with $B(0, 1) \subseteq \mathbb{R}^n$, is a polynomial-time (right-)Lipschitz computable function.*

The previous results are non-uniform since we fix the complexity of the input to be polynomial time. But one can naturally ask what is the (uniform) complexity of the operator *LipIVP* which maps a function f satisfying a Lipschitz condition to the solution y of (3.5). A major problem to tackle is that until recently there was no general theory to measure the complexity of operators which map functions into functions. This problem was solved in [50] using appropriate representations of functions and appropriate notions of reduction, hardness and completeness. Using this setting one can define $\mathscr{C}$-hard and $\mathscr{C}$-complete functions for a complexity class $\mathscr{C}$. Since we are talking of functions, it makes sense to take $\mathscr{C} = FP$ (functions computable in polynomial time) or $\mathscr{C} = FPSPACE$ (functions computable in polynomial space). By using appropriate representations $\delta_\square$ for continuous functions in $C([0, 1])$ and $\delta_{\square L}$ for functions satisfying a Lipschitz condition (which are continuous), and an appropriate reduction $\leq^2_{mF}$ (see [50] for more details) it was also shown in [50, Theorem 4.10] that the operator *LipIVP* is FPSPACE-complete.

Theorem 3.3.5. *LipIVP is $(\delta_{\square L}, \delta_\square)$-FPSPACE-$\leq^2_{mF}$-complete.*

An interesting question is whether smoothness of f helps reduce the computational complexity of solving an IVP (3.5). We say that $f : [0, 1] \times \mathbb{R} \to \mathbb{R}$ is of class $C^{(i,j)}$ if the partial derivative $\partial^{n+m} f(t, y)/\partial t^n \partial y^m$ exists and is continuous for all $n \leq i$ and $m \leq j$; it is said to be of class $C^{(\infty,j)}$ if it is of class $C^{(i,j)}$ for all $i \in \mathbb{N}$. In [51] this question is analyzed and Theorem 3.3.3 is generalized to $C^{(\infty,1)}$ functions.

Theorem 3.3.6. *There exists a polynomial-time computable function $f : [0, 1] \times [-1, 1] \to \mathbb{R}$ of class $C^{(\infty,1)}$ such that the unique solution $y : [0, 1] \to \mathbb{R}$ of (3.5) is PSPACE-complete.*

In that paper it is also shown [51, Theorem 2] that if f is more than once differentiable, then the unique solution can be *CH*-hard, where *CH* $\subseteq$ *PSPACE* is the counting hierarchy.

Theorem 3.3.7. *Let k be a positive integer. There exists a polynomial-time computable function $f : [0,1] \times [-1,1] \to \mathbb{R}$ of class $C^{(\infty,k)}$ such that the unique solution $y : [0,1] \to \mathbb{R}$ of (3.5) is CH-hard.*

In the most extreme case, when f is analytic, the solution of (3.5) is polynomial-time computable on a compact set. This follows from results of Müller [60] and Ko and Friedman [57], which show that polynomial-time computability of an analytic function on a compact interval is equivalent to polynomial-time computability of the sequence of its Taylor coefficients at a rational point. Given a sequence of Taylor coefficients for f one can compute the Taylor coefficients for the solution y of (3.5) [61, Theorem 2.1]. Note that it is known that if f is analytic, then so is the solution y of (3.5) [5, Section 32.4]. By analytic continuation [60], polynomial-time computability of y follows.

Theorem 3.3.8. *Let $f : [0,1] \times B(0,1) \to B(0,1)$ with $B(0,1) \subseteq \mathbb{R}^n$ be a polynomial-time computable analytic function and $y : [0,1] \to B(0,1)$ be the solution of (3.5). Then y is polynomial-time computable.*

As remarked in [51, last paragraph of Section 5.2], the previous argument also shows uniform polynomial-time computability of the operator *LipIVP*, if we represent analytic functions by their sequence of Taylor coefficients, obtaining a representation δ_{Taylor}, since the Taylor sequence of y is easy to compute from the Taylor sequence of f. In the next theorem *LipIVP* $\upharpoonright^{\mathscr{D}}$ represents the operator *LipIVP* restricted to the class of (real) functions $\mathscr{D}$ and C^{ω} represents the class of analytic functions.

Theorem 3.3.9. *The operator LipIVP $\upharpoonright^{C^{\omega}}$ belongs to $(\delta_{Taylor}, \delta_{Taylor})$-FP.*

Theorem 3.3.6 can also be extended to a uniform version by showing that the operator *LipIVP* $\upharpoonright^{C^{(\infty,1)}}$ [51, last paragraph of Section 5.2] is *FPSPACE* complete, using the polynomial-time Weihrauch reduction $\leq_W$.

Theorem 3.3.10. *The operator LipIVP $\upharpoonright^{C^{(\infty,1)}}$ is $(\delta_{\square L}, \delta_{\square})$-FPSPACE-$\leq_W$-complete.*

Computability in polynomial time of the solution of (3.5) is also obtained in [29] by using polynomial enclosures in Picard's method, provided that some assumptions are made on the function f in (3.5), such as that f is Lipschitz and that its representation via polynomial enclosures satisfies certain properties (e.g., this representation should not take "too much space").

3.3.2 Results for Non-compact Sets

In the previous section we have analyzed the computational complexity of solutions of an IVP (3.5) (or, equivalently, (3.2)) where f is considered over a compact set $[0,1] \times B(0,1)$, with $B(0,1) \subseteq \mathbb{R}^n$. But what happens when $f : \mathbb{R}^{n+1} \to \mathbb{R}^n$ and the solution y of (3.5) is considered over its maximal interval of definition? That is, what is the computational complexity of $y : \mathbb{R} \to \mathbb{R}^n$, or more generally of the operator mapping f to y?

As a first attempt we could consider the approach, given $t \in [0, +\infty)$, of rescaling the IVP (3.5) given by $f : [0,t] \times B(0,M) \to \mathbb{R}^n$, where M is big enough so that $y(t) \in B(0,M)$ for all $t \in [0,t]$, to an IVP (3.5) which uses a function $\bar{f} : [0,1] \times B(0,1)$ instead of f, as explained in the previous section. Since, for a fixed M (and fixed T, with $t \leq T$), the complexity of solving (3.5) is the same as the complexity of solving (3.5) where $\bar{f}$ is used instead of f, we might be led to conclude that the complexity of obtaining the solution of (3.5) for the non-compact case where $f : \mathbb{R}^{n+1} \to \mathbb{R}^n$ is the same as when f is taken over a compact set $[0,1] \times B(0,1)$. However this reasoning is incorrect, since M (and T) is not fixed in this case and must be uniformly computed from y, i.e., $M = M(y(t)) = M(t)$. This is not a trivial task since to compute $y(t)$, we need to know $M(t)$. On the other side, to know $M(t)$, we need to know $\max_{0 \leq u \leq t} \|y(u)\|$ and therefore we get into a circular argument where $M(t)$ is needed to compute $y(t)$ and vice versa. To better illustrate this problem, consider the following system taken from [64]

$$\begin{cases} y_1(0) = 1 \\ y_2(0) = 1 \\ \ldots \\ y_d(0) = 1 \end{cases} \quad \begin{cases} y_1'(t) = y_1(t) \\ y_2'(t) = y_1(t)y_2(t) \\ \ldots \\ y_d'(t) = y_1(t) \cdots y_d(t) \end{cases}$$

The results of Section 3.3.1 show that for any fixed, compact $I = [0,T]$ the solution y is polynomial-time computable since the ODE is analytic. On the other hand, this system can be solved explicitly and yields:

$$y_1(t) = e^t \qquad y_{i+1}(t) = e^{y_i(t)-1} \qquad y_d(t) = e^{e^{\cdot^{\cdot^{\cdot^{e^{e^t-1}}}}}-1}$$

One immediately sees that y_d being a tower of exponentials prevents y from being polynomial-time computable over $\mathbb{R}$, since one might need (supra-)exponential time just to write down an integer approximation of y_d with precision $1/2 = 2^{-1}$ for $d \geq 2$. This example shows that the solution of an analytic IVP (or even of a polynomial IVP) can be polynomial-time computable on any fixed compact set, while it may not necessarily be polynomial-time computable over $\mathbb{R}$.

A possible way to solve this problem is to analyze the complexity of (3.5) using a bound on the growth of the solution y as a parameter on the function used to measure the complexity (this is an example of *parametrized 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 in [13] for the case of polynomial IVPs (solutions of polynomial IVPs (3.5) where f is formed by polynomials are sometimes called PIVP functions) and in [12] for the case of analytic functions. The motivation for studying PIVPs is that the class of PIVP functions is well behaved and includes many interesting functions. In particular it contains all of the usual functions of analysis (exponential, trigonometric functions, polynomials, their inverses) and is closed under the usual arithmetic operations, composition, and ODE solving [21, 34]. It can also be shown that this class is exactly the class of functions generated by Shannon's General Purpose Analog Computer (GPAC) [77, 35], which is the mathematical model for analog computers (differential analyzers) used before the advent of digital computers [20] (see also the chapter "A Survey on Analog Models of Computation" in this handbook). The following result is from [13] and shows that the solution of a polynomial IVP can be computed in polynomial time with respect to a bound of $Y(t) = \max_{0 \leq u \leq t} \|y(u)\|$.

Theorem 3.3.11. *There exists an algorithm $\mathscr{A}$ which on any vector of polynomials p with polynomial-time computable coefficients, $\mu \in \mathbb{N}$, $t \in \mathbb{Q}$ with $t \geq 0$, and $Y \in \mathbb{Q}$ such that $Y \geqslant \max_{0 \leq u \leq t} \|y(u)\|$, satisfies*

$$\|\mathscr{A}(p,\mu,t,Y) - y(t)\|_{\infty} \leqslant 2^{-\mu}$$

where y is the solution of (3.5). Furthermore $\mathscr{A}(p,\mu,t,Y)$ is computed in time polynomial in the value of μ, t, and Y.

This result can be extended to IVPs where f is not necessarily a polynomial, assuming polynomial-time computability of the higher derivatives of f and an appropriate (polynomial) bound on the growth of those derivatives (see [13, Section 7]).

We note that Theorem 3.3.11 can be extended to the case where the initial condition is of the form $y(t_0) = y_0$, where $t_0 \in \mathbb{Q}$ and y_0 is polynomial-time computable (and it is actually stated in this way in the original paper [13]). This result is proved by using a variable-order method in which the solution of (3.5) is computed over a succession of subintervals $[a_i, a_{i+1}]$, with $i = 0, 1, 2, \ldots$ and $a_0 = 0$, whose union gives the maximal interval of definition of the solution y of (3.5). On each subinterval a Taylor approximation of the solution is computed, but using a variable-order method: instead of using an approximation of fixed order for each subinterval $[a_i, a_{i+1}]$, the order of the approximation is allowed to change on each interval. Note that the usual methods for numerical integrations (including basic Euler's method, etc.) fall in the general theory of n-order methods for some n. It was already suggested in [79, Section 3] that fixed-order methods might only run in exponential time when solving (3.5) for certain polynomial-time computable functions, but that variable-order methods might solve the same problem in polynomial time. Variable-order methods have also been used in certain contexts to solve IVPs, but usually without complexity results or with complexity results valid only for compact sets, see, e.g., [27, 45, 8, 1].

The previous result has the drawback that we need to know a bound $Y \geqslant \max_{0 \leq u \leq t} \|y(u)\|$ (which is used as input) to be able to compute $y(u)$ for $u \in [0, t]$.

This is improved in [64], where T and Y are replaced by a single parameter – the length of the solution curve y, with the added benefit that this parameter is not needed as an input to the algorithm. We recall that the length of the curve defined by the graph of a function f between $x = a$ and $x = b$ is

$$\text{length} = \int_a^b \sqrt{1 + (f'(x))^2} \, dx$$

In the case of the solution of (3.5) where $f = p$ is a vector of polynomials (i.e., each component of the vector p is a polynomial), we note that the derivative of the solution y is given by $p(y)$. If the degree of p is k (in the case where p has more than one component, the degree of p is the maximum of the degrees of its components), the length of the solution has a value which has an order of magnitude similar to the following quantity

$$\text{Len}(t) = \int_0^t \Sigma p \max(1, \|y(u)\|)^k \, du$$

where Σp denotes the sum of the absolute values of all the coefficients of p (or the maximum of such sums taken over each component of p, if p has more than one component). The following theorem is from [64].

Theorem 3.3.12. *There exists an algorithm $\mathscr{B}$ such that if $t_0, t \in \mathbb{R}$ with $t_0 \leq t$, $\varepsilon > 0$, and y satisfies (3.2) over $[t_0, t]$ and if*

$$x = \mathscr{B}(t_0, y_0, p, t, \varepsilon)$$

then $\|x - y(t)\| \leqslant \varepsilon$ and the algorithm finishes in time

$$O((\text{poly}(k, \text{Len}(t_0, t), \log \|y_0\|, \log \Sigma p, -\log \varepsilon))^n)$$

where $\text{Len}(t_0, t) = \int_{t_0}^t \Sigma p \max(1, \|y(u)\|)^k \, du$, *$k$ is the degree of p, n is the number of components of p, and* $\text{poly}(a_1, \ldots, a_j)$ *means polynomial in $a_1, \ldots, a_j$.*

More recently [85], [52] a similar result was established for IVPs with the form (3.2), where f is analytic, using techniques similar to those of [13] (i.e., using power series to compute the solution of the IVP). The authors also characterize the complexity of solving (3.2) using parametrized complexity. In particular, if f is polynomial-time computable in several input quantities (such as a bound for a complex extension of f; the inverse of the distance to a singularity; a bound on the norm of the point where f is evaluated; the precision up to which f must be calculated), then the solution of the IVP (3.2) is also polynomial-time computable on those inputs. See [85, Theorem 4.4.1] for more details.

In [85], [54] the above result is applied to some classes of volume-preserving systems such as Hamiltonian systems. In particular, for some classes of such systems, it can be shown that their average time complexity is polynomial-time computable, provided that hard instances are relatively rare. The authors also apply this result to show that the planar circular restricted three-body problem is polynomial-time computable on average under certain assumptions.

3.4 Computability of Qualitative Behaviors of Ordinary Differential Equations

Qualitative analysis of asymptotic (long-term) behaviors is a key tool for exploring ordinary differential equations because most ODEs do not have representation formulas for their solutions and numerically computed solutions are usually limited to the short term only. In some circumstances, in particular for hyperbolic systems, qualitative analysis may also provide algorithmic descriptions, to various degrees, of certain asymptotic behaviors. There are only a few results of this nature though, which are mainly about computation of trajectory behaviors near a hyperbolic equilibrium point.

We note that a statistical approach can also be used to obtain information about the asymptotic behavior of a dynamical system. However, since this approach involves measure theory, which is discussed in its own dedicated chapter of this book, results of this nature are not discussed in this chapter.

The asymptotic behavior of a differential equation is captured by its limit sets, which are the states the solutions reach after an infinite amount of time has passed. A limit set can be a point, a finite set of points, a curve, a manifold, or even a complicated set with a fractal structure known as a strange attractor. Limit sets are well understood in dimensions one and two: in dimension one and for continuous f in (3.1), the only possible limit sets are equilibrium points; in dimension two and for C^1 smooth f, a closed and bounded limit set other than a periodic orbit or an equilibrium point consists of equilibria and solutions connecting them according to the Poincaré-Bendixson Theorem, a celebrated result in the field of dynamical systems. Structures of limit sets in dimensions greater than two become much richer and more complex; for instance, strange attractors abound in $\mathbb{R}^3$.

An equilibrium point is the simplest type of limit set.

Definition 3.4.1. 1. A point $\tilde{y} \in \mathbb{R}^n$ is called an equilibrium point of (3.1) if $f(\tilde{y}) = 0$.
 2. An equilibrium point $\tilde{y}$ is called hyperbolic if none of the eigenvalues of $Df(\tilde{y})$ has zero real part, where $Df(\tilde{y})$ is the Jacobian of f at $\tilde{y}$. If the real part of every eigenvalue of $Df(\tilde{y})$ is negative (positive, respectively), then $\tilde{y}$ is called a sink (source, respectively).

The structure of solutions, also known as trajectories or orbits, near a hyperbolic equilibrium is characterized by the Stable Manifold Theorem, which is one of the most important results concerning the local qualitative theory of differential equations. From the perspective of computation, hyperbolicity ensures that Df does not have a computational singularity near $\tilde{y}$, an indication of the possible existence of an algorithmic characterization for the orbit structure near $\tilde{y}$. Note that Df presents the directions of the trajectories. We recall that the condition $x = 0$ for computable real numbers is considered a computational singularity because it cannot be decided effectively. The following theorem is an effective version of the Stable Manifold Theorem, which demonstrates that the trajectory structure near a hyperbolic equilibrium is indeed computable.

Theorem 3.4.2. *[Effective Stable Manifold Theorem [37]] Assume that $\tilde{y} \in \mathbb{R}^n$ is a hyperbolic equilibrium point of (3.1). Let $\phi(t, y_0)$ (or $\phi_t(y_0)$) denote the solution of (3.1) with the initial value y_0 at $t = 0$. Then there is a (Turing) algorithm that computes, on input $(f, Df, \tilde{y})$, a k-dimensional manifold $S \subset \mathbb{R}^n$, an $(n-k)$-dimensional manifold $U \subset \mathbb{R}^n$ with $0 \leq k \leq n$, both S and U containing $\tilde{y}$, and three positive rational numbers γ, ε, and δ such that*

(i) For all $y_0 \in S$, $\lim_{t \to +\infty} \phi(t, y_0) = \tilde{y}$ and for all $y_0 \in U$, $\lim_{t \to -\infty} \phi(t, y_0) = \tilde{y}$;
(ii)$|\phi(t, y_0) - \tilde{y}| \leq \gamma 2^{-\varepsilon t}$ for all $t \geq 0$ whenever $y_0 \in S$ and $y_0 \in B(\tilde{y}, \delta)$; $|\phi(t, y_0) - \tilde{y}| \leq \gamma 2^{\varepsilon t}$ for all $t \leq 0$ whenever $y_0 \in U$ and $y_0 \in B(\tilde{y}, \delta)$.

Moreover, if $k < n$ ($k > 0$), then a rational number η and a ball D centered at $\tilde{y}$ can be computed from f such that for any solution $\phi(t, y_0)$ to the equation (3.1) with $y_0 \in D \setminus S$, $\{\phi(t, y_0) : t \geq 0\} \not\subset B(\tilde{y}, \eta)$ ($y_0 \in D \setminus U$, $\{\phi(t, y_0) : t \leq 0\} \not\subset B(\tilde{y}, \eta)$, respectively) no matter how close the initial value y_0 is to $\tilde{y}$.

The classical proof of the Stable Manifold Theorem relies on the Jordan canonical form of $A = Df(\tilde{y})$. To reduce A to its Jordan form, one needs to find a basis of generalized eigenvectors. Since the process of finding eigenvectors from corresponding eigenvalues is not continuous in general, it is a non-computable process. Thus if one wishes to construct an algorithm that computes some S and U of (3.1) at y_0, a different method is needed. In [37] an analytic, rather than algebraic, approach to the eigenvalue problem is used to allow the computation of S and U without calling for eigenvectors. The analytic approach is based on function-theoretical treatment of resolvents (see, e.g., [84, 46, 47, 70]).

The local stable and unstable manifolds can be extended to the global stable and unstable manifolds by taking

$$W_f^s(y_0) = \bigcup_{j=0}^{\infty} \phi_{-j}(S), \quad W_f^u(y_0) = \bigcup_{j=0}^{\infty} \phi_j(U) \tag{3.7}$$

respectively. Is the global stable or unstable manifold computable for a computable hyperbolic equilibrium point? Since the global stable manifold of a sink s coincides with the basin of attraction of s, a less ambitious question is whether the basin of attraction of a computable sink is computable. The answer depends on the function f in (3.1). It is known that for hyperbolic rational functions, there are (polynomial-time) algorithms for computing basins of attraction and their complements (Julia sets) with arbitrary precision [9]; in other words, basins of attraction and Julia sets of hyperbolic rational functions are (polynomial-time) computable. On the other hand, for more general but still well-behaved functions f, the answer can be negative, even for computable and analytic functions (see [59, 100] for examples of non-computable basins of attraction occurring in C^k- and C^∞-systems). The following theorem (taken from [33]) shows that the basin of attraction of a computable sink in a computable and analytic system can be non-computable.

Theorem 3.4.3. *There exists an ODE (3.1) defined by an analytic and computable function f which admits a computable hyperbolic sink s such that the global stable manifold (basin of attraction) of s is not computable.*

Although the basin of attraction constructed in the theorem above is not computable, it is an r.e. open subset of $\mathbb{R}^n$ (see [100]); in other words, it can be filled up by a computable sequence of open balls. In general, it can be shown (see [37, Section 5]) that the degree of unsolvability of computing a global stable manifold is essentially Σ_2^0.

The proof of Theorem 3.4.3 is fairly complicated. It starts by encoding a well-known non-decidable problem into the basin of a computable hyperbolic sink in a discrete-time system; then this discrete-time system is embedded into a continuous-time system. The standard suspension method (see Smale [78], Arnold and Avez [6]) for embedding a discrete-time system into a continuous-time system is not sufficient for the construction; instead, a new method is developed for the embedding.

Another problem that is close in spirit to the Stable Manifold Theorem concerns linearization of the flow of a nonlinear ODE (3.1) near a hyperbolic equilibrium point. Poincaré originated the study and, later, a number of researchers contributed to progress on related problems, which resulted in the Hartman-Grobman Theorem (see, e.g., [63, Section 2.8]); the theorem asserts that near a hyperbolic equilibrium point $\tilde{y}$ of a nonlinear ODE (3.1), there is a homeomorphism H such that $H \circ \phi = L \circ H$, where ϕ is the solution to the ODE (3.1) and L is the solution to its linearization $\dot{y} = Df(\tilde{y})\,y$. In other words, near $\tilde{y}$, the ODE (3.1) has the same qualitative structure as its conjugated linear system. This theorem remains important, since it shows the structural stability of hyperbolic equilibria in sufficiently smooth dynamical systems. The computability study of this linearization is motivated by one of the seven open problems in the addendum to the book *Computability in Analysis and Physics* by Pour-El and Richards [67], who asked, "What is the connection between the computability of the original nonlinear operator and the linear operator which results from it?" It is proved in [38] that the homeomorphism H that performs the linearization is computable. The precise statement is given below.

For simplicity only the linearization near the origin is considered. Let $\mathscr{A}_H$ denote the set of all hyperbolic $n \times n$ matrices, where a matrix A is hyperbolic if all of its eigenvalues have nonzero real part. The operator norm is used for $A \in \mathscr{A}_H$; i.e., $\|A\| = \sup_{|x| \neq 0} |Ax|/|x|$. For Banach spaces X and Y, let $C^k(X;Y)$ denote the set of all continuously k-times differentiable functions defined on open subsets of X with ranges in Y, and $\mathscr{L}(X;Y)$ the set of all bounded linear maps from X to Y. Let $\mathscr{O}$ denote the set of all open subsets of $\mathbb{R}^n$ containing the origin of $\mathbb{R}^n$, $\mathscr{I}$ the set of all open intervals of $\mathbb{R}$ containing zero, and $\mathscr{F}$ the set of all functions $f \in C^1(\mathbb{R}^n;\mathbb{R}^n)$ such that the domain of f is in $\mathscr{O}$, $f(0) = 0$, and $Df(0) \in \mathscr{A}_H$. In other words, for any $f \in \mathscr{F}$, 0 is a hyperbolic equilibrium point of f.

Theorem 3.4.4. *[Effective Hartman-Grobman Theorem [38]] There is a computable map $\Theta : \mathscr{F} \to \mathscr{O} \times \mathscr{O} \times C(\mathbb{R}^n;\mathbb{R}) \times C(\mathbb{R}^n;\mathbb{R}^n)$ such that for any $f \in \mathscr{F}$, $f \mapsto (U,V,\mu,H)$, where*

(a) $H : U \to V$ is a homeomorphism;

(b) the unique solution $y(t, y_0) = y(y_0)(t)$ to the initial value problem $\dot{y} = f(y)$ and $y(0) = y_0$ is defined on $(-\mu(y_0), \mu(y_0))$ for all $y_0 \in U$; moreover, $y(t, y_0) \in U$ for all $y_0 \in U$ and $-\mu(y_0) < t < \mu(y_0)$;
(c) $H(y(t, y_0)) = e^{Df(0)t} H(y_0)$ for all $y_0 \in U$ and $-\mu(y_0) < t < \mu(y_0)$.

Recall that for any $y_0 \in \mathbb{R}^n$, $e^{Df(0)t} y_0$ is the solution to the linear problem $\dot{y} = Df(0)y$, $y(0) = y_0$. So the theorem shows that the homeomorphism H, computable from f, maps trajectories near 0, a hyperbolic equilibrium point, of the nonlinear problem $\dot{y} = f(y)$ onto trajectories near the origin of the linear problem $\dot{y} = Df(0)y$. In other words, H is a conjugacy between the linear and nonlinear trajectories near the origin. Since the classical proofs of the Hartman-Grobman linearization theorem are not constructive, the effective version of the theorem cannot be obtained from a classical proof.

The results discussed so far are centered at hyperbolic equilibrium points. Evidently, hyperbolicity and locality (in the spatial sense) are two key conditions for obtaining algorithmic descriptions about asymptotic behaviors of trajectories of the ODE (3.1) near an equilibrium; on the other hand, the role played by the structure f, near an equilibrium, of the ODE (3.1) turns out not to be as crucial, since the computations for a local stable manifold, a local unstable manifold, and the conjugacy H at a hyperbolic equilibrium are all uniform in f.

Equilibrium points are the simplest limit sets, and there are many limit sets other than equilibrium points. What about computability of other limit sets, in particular, those with complicated fractal structures such as strange attractors? The question is largely open. There is however one type of strange attractors – geometric Lorenz attractors – that has been proved to be computable recently. The Lorenz attractor is perhaps the most famous strange attractor. It was introduced in 1963 by E.N. Lorenz as one of the first examples of *strange attractors* (see, e.g., [41, 44] for a treatment of the Lorenz attractor and [4] for a more in-depth analysis of Lorenz-like attractors). However, Lorenz's research was mainly based on (non-rigorous) numerical simulations and, until recently, the proof of the existence of the Lorenz attractor remained elusive. To address that problem, Afraimovich, Bykov, and Shil'nikov [3], and Guckenheimer and Williams [42] originated the study of flows satisfying a certain list of geometric properties based on the behavior observed in the numerical simulations of the Lorenz equation. In particular, they proved that any such flow must contain a strange attractor. These examples came to be known as geometric Lorenz models, and the strange attractor contained in a geometric Lorenz flow is called the geometric Lorenz attractor. In 2002 Tucker [86] used a combination of normal form theory and rigorous numerics to provide a formal proof of the existence of the Lorenz attractor by showing that geometric Lorenz models do indeed correspond to the Lorenz system for certain parameters. Since a geometric Lorenz model supports a strange attractor, so does the Lorenz system. Recently, the following result concerning computability of geometric Lorenz attractors was proved in [40].

Theorem 3.4.5. *For any geometric Lorenz flow, if the data defining the flow are computable, then its attractor is a computable subset of $\mathbb{R}^3$. Moreover, the physical measure supported on this attractor is a computable probability measure.*

We should also note that some related problems can be found in verification theory. For example, in [24] the reachability problem is analyzed from a computability perspective. The reachability problem consists of finding the set of all points which can be reached in finite time from trajectories starting in an initial set. It is shown in [24] that the reachable set is lower-computable for ODEs (or, more generally, differential inclusions). However computability only holds under certain conditions.

3.5 Computability of Partial Differential Equations

The area of partial differential equations is vast and no general theory is known concerning the solvability of all PDEs. In fact, most PDEs do not even have classical solutions (a solution of a PDE of order k is classical if it is at least k times continuously differentiable in the uniform norm). Instead, they may only be solvable in the sense of weak or generalized solutions defined in some function spaces equipped with an energy-type norm such as Sobolev spaces or in the space of generalized functions. Computability theories of these spaces are established in [98, 101]. Many modern approaches to the subject – energy methods within Sobolev spaces, the calculus of variations, conservation laws, semigroups, etc. – are also useful tools in the recursion-theoretic study of the subject; in particular, energy methods augmented by a fixed-point argument.

An equation is exactly solvable if there is an explicit formula for its (classic or weak) solution. Only a few solvable partial differential equations are exactly solvable. Pour-El and Richards have studied three important second-order linear exactly solvable PDEs from the viewpoint of computability [67]: the heat equation, Laplace's equation, and the wave equation. They showed, by exploiting the solution formulas, that the heat equation and Laplace's equation admit directly computable classical solutions, at least in the case where good domains such as rectangular regions with computable corners are considered. For the wave equation, they demonstrated that a computable initial datum may generate a non-computable wave propagation measured in the uniform norm (also see [66, 68]. A discussion of those results can also be found in [31]); but in a more physically relevant setting, the wave equation does admit computable solutions in the energy norm (see, for example, [92, 88, 91]). This study indicates that computability of the solution operator of a PDE, if the PDE is uniquely solvable, depends not only on the structure of the equation but also on the physical measurements used for the initial/boundary function(s) and for the solution. Another notable fact revealed by the study is that the uniqueness of solution of a computable initial value problem for a PDE doesn't guarantee the solution being computable – a significant distinction between the computability theories of ODEs and PDEs. A related result can be found in [22], where the difficulty of solving (ordinary and partial) differential equations is studied using in-

dex sets. Several cases of differential equations are considered (e.g., equations with locally computable solutions) and are shown to correspond to certain levels in the arithmetical hierarchy, which depend on the particular problem being considered. This type of results provides more precise characterizations of non-computability results for differential equations presented in [65] and [66].

Since no general theory is known concerning the solvability of all partial differential equations, research in the field focuses on various particular PDEs which are important for applications within and outside of mathematics; for example, the KdV equation, the Schrödinger equation, and the Navier-Stokes equation. These important equations also lie at the center of the computability study of nonlinear PDEs. Indeed, as one of the seven open problems included in the Addendum in their 1989 book Computability in Analysis and Physics, Pour-El and Richards suggested: "A third area of study is the recursion-theoretic study of popular nonlinear problems of classical importance. Examples are the Navier-Stokes equation, the KdV equation, and the complex of problems associated with Feigenbaum's constant." The two theorems below give partial answers to the open question.

The first result concerns the KdV equation, which is a mathematical model of waves on shallow water surfaces. The initial value problem of the KdV equation posed on the whole real line $\mathbb{R}$:

$$\begin{cases} u_t + u u_x + u_{xxx} = 0, \, t, x \in \mathbb{R} \\ u(x,0) = \varphi(x) \end{cases}$$

defines a nonlinear map (the solution operator) $K_{\mathbb{R}}$ from the Sobolev space $H^s(\mathbb{R})$ to the space $C(\mathbb{R}; H^s(\mathbb{R}))$ for real numbers $s \geq 0$, where u represents the amplitude of the wave. It is shown in [94, 93] that for any integer $s \geq 3$, the map $K_{\mathbb{R}} : H^s(\mathbb{R}) \to C(\mathbb{R}; H^s(\mathbb{R}))$ is Turing computable, which means that the solution $K_{\mathbb{R}}(\varphi)$ can be computed with arbitrary precision on Turing machines when sufficiently good approximations to the initial function φ are available (in [30] a preliminary computability result was presented for the periodic case). Note that the solution operator $K_{\mathbb{R}}$ is defined for real numbers $s \geq 0$ while the computability of $K_{\mathbb{R}}$ is established only for integers $s \geq 3$ (the proof can be extended to real numbers $s \geq 3$ with some modifications). The stronger smoothness requirement is used for ensuring that several derivatives in the construction of approximations remain computable. It would be very interesting to know whether $K_{\mathbb{R}}$ remains computable for real numbers $0 \leq s < 3$.

We also note that computability results exist for similar PDEs. For example, in [99] the initial and boundary value problem (IBVP)

$$\begin{cases} u_t + u_x + u u_x + u_{xxx} = 0, \qquad t, x \geq 0 \\ u(x,0) = \varphi(x), u(0,t) = h(t) \end{cases}$$

is considered; it is shown that the operator which maps the initial and boundary data to the solution of the IBVP is computable.

The (incompressible) Navier-Stokes Equation

$$\partial_t \mathbf{u} - \triangle \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla P = \mathbf{f}, \quad \nabla \mathbf{u} = 0, \quad \mathbf{u}(0) = \mathbf{a}, \quad \mathbf{u}\big|_{\partial\Omega} \equiv \mathbf{0} \qquad (3.8)$$

describes the motion of a viscous incompressible fluid filling a rigid box $\overline{\Omega}$. The vector field $\mathbf{u} = \mathbf{u}(\mathbf{x},t) = \big(u^{(1)}, u^{(2)}, \ldots, u^{(d)}\big)$ represents the velocity of the fluid, $P = P(\mathbf{x},t)$ is the scalar pressure, and $\mathbf{f}$ is a given external force. The question of global existence and smoothness of solutions of Equation (3.8), even in the homogeneous case $\mathbf{f} \equiv \mathbf{0}$, is one of the seven Millennium Prize Problems posted by the Clay Mathematics Institute at the beginning of the twenty-first century. Local existence has been established, though, in various L^p settings [32], and uniqueness of weak solutions in dimension 2, but not in dimension 3 [80, 14]. Nevertheless, numerical solution methods have been devised in abundance, often based on pointwise (or even uniform, rather than L^p) approximation and struggling with unphysical artifacts [62]. Thus, it becomes useful to know whether it is possible to compute a local solution with arbitrary precision in a rigorous mathematical/physical setting. A recent result provides a positive answer: it is shown in [83] that the solution operator is indeed Turing computable. More precisely, it is shown that there exist a computable map T,

$$T : L_{2,0}^\sigma(\Omega) \times C([0,\infty), L_{2,0}^\sigma(\Omega)) \to (0,\infty), (\mathbf{a}, \mathbf{f}) \mapsto T(\mathbf{a}, \mathbf{f})$$

and a computable map $\mathscr{S}$,

$$\mathscr{S} : L_{2,0}^\sigma(\Omega) \times C([0,\infty), L_{2,0}^\sigma(\Omega)) \to C([0,\infty), L_{2,0}^\sigma(\Omega) \times L_2(\Omega))$$

such that for every $\mathbf{a} \in L_{2,0}^\sigma(\Omega)$ and $\mathbf{f} \in C([0,\infty), L_{2,0}^\sigma(\Omega))$, the restriction

$$\mathscr{S}(\mathbf{a}, \mathbf{f})\big|_{[0,T(\mathbf{a},\mathbf{f})]} = (\mathbf{u}, P)$$

constitutes a (strong local) solution to Equation (3.8), where $\Omega = (-1,1)^2$ and $L_{2,0}^\sigma(\Omega)$ is the set of all square-integrable, divergence-free and boundary-free functions defined on Ω.

The proofs of both theorems make use of energy methods augmented by a fixed-point argument by converting the PDE into an integral equation, assembling an iterate scheme, and then proving that the sequence generated by the iteration is computable as well as effectively convergent and the unique limit is the computable solution sought after. There are several typical difficulties in such constructions. Firstly, there is often a need for some custom-designed representations for the spaces of input and output functions, which are different from known canonical ones. For example, the weak solution of the Navier-Stokes equation (3.8) defined on a two-dimensional good region is a locally square-integrable, solenoidal (i.e., divergence-free), and boundary-free vector field. The canonical representation for an L^p space using polynomials with rational coefficients is not rich enough to encode a boundary-free solenoid. Yet a strengthened representation for the boundary-free solenoids must be sufficiently general and robust so that the information carried by the representation won't be destroyed by elementary analytical operations such as (distributive) differentiation and (primitive) integration. The second challenging

issue is that the iteration usually involves several operations, which may move back and forth among different spaces. Additional care for representations may become inevitable in order to keep the computations flowing in and out from one space to another. Thirdly, the iterate sequence may (effectively) converge only in a special subspace of the space where the sequence is defined, which may further complicate matters. For instance, Bourgain's space is needed in showing that the iterate sequence is contracting and therefore converges to the solution of the KdV equation.

In addition to particular equations, there are studies on computability aspects of several classes of PDEs. One such class is linear parabolic equations. An initial value problem of a linear parabolic equation can be written in the form:

$$\begin{cases} du(t)/dt + \mathscr{A}u(t) = f(t), & t > 0 \\ u(0) = x, & x \in X \end{cases} \tag{3.9}$$

where $f : [0,T] \to X$ is a continuous map and $\mathscr{A}$ is a strong elliptic operator, bounded or unbounded, on a Banach space X. When $X = L^2(\mathbb{R}^n)$, the solution to the initial value problem (3.9) is given by the formula:

$$u(t) = W(t)x + \int_0^t W(t-s)f(s)\mathrm{d}s$$

where $W(t)$ is a C_0 semigroup generated by the infinitesimal generator $\mathscr{A}$. Thus if the semigroup W is computable from $\mathscr{A}$, then the solution operator $\mathscr{K} : (\mathscr{A}, f, x) \mapsto u$ is computable uniformly from the data $(\mathscr{A}, f, x)$ defining the problem. The study concerning computability of semigroups is carried out in [96, 97]. It is shown in [96] how to compute on Turing machines a uniformly continuous or strongly continuous semigroup uniformly from its infinitesimal generator and vice versa.

Another class for which a master algorithm exists for computing (local) solutions of any of its members is the class of general (nonlinear) first-order PDEs for scalar functions with several independent variables. A problem in this class can be written in the form:

$$F(Du, u, x) = 0 \quad \text{in } U \tag{3.10}$$

subject to the boundary condition

$$u = g \quad \text{on } \partial U \text{ (the boundary of } U) \tag{3.11}$$

where $x \in U$, U is an open subset of $\mathbb{R}^n$, $F : \mathbb{R}^n \times \mathbb{R} \times V \to \mathbb{R}$ and $g : \partial U \to \mathbb{R}$ are given, V is an open subset of $\mathbb{R}^n$ containing $\overline{U}$, the closure of U, and $u : \overline{U} \to \mathbb{R}$ is the unknown and Du is the gradient of u. A function u is called a (strong) solution of the boundary value problem if it solves (3.10), (3.11). A special feature of equation (3.10) is that it can, locally at least, be transformed into a system of first-order ordinary differential equations, called the characteristic equations, as follows: to compute the solution $u(x)$ at a point $x \in U$, one finds some curve $\mathbf{x}(s)$ lying within U, connecting x with a point $x^0 \in \partial U$, such that along the curve $\mathbf{x}(s)$ the boundary value problem for the PDE is reduced to an initial value problem for the characteristic equations. As discussed in Section 3.2, a system of ODEs is in prin-

ciple algorithmically uniformly solvable, sometimes quite explicitly. It is shown in
[82] that the method of characteristics can be used to compute local solutions of
(3.10), (3.11), at least for feasible instances and when the boundary ∂U is effec-
tively locally C^1. In the same article, it is also shown that although the solution is
locally computable the maximal region of existence of a computable local solution
may not be computable. Moreover, the problem whether a boundary value problem
has a global solution is not algorithmically decidable. The negative results hold even
within the class of quasilinear equations defined by analytic computable functions
over particularly simple domains (quasilinear equations are among the simplest first-
order nonlinear partial differential equations). This fact shows that the algorithmic
unsolvability is intrinsic.

In [15] uniform computability results are given for some simple one-dimensional
elliptic boundary value problems. In particular, computability results are given for
problems with the following format: (i) $-u'' + u' + u = f$ on $[0,1]$, with $u'(0) =
0$ and $u'(1) = 0$; (ii) $-u'' + au' + u = f$ on $[0,1]$, with $u(0) = u(1) = 0$; and (iii)
$-u'' = f$ on $[0,1]$, with $u(0) = u(1) = 0$. The computability of the solutions of these
problems follows from computable versions of the Fréchet-Riesz Theorem and the
Lax-Milgram Theorem, which are also proved in [15].

In [53] the computational complexity of the Dirichlet problem for Poisson's
equation

$$\Delta u = f \text{ in } \Omega, u|_{\partial \Omega} = g$$

is analyzed, where $\Omega \subseteq \mathbb{R}^d$ is a domain, $f : \Omega \to \mathbb{R}$ and $g : \partial \Omega \to \mathbb{R}$ are given
functions, and Δ is the Laplace operator. It is shown that, for fixed d and for $\Omega =
B(0,1) = \{x \in \mathbb{R}^d : \|x\| \leq 1\}$, solving the Dirichlet problem for Poisson's equation
is "complete" for the complexity class #P in the sense that $\mathscr{FP} = \#P$ if and only if
the solution of the Dirichlet problem for Poisson's equation on $B(0,1)$ is computable
in polynomial time for any choice of polynomial-time computable functions f and
g.

In [95] (see also [89, 90]) the computability of the initial value problems for the
linear Schrödinger equation $u_t = i\Delta u + \phi$ and the nonlinear Schrödinger equation
$iu_t = -\Delta u + mu + |u|^2 u$ is studied and it is shown that the solution operator is com-
putable if the initial data are Sobolev functions but noncomputable in the linear case
if the initial data are L^p-functions and $p \neq 2$.

In [72] the authors study symmetric hyperbolic systems of PDEs of the form

$$\begin{cases} A\frac{\partial u}{\partial t} + \sum_{i=1}^m B_i \frac{\partial u}{\partial x_i} = 0 \\ u(x,0) = \varphi(x) \end{cases} \tag{3.12}$$

where $A, B_1, \ldots, B_m$ are computable constant symmetric $n \times n$ matrices, $t \geq 0$, and
$x \in [0,1]^m$. There it is shown that if the matrices are fixed and if the first- and second-
order derivatives of the initial function φ are bounded, the operator which maps
the initial condition φ to the solution of the PDE is computable (see also [73] for
a correction and [74, Remark 5.5.3] for the statement given here, which removes
some unneeded assumptions used in the original formulation of [72]). Later it was
shown in [74, Theorem 5.3] that the operator which maps an initial condition and the

matrices $A, B_1, \ldots, B_m$ to the solution of the PDE is uniformly computable, provided the matrices satisfy some algebraic conditions.

More recently, the computational complexity of solving PDEs with the format (3.12) was studied in [76, 75]. There the (uniform) complexity of this problem is shown to be in EXPTIME in general and in PTIME under certain conditions.

The Dirichlet Problem was also analyzed in [16, 17, 18] using Bishop's framework of constructive mathematics. It is shown that weak solutions of the Dirichlet Problem exist if certain conditions hold.

3.6 Some Open Problems

In this chapter, we have presented several results concerning the computability and computational complexity of differential equations. However, the field of differential equations is vast, and many more problems in the field await exploration from the perspectives of computable analysis. Here we merely give two open problems.

1. The first problem is to give general principles from which the computability or noncomputability of limit sets of ODEs follows as corollaries. For example, it has been shown that a basin of attraction of an ODE defined by a hyperbolic rational function f is computable, uniformly in f ([9]); however, on the other hand, if f is analytic, computable but non-hyperbolic, a basin of attraction is no longer necessarily computable ([33]). It would be interesting to know whether a computable, analytic and hyperbolic f ensures the computability of a basin of attraction.
2. The second problem is to characterize the computational complexity of well-known computable PDEs such as the KdV equation, the Schrödinger equation, and the Navier-Stokes equation.

References

1. Abad, A., Barrio, R., Blesa, F., Rodríguez, M.: Algorithm 924: Tides, a Taylor series integrator for differential equations. ACM Trans. Math. Softw. **39**(1), 5:1–5:28 (2012)
2. Aberth, O.: The failure in computable analysis of a classical existence theorem for differential equations. Proceedings of the American Mathematical Society **30**, 151–156 (1971)
3. Afraimovich, V.S., Bykov, V.V., Shil'nikov, L.P.: On the appearence and structure of the Lorenz attractor. Dokl. Acad. Sci. USSR **234**, 336–339 (1977)
4. Araújo, V., Pacifico, M.: Three-Dimensional Flows. Springer (2010)
5. Arnold, V.I.: Ordinary Differential Equations. MIT Press (1978)

6. Arnold, V.I., Avez, A.: Ergodic Problems of Classical Mechanics. W.A. Benjamin (1968)
7. Atkinson, K.E.: An Introduction to Numerical Analysis, 2nd edn. John Wiley & Sons (1989)
8. Barrio, R., Rodríguez, M., Abad, A., Blesa, F.: Breaking the limits: the Taylor series method.
    Appl. Math. Comput. **217**(20), 7940–7954 (2011)
9. Beardon, A.F.: Iteration of Rational Functions. Springer (1991)
10. Birkhoff, G. (ed.): A Source Book in Classical Analysis, chap. Osgood's Existence Theorem,
    pp. 251–258. Harvard University Press (1973)
11. Birkhoff, G., Rota, G.C.: Ordinary Differential Equations, 4th edn. John Wiley & Sons
    (1989)
12. Bournez, O., Graça, D.S., Pouly, A.: Solving analytic differential equations in polynomial
    time over unbounded domains. In: F. Murlak, P. Sankowski (eds.) Mathematical Foundations
    of Computer Science 2011, *Lecture Notes in Computer Science*, vol. 6907, pp. 170–181.
    Springer, Heidelberg (2011)
13. Bournez, O., Graça, D.S., Pouly, A.: On the complexity of solving initial value problems. In:
    J. van der Hoeven, M. van Hoeij (eds.) ISSAC 2012—Proceedings of the 37th International
    Symposium on Symbolic and Algebraic Computation, pp. 115–121. ACM, New York (2012)
14. Boyer, F., Fabrie, P.: Mathematical Tools for the Study of the Incompressible Navier-Stokes
    Equations and Related Models. Springer (2013)
15. Brattka, V., Yoshikawa, A.: Towards computability of elliptic boundary value problems in
    variational formulation. Journal of Complexity **22**(6), 858–880 (2006)
16. Bridges, D.S., Wang, Y.: Constructive aspects of the Dirichlet problem. Journal of Universal
    Computer Science **3**(11), 1148–1161 (electronic) (1997)
17. Bridges, D.S., Wang, Y.: Constructive weak solutions of the Dirichlet problem. The Journal
    of the London Mathematical Society. Second Series **57**(3), 655–667 (1998)
18. Bridges, D.S., Wang, Y.: Weak solutions of the Dirichlet problem and the locatedness of
    $H_0^1(\omega)$. New Zealand Journal of Mathematics **27**(1), 1–5 (1998)
19. Buser, P., Scarpellini, B.: Recursive analysis of singular ordinary differential equations. An-
    nals of Pure and Applied Logic **162**, 20–35 (2010)
20. Bush, V.: The differential analyzer. A new machine for solving differential equations. J.
    Franklin Inst. **212**, 447–488 (1931)
21. Carothers, D.C., Parker, G.E., Sochacki, J.S., Warne, P.G.: Some properties of solutions to
    polynomial systems of differential equations. Electron. J. Diff. Eqns. **2005**(40) (2005)
22. Cenzer, D., Remmel, J.B.: Index sets for computable differential equations. Mathematical
    Logic Quarterly **50**(4,5), 329–344 (2004)
23. Cleave, J.: The primitive recursive analysis of ordinary differential equations and the com-
    plexity of their solutions. Journal of Computer and Systems Sciences **3**, 447–455 (1969)
24. Collins, P.: Continuity and computability on reachable sets. Theoretical Computer Science
    341, 162–195 (2005)
25. Collins, P., Graça, D.: Effective computability of solutions of ordinary differential equations
    the thousand monkeys approach. In: V. Brattka, R. Dillhage, T. Grubba, A. Klutsch (eds.)
    CCA 2008, Fifth International Conference on Computability and Complexity in Analysis,
    Electronic Notes in Theoretical Computer Science, vol. 221, pp. 103–114. Elsevier (2008),
    CCA 2008, Fifth International Conference, Hagen, Germany, August 21–24, 2008
26. Collins, P., Graça, D.S.: Effective computability of solutions of differential inclusions: the
    ten thousand monkeys approach. Journal of Universal Computer Science **15**(6), 1162–1185
    (2009)
27. Corliss, G., Chang, Y.F.: Solving ordinary differential equations using Taylor series. ACM
    Trans. Math. Softw. **8**(2), 114–144 (1982)
28. Edalat, A., Pattinson, D.: A domain-theoretic account of Picard's theorem. LMS Journal of
    Computation and Mathematics **10**, 83–118 (2007)
29. Farjudian, A.: Polynomial-time solution of initial value problems using polynomial enclo-
    sures. In: L. Ong, R. de Queiroz (eds.) Logic, Language, Information and Computation,
    Lecture Notes in Computer Science, vol. 7456, pp. 232–245. Springer, Heidelberg (2012)
30. Gay, W., Zhang, B.Y., Zhong, N.: Computability of solutions of the Korteweg-de Vries equa-
    tion. Mathematical Logic Quarterly **47**(1), 93–110 (2001)

31. Gherardi, G.: Computability and incomputability of differential equations. In: G. Corsi, R. Lupacchini (eds.) Deduction, Computation, Experiment, pp. 223–242. Springer Verlag, Milano (2008)
32. Giga, Y., Miyakawa, T.: Solutions in L_r of the Navier-Stokes initial value problem. Archive for Rational Mechanics and Analysis **89**, 267–281 (1985)
33. Graça, D., Zhong, N.: An analytic system with a computable hyperbolic sink whose basin of attraction is non-computable. Theory of Computing Systems **57**(2), 478–520 (2015)
34. Graça, D.S., Campagnolo, M.L., Buescu, J.: Computability with polynomial differential equations. Advances in Applied Mathematics **40**(3), 330–349 (2008)
35. Graça, D.S., Costa, J.F.: Analog computers and recursive functions over the reals. Journal of Complexity **19**(5), 644–664 (2003)
36. Graça, D.S., Zhong, N., Buescu, J.: Computability, noncomputability and undecidability of maximal intervals of IVPs. Transactions of the American Mathematical Society **361**(6), 2913–2927 (2009)
37. Graça, D.S., Zhong, N., Buescu, J.: Computability, noncomputability, and hyperbolic systems. Appl. Math. Comput. **219**(6), 3039–3054 (2012)
38. Graça, D.S., Zhong, N., Dumas, H.S.: The connection between computability of a nonlinear problem and its linearization: The Hartman-Grobman theorem revisited. Theoret. Comput. Sci. **457**, 101–110 (2012)
39. Graça, D.S., Buescu, J., Campagnolo, M.L.: Boundedness of the domain of definition is undecidable for polynomial ODEs. In: R. Dillhage, T. Grubba, A. Sorbi, K. Weihrauch, N. Zhong (eds.) Proceedings of the Fourth International Conference on Computability and Complexity in Analysis (CCA 2007), *Electronic Notes in Theoretical Computer Science*, vol. 202, pp. 49–57. Elsevier (2008), CCA 2007, Siena, Italy, June 16–18, 2007
40. Graça, D.S., Rojas, C., Zhong, N.: Computing geometric Lorenz attractors with arbitrary precision. Transactions of the American Mathematical Society **370**(4), 2955–2970 (2018)
41. Guckenheimer, J., Holmes, P.: Nonlinear Oscillations, Dynamical Systems, and Bifurcation of Vector Fields. Springer (1983)
42. Guckenheimer, J., Williams, R.F.: Structural stability of Lorenz attractors. Publ. Math. IHES **50**, 59–72 (1979)
43. Hartman, P.: Ordinary Differential Equations, 2nd edn. Birkhäuser (1982)
44. Hirsch, M.W., Smale, S., Devaney, R.: Differential Equations, Dynamical Systems, and an Introduction to Chaos. Academic Press (2004)
45. Jorba, À., Zou, M.: A software package for the numerical integration of ODEs by means of high-order taylor methods. Experimental Mathematics **14**(1), 99–117 (2005)
46. Kato, T.: On the convergence of the perturbation method. i. Progr. Theor. Phys. **4**(4), 514–523 (1949)
47. Kato, T.: On the convergence of the perturbation method. ii. Progr. Theor. Phys. **5**(1), 95–101 (1950)
48. Kawamura, A.: Lipschitz continuous ordinary differential equations are polynomial-space complete. In: 24th Annual IEEE Conference on Computational Complexity, pp. 149–160. IEEE Computer Soc., Los Alamitos, CA (2009)
49. Kawamura, A.: Lipschitz continuous ordinary differential equations are polynomial-space complete. Computational Complexity **19**(2), 305–332 (2010)
50. Kawamura, A., Cook, S.A.: Complexity theory for operators in analysis. ACM Transactions on Computation Theory **4**(2), 5:1–5:24 (2012)
51. Kawamura, A., Ota, H., Rösnick, C., Ziegler, M.: Computational complexity of smooth differential equations. Logical Methods in Computer Science **10**, 1:6,15 (2014)
52. Kawamura, A., Steinberg, F., Thies, H.: Parameterized complexity for uniform operators on multidimensional analytic functions and ODE solving. In: L. Moss, R. de Queiroz, M. Martinez (eds.) Logic, language, information, and computation, *Lecture Notes in Computer Science*, vol. 10944, pp. 223–236. Springer, Berlin (2018)
53. Kawamura, A., Steinberg, F., Ziegler, M.: On the computational complexity of the Dirichlet problem for Poisson's equation. Mathematical Structures in Computer Science **27**(8), 1437–1465 (2017)

54. Kawamura, A., Thies, H., Ziegler, M.: Average-case polynomial-time computability of Hamiltonian dynamics. In: 43rd International Symposium on Mathematical Foundations of Computer Science, *LIPIcs. Leibniz Int. Proc. Inform.*, vol. 117, pp. Art. No. 30, 17. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern (2018)
55. Ko, K.I.: On the computational complexity of ordinary differential equations. Inform. Contr. **58**, 157–194 (1983)
56. Ko, K.I.: Complexity Theory of Real Functions. *Progress in Theoretical Computer Science*. Birkhäuser, Boston (1991)
57. Ko, K.I., Friedman, H.: Computing power series in polynomial time. Advances in Applied Math. **9**, 40–50 (1988)
58. Miller, W.: Recursive function theory and numerical analysis. Journal of Computer and Systems Sciences **4**, 465–472 (1970)
59. Moore, C.: Generalized shifts: unpredictability and undecidability in dynamical systems. Nonlinearity **4**(2), 199–230 (1991)
60. Müller, N.T.: Uniform computational complexity of Taylor series. In: T. Ottmann (ed.) Proceedings of the 14th International Colloquium on Automata, Languages, and Programming, *Lecture Notes in Computer Science*, vol. 267, pp. 435–444. Springer, Berlin (1987)
61. Müller, N.T., Moiske, B.: Solving initial value problems in polynomial time. In: Proceedings of the 22th JAIIO - Panel'93, Part 2, pp. 283–293 (1993), Buenos Aires, 1993
62. Patel, M.K., Markatos, N.C., Cross, M.: A critical evaluation of seven discretization schemes for convection-diffusion equations. International Journal for Numerical Methods in Fluids **5**(3), 225–244 (1985)
63. Perko, L.: Differential Equations and Dynamical Systems, 3rd edn. Springer (2001)
64. Pouly, A., Graça, D.S.: Computational complexity of solving polynomial differential equations over unbounded domains. Theoretical Computer Science **626**, 67–82 (2016)
65. Pour-El, M.B., Richards, J.I.: A computable ordinary differential equation which possesses no computable solution. Annals Math. Logic **17**, 61–90 (1979)
66. Pour-El, M.B., Richards, J.I.: The wave equation with computable initial data such that its unique solution is not computable. Advances in Math. **39**, 215–239 (1981)
67. Pour-El, M.B., Richards, J.I.: Computability in Analysis and Physics. *Perspectives in Mathematical Logic*. Springer, Berlin (1989)
68. Pour-El, M.B., Zhong, N.: The wave equation with computable initial data whose unique solution is nowhere computable. Mathematical Logic Quarterly **43**(4), 499–509 (1997)
69. Rettinger, R., Weihrauch, K., Zhong, N.: Topological complexity of blowup problems. Journal of Universal Computer Science **15**(6), 1301–1316 (2009)
70. Robinson, C.: Dynamical Systems. CRC Press (1995)
71. Ruohonen, K.: An effective Cauchy-Peano existence theorem for unique solutions. Internat. J. Found. Comput. Sci. **7**(2), 151–160 (1996)
72. Selivanova, S., Selivanov, V.: Computing solutions of symmetric hyperbolic systems of PDE's. In: V. Brattka, R. Dillhage, T. Grubba, A. Klutsch (eds.) CCA 2008, Fifth International Conference on Computability and Complexity in Analysis, *Electronic Notes in Theoretical Computer Science*, vol. 221, pp. 243–255. Elsevier (2008), CCA 2008, Fifth International Conference, Hagen, Germany, August 21–24, 2008
73. Selivanova, S., Selivanov, V.: Computing the solution operators of symmetric hyperbolic systems of PDE. Journal of Universal Computer Science **15**(6), 1337–1364 (2009)
74. Selivanova, S., Selivanov, V.: Computing solution operators of boundary-value problems for some linear hyperbolic systems of PDEs. Logical Methods in Computer Science **13**(4) (2017)
75. Selivanova, S., Selivanov, V.: Bit complexity of computing solutions for symmetric hyperbolic systems of PDEs with guaranteed precision. Computability **10**, (to appear) (2021)
76. Selivanova, S.V., Selivanov, V.L.: Bit complexity of computing solutions for symmetric hyperbolic systems of PDEs (extended abstract). In: F. Manea, R. Miller, D. Nowotka (eds.) Sailing Routes in the World of Computation, *Lecture Notes in Computer Science*, vol. 10936, pp. 376–385. Springer, Cham (2018)
77. Shannon, C.E.: Mathematical theory of the differential analyzer. J. Math. Phys. MIT **20**, 337–354 (1941)

78. Smale, S.: Differentiable dynamical systems. Bull. Amer. Math. Soc. **73**, 747–817 (1967)

79. Smith, W.D.: Church's thesis meets the N-body problem. Applied Mathematics and Computation **178**(1), 154–183 (2006)

80. Sohr, H.: The Navier-Stokes Equations: An Elementary Functional Analytic Approach. Birkhäuser (2001)

81. Sun, S.M., Zhong, N.: On effective convergence of numerical solutions for differential equations. ACM Transactions on Computation Theory **6**(1), Art. 4, 25 (2014)

82. Sun, S.M., Zhong, N.: Computability aspects for 1st-order partial differential equations via characteristics. Theoretical Computer Science **583**, 27–39 (2015)

83. Sun, S.M., Zhong, N., Ziegler, M.: On computability of Navier-Stokes' equation. In: A. Beckmann, V. Mitrana, M. Soskova (eds.) Evolving Computability, *Lecture Notes in Computer Science*, vol. 9136, pp. 334–342. Springer, Cham (2015), 11th Conference on Computability in Europe, CiE 2015, Bucharest, Romania, June 29–July 3, 2015

84. Szőkefalvi-Nagy, B.: Spektraldarstellung linearer Transformationen des Hilbertschen Raumes. Springer (1942)

85. Thies, H.: Uniform computational complexity of ordinary differential equations with applications to dynamical systems and exact real arithmetic. Graduate School of Arts and Sciences, University of Tokyo, Tokyo, Japan (2018)

86. Tucker, W.: A rigorous ODE solver and Smale's 14th problem. Foundations of Computational Mathematics **2**(1), 53–117 (2002)

87. Weihrauch, K.: Computable Analysis. Springer, Berlin (2000)

88. Weihrauch, K., Zhong, N.: The wave propagator is Turing computable. In: J. Wiedermann, P. van Emde Boas, M. Nielsen (eds.) Automata, Languages and Programming, *Lecture Notes in Computer Science*, vol. 1644, pp. 697–706. Springer, Berlin (1999), 26th International Colloquium, ICALP'99, Prague, Czech Republic, July, 1999

89. Weihrauch, K., Zhong, N.: Is the linear Schrödinger propagator Turing computable? In: J. Blanck, V. Brattka, P. Hertling (eds.) Computability and Complexity in Analysis, *Lecture Notes in Computer Science*, vol. 2064, pp. 369–377. Springer, Berlin (2001), 4th International Workshop, CCA 2000, Swansea, UK, September 2000

90. Weihrauch, K., Zhong, N.: Turing computability of a nonlinear Schrödinger propagator. In: J. Wang (ed.) Computing and Combinatorics, *Lecture Notes in Computer Science*, vol. 2108, pp. 596–599. Springer, Berlin (2001), 7th Annual International Conference, COCOON 2001, Guilin, China, August 20-23, 2001

91. Weihrauch, K., Zhong, N.: Is wave propagation computable or can wave computers beat the Turing machine? Proceedings of the London Mathematical Society **85**(2), 312–332 (2002)

92. Weihrauch, K., Zhong, N.: The solution operator of the Korteweg-de Vries equation is computable. In: V. Brattka, M. Schröder, K. Weihrauch (eds.) CCA 2002 Computability and Complexity in Analysis, *Electronic Notes in Theoretical Computer Science*, vol. 66. Elsevier, Amsterdam (2002), 5th International Workshop, CCA 2002, Málaga, Spain, July 12–13, 2002

93. Weihrauch, K., Zhong, N.: Computing the solution of the Korteweg-de Vries equation with arbitrary precision on Turing machines. Theoretical Computer Science **332**(1–3), 337–366 (2005)

94. Weihrauch, K., Zhong, N.: An algorithm for computing fundamental solutions. SIAM Journal on Computing **35**(6), 1283–1294 (2006)

95. Weihrauch, K., Zhong, N.: Computing Schrödinger propagators on Type-2 Turing machines. Journal of Complexity **22**(6), 918–935 (2006)

96. Weihrauch, K., Zhong, N.: Computable analysis of the abstract Cauchy problem in a Banach space and its applications (I). In: D. Cenzer, R. Dillhage, T. Grubba, K. Weihrauch (eds.) Proceedings of the Third International Conference on Computability and Complexity in Analysis, *Electronic Notes in Theoretical Computer Science*, vol. 167, pp. 33–59. Elsevier, Amsterdam (2007), CCA 2006, Gainesville, Florida, USA, November 1–5, 2006

97. Weihrauch, K., Zhong, N.: Computable analysis of the abstract Cauchy problem in a Banach space and its applications I. Mathematical Logic Quarterly **53**(4–5), 511–531 (2007)

98. Zhong, N.: Computability structure of the Sobolev spaces and its applications. Theoretical Computer Science **219**, 487–510 (1999)
99. Zhong, N.: Computable analysis of a non-homogeneous boundary-value problem for the Korteweg-de Vries equation. In: S.B. Cooper, B. Löwe, L. Torenvliet (eds.) New Computational Paradigms, *Lecture Notes in Computer Science*, vol. 3526, pp. 552–561. Springer, Berlin (2005), First Conference on Computability in Europe, CiE 2005, Amsterdam, The Netherlands, June 2005
100. Zhong, N.: Computational unsolvability of domains of attraction of nonlinear systems. Proc. Amer. Math. Soc. **137**(8), 2773–2783 (2009)
101. Zhong, N., Weihrauch, K.: Computability theory of generalized functions. Journal of the Association for Computing Machinery **50**(4), 469–505 (2003)