

An analytic system with a computable hyperbolic sink whose basin of attraction is non-computable

Daniel S. Graça
CEDMES/FCT, Universidade do Algarve, C. Gambelas,
8005-139 Faro, Portugal
& SQIG/Instituto de Telecomunicações, Lisbon, Portugal

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

January 15, 2015

Abstract

In many applications one is interested in finding the stability regions (basins of attraction) of some stationary states (attractors). In this paper we show that one cannot compute, in general, the basins of attraction of even very regular systems, namely analytic systems with hyperbolic asymptotically stable equilibrium points. To prove the main theorems, a new method for embedding a discrete-time system into a continuous-time system is developed.

1 Summary of the paper

In the study of dynamical systems, asymptotically stable equilibrium points and closed orbits are important since they represent stationary or repeatable behavior. In fact, to determine stability regions (basins or domains of attraction) of asymptotically stable equilibrium points is a fundamental problem in nonlinear systems theory with great importance in a number of applications such as in the fields of engineering (electric power system, chemical reactions), ecology, biology, economics, etc. In the late 1960's there was a surge of theoretical studies analyzing properties of such domains. In recent years much effort has been devoted to development of numerical methods for estimation of these domains, which has resulted in numerous numerical algorithms (see e.g. [14], [21]). In contrast, relatively little theoretical work on computability of these domains exists.

It is known that these domains are non-computable in some instances. For example, by “gluing” different dynamics over different regions of the space, one may obtain C^k -systems (even C^∞ -systems) with domains of attraction which encode non-computable problems and which are thus non-computable [22], [33].

On the other hand, it is also known that for hyperbolic rational functions, there are (polynomial-time) algorithms for computing basins of attraction and

their complements (Julia sets) with arbitrary precision [3]; in other words, basins of attraction and Julia sets of hyperbolic rational functions are (polynomial-time) computable.

So the question is, where does the boundary between computability and non-computability of basins of attraction lie? In particular, the question of computability remains open for (analytic) non-rational systems. Analyticity imposes a strong degree of regularity, which is higher than that of C^∞ continuity, since the local behavior of an analytic function determines how it behaves globally; thus no “gluing” of different dynamics in different regions is allowed. Another common requirement of regularity found in dynamical systems theory is hyperbolicity (see e.g. [25]). In short, hyperbolicity requires that near attractors (stationary states), the flow must converge to these attractors at a (at least) uniform rate to avoid pathological behavior due to a convergence which is “too slow”. In particular, if we consider the simplest type of attractors, i.e. asymptotically stable equilibrium points, hyperbolicity implies that near an (hyperbolic) asymptotically stable equilibrium point the flow must converge to this point at a rate which is equal to or greater than $e^{-\lambda t}$, for some contraction rate $\lambda > 0$.

In this paper we show that:

Main Theorem. There is an analytic and computable dynamical system with a computable hyperbolic sink s such that the basin of attraction of s is not computable (technically more precise results are given in Theorem 5 and Corollary 7 below).

Thus our result implies that no algorithmic characterization exists, in general, for a given basin of attraction even if an high degree of regularity (analyticity + hyperbolicity) is imposed on the system and even if only the simplest type of attractors (equilibrium points) is considered.

In the case of discrete-time systems, we prove the result by encoding a well-known non-decidable problem into the basin of attraction of s . In the case of continuous-time systems, we prove the result by embedding a discrete-time system with a non-computable basin of attraction into a continuous-time system. The standard suspension method (see Smale [29], Arnold and Avez [1]) for embedding a discrete-time system into a continuous-time system is not sufficient for our case; we instead develop a new method.

The structure of the paper is as follows. In Section 2 we briefly mention related work and then discuss basic notions from dynamical systems and from computability with real numbers. We also delve into our results in a more precise way (see Theorems 5 and Corollary 7). Then in Section 3 we present two formulations of Theorems 5: one for the discrete-time case and another for the continuous-time case. Each case will be proved separately in Sections 4 and 5. Some proofs become quite technical, so we provide a brief guide in Section 3.1.

2 Introduction

2.1 Related work

Analytic dynamical systems lie between polynomial and C^∞ systems. As noted in the summary, basins of attraction of dynamical systems generated by hyperbolic polynomials are polynomial-time computable. On the other hand, it is shown in [33] that for C^∞ dynamical systems, basins of attraction of hyperbolic sinks can be non-computable. Thus the case of analytic systems is the cutoff point in terms of computability of basins.

By the main theorem of this paper, there is a dynamical system generated by an analytic function f with a hyperbolic sink s such that the basin of attraction of s is non-computable, even though for the corresponding systems generated by any initial finite segment of the Taylor series of f , the basins of s are polynomial-time computable. This seems a bit surprising since it is well known that, in contrast to the C^∞ case, analytic functions enjoy pervasive computability; for example, the sequence $\{f^{(n)}\}$ of derivatives is computable if the computable function f is analytic but may fail to be so if f is only C^∞ . On the other hand, the non-computability in the analytic case cannot be proved as for the C^∞ case, because the proof for the C^∞ case makes crucial use of the fact that a non-constant C^∞ function can take a constant value on a non-empty open subset; such a function cannot be analytic. The idea underlying the proof of the C^∞ case is that, starting with a non-computable open set, one constructs two sequences of C^∞ functions such that one contracts on the open set and the other expands on the complement of the set. Then one glues the two sequences together to produce a C^∞ system such that the non-computable open set gives rise to the desired non-computable basin [33]. This construction will not work for analytic systems, for the local behavior of an analytic function determines how it behaves globally. A completely different construction is developed in this paper in order to show that basins of attraction can be non-computable, even in the case of analytic systems.

2.2 Dynamical systems and hyperbolicity

We recall that there are two broad classes of dynamical systems: discrete-time and continuous-time (for a general definition of dynamical systems, encompassing both cases, see [17]). A discrete-time dynamical system is defined by the iteration of a map $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, while a continuous-time system is defined by an ordinary differential equation (ODE) $x' = f(x)$. Common to both classes of systems is the notion of trajectory: in the discrete-time case, a *trajectory* starting at the point x_0 is the sequence

$$x_0, f(x_0), f(f(x_0)), \dots, f^{[k]}(x_0), \dots$$

where $f^{[k]}$ denotes the k th iterate of f , while in the continuous time case it is the solution to the following initial-value problem

$$\begin{cases} x' = f(x) \\ x(0) = x_0 \end{cases}$$

Trajectories $x(t)$ may converge to some *attractor*. Attractors are invariant sets in the sense that if a trajectory reaches an attractor, it stays there. Given an

attractor A , its *basin of attraction* is the set

$$\{x \in \mathbb{R}^n \mid \text{the trajectory starting at } x \text{ converges to } A \text{ as } t \rightarrow \infty\}$$

Attractors come in different varieties: they can be points, periodic orbits, strange attractors, etc. In this paper we focus on the simplest type of attractors: single points (also called *fixed points*). If there is a neighborhood around a fixed point s which is contained in the basin of attraction of s (i.e. every trajectory starting in this neighborhood will converge to s), then s is called a sink.

Among smooth dynamical systems, hyperbolic systems play a central role. In the case of a sink s , hyperbolicity amounts to requiring a uniform rate of convergence at which every trajectory starting in a neighborhood of s converges to s . More details can be found in [17].

2.3 Introduction to computability over real numbers

Classical Computability. At the heart of computability theory lies the notion of *algorithm*. Roughly speaking, a problem is *computable* if there is an algorithm that solves it. For example, the problem of finding the greatest common divisor of two positive integers is computable since this problem can be solved using Euclid’s algorithm. On the other hand, there are many non-computable problems. For example, Hilbert’s 10th problem is not solvable by any algorithm. But how can we show that no algorithm solves this problem? To answer such questions, the notion of algorithm has to be formalized. The formal notion of algorithm makes use of *Turing machines* (see e.g. [28], [24] for more details), which were introduced in 1936 by Alan Turing [30]. The very simple behavior of a Turing machine (TM for short), which mimics human beings executing an algorithm, and a series of equivalence results between different models of computation led to the following conclusion (*Church-Turing thesis*): problems solvable by algorithms (ordinary computers) are exactly those solvable by Turing machines. Notice that the Church-Turing thesis cannot be proved – it simply formalizes the notion of algorithm – but it is unanimously accepted by the scientific community.

Classical computability is carried out over discrete structures. Formal definitions usually involve strings of symbols (e.g. binary words), but they can be stated equivalently using natural numbers as follows. Let \mathbb{N} , \mathbb{Z} , \mathbb{Q} , and \mathbb{R} denote the set of natural numbers (including 0), integers, rational numbers, and real numbers, respectively.

Definition 1 *A function $f : \mathbb{N}^k \rightarrow \mathbb{N}$ is computable if there is a Turing machine that on input $(x_1, \dots, x_k) \in \mathbb{N}^k$ outputs the value $f(x_1, \dots, x_k)$.*

In many applications, given an input $(x_1, \dots, x_k) \in \mathbb{N}^k$, we are only interested in a “yes/no” output. This can be done, without loss of generality, by assuming that $f(x_1, \dots, x_k) = 1$ is a “yes” answer and $f(x_1, \dots, x_k) = 0$ is a “no” answer.

Computing with real numbers. In order to study computability problems over \mathbb{R}^n , one has to generalize the previous notions. Turing, in his seminal paper [30], already provided an approach: code each real number as its decimal expansion and then carry out computations over this symbolic representation. Although simple, this approach does not work as noted by Turing himself in a

later paper [31], since trivial functions like $x \mapsto 3x$ would not be computable in this setting. This is because the decimal representation does not preserve the topology of the real line: $0.9999\dots$ and $1.0000\dots$ are far from each other if considered as strings of symbols, yet they represent the same real number. Details can be found in [32], [7].

Nevertheless, the previous idea can still be used if we use a representation of real numbers that preserves the topology of the real line. Here lies the foundations for computable analysis, which draws on both computability theory and topology, and allows one to compute over topological spaces in addition to \mathbb{N} . In particular, for \mathbb{R} , among many different equivalent possibilities, we use the following formulation: the function $\phi : \mathbb{N} \rightarrow \mathbb{Q}$ is an *oracle* (also called ρ -name) for a real number x if $|x - \phi(n)| \leq 2^{-n}$ for all $n \in \mathbb{N}$. In other words, ϕ provides rationals which approximate x within any desired precision. We can then define *oracle Turing machines* by giving the TM access to an oracle: at any step of the computation the TM can query the value $\phi(n)$ for any n . In this way one introduces computability with real numbers.

Definition 2 *A number $x \in \mathbb{R}$ is computable if there is a TM which computes an oracle ϕ for x : on input n , the machine outputs $\phi(n)$.*

Intuitively, a real number x is computable if there exists a TM which can compute a rational approximation of x to any desired precision. All familiar real numbers (rational numbers, algebraic numbers, e , π , etc.) are computable. Notice that there are only countably many Turing machines. Thus there are only countably many computable real numbers.

What about computable functions? From the above considerations, one cannot expect that a computable function $f : \mathbb{R} \rightarrow \mathbb{R}$ generates only computable real numbers. What we can expect is that there is an oracle TM such that, given an oracle coding the input, the TM outputs $f(x)$.

Definition 3 *A function $f : \mathbb{R} \rightarrow \mathbb{R}$ is computable if there is an oracle TM M such that if ϕ is an oracle for x , then M^ϕ computes an oracle for $f(x)$: on input n , M^ϕ outputs a rational q such that $|q - f(x)| < 2^{-n}$.*

Thus a function is computable if, given an oracle for the input, one can compute the output to any desired precision. Elementary functions from analysis are computable functions. These notions can be extended in an obvious way to \mathbb{R}^n .

Finally, because we will study the computability of basins of attraction of hyperbolic sinks (which are open sets of \mathbb{R}^n), we need to define computable open subsets of \mathbb{R}^n . This can be done by encoding the set into an oracle (using the bases which generate the upper and lower Fell topology) and showing that there exists a TM which computes this oracle (see [7] for more details). However, for practical reasons, we use instead the following equivalent definition.

Definition 4 *An open set $O \subseteq \mathbb{R}^n$ is computable if the distance function $d_{\mathbb{R}^n \setminus O} : \mathbb{R}^n \rightarrow \mathbb{R}$ is computable, where*

$$d_{\mathbb{R}^n \setminus O}(x) = \inf_{y \in \mathbb{R}^n \setminus O} d(x, y).$$

Intuitively an open set O is computable if O can be generated by a computer with arbitrary precision. In this paper we'll show the following result.

Theorem 5 (main result) *There exists a dynamical system defined by an analytic and computable function f , which admits a computable hyperbolic sink s , such that the basin of attraction of s is not computable.*

We'll prove two versions of this result: one for discrete-time systems and the other for continuous-time systems (Theorems 8 and 9 of Section 3).

Notice that computability does not need to be restricted to \mathbb{N} or \mathbb{R} . Given a topological space X and a suitable coding of elements of X in the form of oracles (which depends on the topology of X), one can define computable elements of X similarly to computable real numbers. Using the same idea, one can further define computable functions $f : X \rightarrow Y$. An important result is the following (cf. [7, Corollary 4.19]).

Proposition 6 *If $x \in X$ and $f : X \rightarrow Y$ are computable, then $f(x)$ is computable.*

In particular, if x is computable, but $f(x)$ is not, then f cannot be computable. Thus, when X is taken to be a set of real functions, Theorem 5 implies the following corollary.

Corollary 7 *There is no algorithm which on input (f, s) , where f is a real analytic function and s is a hyperbolic sink of the dynamical system defined by f , computes the basin of attraction of s .*

In fact, Theorem 5 is stronger than Corollary 7. Corollary 7 ensures that no *single* algorithm can be used to compute the basin of attraction for *any* input (f, s) . This is known as *uniform computability*. A weaker version is *non-uniform computability*. In the latter case, we require again an input (f, s) , with the difference being that different algorithms can be used for different inputs. But Proposition 6 also holds for non-uniform computability and thus a non-uniform version of Corollary 7 also holds.

It should be noted that computational problems about the long-term behavior of dynamical systems have been discussed in [4]. Their notion of computability is, however, quite different from the one used here. They allow the use of infinite precision calculations but require that exact sets are generated. Under this model simple sets like the epigraph of the exponential $E = \{(x, y) \in \mathbb{R}^2 \mid y \geq e^x\}$ are not computable [6]. Thus in this model results do not correspond to computing practice [9].

3 Main results

The main theorem, Theorem 5, has the following two versions:

Theorem 8 (discrete-time case) *There is an analytic and computable function $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ which defines a discrete-time dynamical system with the following properties:*

1. *It has a computable hyperbolic sink s ;*
2. *The basin of attraction of s is not computable.*

Theorem 9 (continuous-time case) *There is an analytic and computable function $g : \mathbb{R}^6 \times (-1, +\infty) \times (-1, +\infty) \rightarrow \mathbb{R}^8$ which defines a continuous-time dynamical system via an ODE $y' = g(y)$ with the following properties:*

1. *It has a computable hyperbolic sink s ;*
2. *The basin of attraction of s is not computable.*

We will prove the two theorems constructively in the sense that the functions f and g are explicitly constructed.

3.1 Road map to results

The proofs will at some point become quite involved. For this reason we provide a brief guide which may help the reader to keep track of results.

In Section 4 we prove Theorem 8. The idea underlying the proof is to encode a non-computable problem into the basin of attraction of s . Thus, if one can compute the basin of attraction of s , then one can algorithmically solve a non-computable problem, a contradiction.

The problem which is encoded is the famous Halting problem. We encode an input x_0 of the Turing machine into a triple in \mathbb{N}^3 , the Turing machine itself into a map $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, and the halting configuration into a hyperbolic sink $s \in \mathbb{N}^3$ (assuming w.l.o.g. that the halting configuration is unique). Thus the problem of determining whether a trajectory starting at x_0 will converge to s is equivalent to the Halting problem and is therefore not computable.

Although conceptually simple, this idea is not so easy to implement in practice. There are several issues to be dealt with. To begin with, the value of $f(x)$ depends upon only a finite amount of information carried by x . Thus one has to extract this information first (using the tools of Section 4.2; the exact procedure is explained in the beginning of Section 4.3.2) and then compute $f(x)$ based on this finite amount of information. Here techniques of interpolation are employed (Section 4.3.2).

Moreover, for technical reasons, f needs to be a contraction near integers, that is,

$$f(B(\alpha, \varepsilon)) \subseteq B(f(\alpha), \varepsilon) \tag{1}$$

for some fixed $\varepsilon > 0$, where $\alpha \in \mathbb{N}^3$ and $B(\alpha, \varepsilon) = \{x \in \mathbb{R}^3 \mid \|x - \alpha\| < \varepsilon\}$. Thus we need to further study how error propagates through f and fine-tune f whenever necessary (using functions defined in Section 4.2) so that f satisfies (1). The details are given in Sections 4.3.2 and 4.4.

The continuous-time case is proved differently. The key idea underlying the proof is to embed the previous discrete-time dynamical system into a continuous-time system such that s is still a hyperbolic sink and its basin of attraction remains unchanged through the embedding. Thus the basin of attraction of s remains non-computable in the continuous-time system. The challenge in this approach is to obtain such an embedding (see Sections 5.1, 5.2 and 5.3 for the construction of an appropriate embedding). Assuming the existence of such an embedding, Theorem 9 is proved in Section 5.6. The remainder of this road map deals with Sections 5.1, 5.2, 5.3, 5.4, and 5.5. The reader may prefer to skip these sections in a first reading.

There is a standard technique for embedding discrete-time systems into continuous-time systems, called *suspension* [1], [29]. This procedure relies on equivalence relations and is ill suited for our purposes. Indeed, if we pick a discrete-time dynamical system defined by an analytic and computable function $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and f admits a hyperbolic sink s , then using the suspension method would yield a continuous-time system $x' = g(x)$ with one or more or all of the following undesirable features:

- (i) g may be non-analytic (discontinuities in $g^{(k)}$, the k th order derivative of g , may occur);
- (ii) g may be non-computable (the procedure is not constructive);
- (iii) s may be part of an attractor, but this attractor may not be a sink (it may be, for example, a cycle);
- (iv) even if s is a fixed point, there is no guarantee that it is hyperbolic (in the continuous-time system).

In this paper, we develop a technique for the embedding that is different from the suspension method. Our idea is based on the techniques developed in [5], [13], [11], [12]. We observe that if a discrete-time system defined by $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is embedded into a continuous-time system $y' = g(y)$, then the trajectory $y(t)$, starting at x_0 , must arrive at the states $f(x_0)$, $f(f(x_0))$, and so on at times $t = 1, 2, \dots$; that is, $y(0) = x_0, y(1) = f(x_0), y(2) = f(f(x_0)), \dots$. This requirement of $y(k) = f^{[k]}(x_0)$ for all $k \in \mathbb{N}$ is what makes the embedding difficult - there are infinitely many conditions, $y(k) = f^{[k]}(x_0)$, to be met. The idea used for tackling the problem is to enlist a second companion system and use, alternatively, one system to guide the trajectory of the other moving forward, satisfying one requirement at a time. Let us look into the details a bit more. To embed the discrete-time dynamical system defined by $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ into a continuous-time system, we design a continuous-time system $y' = g(y)$ that is defined on \mathbb{R}^{2n} . The system has two components y_A and y_B , both of dimension n . The components y_A and y_B both start at value x_0 for $t = 0$. In the time interval $[0, 1/2]$, y_B is unchanged (serving as a memory) with the constant value x_0 and the values of y_A are updated so that $y_A = f(y_B) = f(x_0)$ at $t = 1/2$. In the next half-unit time interval $[1/2, 1]$, y_A serves as a memory taking the constant value $f(x_0)$, while the values of y_B are updated so that $y_B = y_A$ at $t = 1$, i.e. $y_B = f(x_0)$. Thus at $t = 1$, one has $y_A(1) = y_B(1) = f(x_0)$. The procedure is then repeated in the subsequent time intervals $[1, 2], [2, 3], \dots$, which leads to the following desired property:

$$y_A(2) = y_B(2) = f(f(x_0)), \quad y_A(3) = y_B(3) = f(f(f(x_0))), \quad \dots$$

The procedure is graphically depicted in Fig. 1.

The subtlety is to always have at each moment a component which acts as a “memory” of the last value computed (see Section 5.1 for details). Using this technique, one can embed the discrete-time dynamical system defined by $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ into a continuous-time system $y' = g(y)$, where $g : \mathbb{R}^6 \rightarrow \mathbb{R}^6$. Moreover, if f is computable, so is g . Thus the problem (ii) above is solved satisfactorily. However, the remaining problems (i), (iii) and (iv) have yet to be addressed. In particular, because the components of g vary in some open time intervals but take constant values in others, g cannot be analytic.

In this paper, we refine the previous embedding so that it becomes analytic (if we start with an analytic function f); the problem (i) is then subsequently

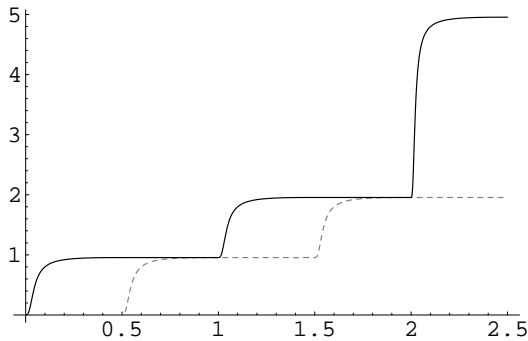


Figure 1: Simulation of the iteration of the map $f(n) = n^2 + 1$ via an ordinary differential equation, where n is initially 0. The solid line represents the variable y_A and the dashed line represents y_B .

solved. We believe that this cannot be done in general, because the propagation of error due to a slight perturbation on x_0 (thus on $f(x_0)$ as well) may alter the behavior of y in the long run, resulting in $y_A(k)$ being far different from $f^{[k]}(x_0)$ for large k 's. Fortunately, in our case, the function f satisfies the inclusion (1) near all integers, which in turn implies that a sufficiently good approximation to x_0 ($f(x_0)$, respectively) can be used to generate a trajectory that is a good enough approximation to the entire trajectory defined by f and starting at x_0 . Using the crucial inclusion (1), one can allow y_A and y_B to be almost constant (but analytic) instead of requiring them to be constant in some time intervals. Of course, we'll then have $y_A(1) \simeq y_B(1) \simeq f(x_0)$, but since the propagation of error through the trajectory is well contained due to (1), we will obtain

$$y_A(2) \simeq y_B(2) \simeq f(f(x_0)), \quad y_A(3) \simeq y_B(3) \simeq f(f(f(x_0))), \dots \quad (2)$$

Moreover the new system $y' = g(y)$ will still be computable and analytic (the construction is presented in Section 5.2). At this point, problems (i) and (ii) are taken care of.

It remains to address problems (iii) and (iv). Notice that if x_0 belongs to the basin of attraction of the discrete-time dynamical system, then the previous embedding only ensures that the (continuous-time) trajectory starting at x_0 will wander around s due to (2). It does not guarantee though that s is a sink nor does it guarantee hyperbolicity. Hyperbolicity of s (assuming that s is indeed a sink) can be shown by demonstrating that the Jacobian of the system at s admits only eigenvalues with negative real part (Section 5.5), thus solving problem (iv). The last problem, problem (iii), is solved by changing the dynamics of $y' = g(y)$ so that all points in $B(s, 1/4)$ will be converging to s and, during the changing process, problems (i), (ii), and (iv) are kept under control and do not resurface. This guarantees that s is a hyperbolic sink and therefore we have our desired embedding. The latter result is proved in Section 5.3. The proof uses various techniques and spans several pages. The reader may prefer to skip it on a first reading. Finally, Section 5.4 shows that the system which satisfies (i)-(iv) still simulates a Turing machine, which guarantees non-computability of the basin of attraction for the hyperbolic sink.

4 The discrete-time case

Recall that Turing machines emulate computer programs. One can always stop the execution of a program and record the value of the variables and the line where the program halted. This recorded information is called a *configuration* in the context of TMs and will be used below. It has all the information we need to continue the computation, if we wish.

Each time we start a new computation, it starts on some special class of configurations – the *initial configurations*. The computation goes on until one eventually reaches a *halting configuration* where the TM stops its execution (it *halts* – which may never happen, i.e. the TM may run forever). Let M be a universal Turing machine (the precise definition of a universal Turing machine is not important here, but can be found in [28]). The Halting problem can be stated as follows: “Given some initial configuration x_0 of M , will the computation reach some halting configuration?” This problem is well known to be non-computable (cf. [28]). Without loss of generality, we may assume that a TM has just one halting configuration (e.g. just before ending, set all variables to 0 and go to some special line with a command `break`; thus the final configuration is unique).

Configurations can be encoded as points of \mathbb{N}^3 (see Section 4.3.1). Since essentially what a TM does is to update one configuration to the next configuration and so on, until the TM halts, associated to each TM M there is a transition function $f_M : \mathbb{N}^3 \rightarrow \mathbb{N}^3$ which describes how M behaves. To prove the main theorem in the discrete-time setting, we need to extend f_M to an analytic function $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ which is non-expanding around the inputs of the transition function f_M , i.e., f maps points near a configuration $x \in \mathbb{N}^3$ to points near $f(x)$.

The simulation of Turing machines with analytic maps was first studied in [19]. A different method was used in [15] to simulate Turing machines with analytic maps which has the advantage of presenting some robustness to perturbations. This robustness property can then be used to simulate Turing machines in continuous time through the use of an ODE [15]. The results shown there guarantee that points near a configuration x will be mapped into some bounded vicinity of $f(x)$, which, of course, does not ensure non-expansiveness. More concretely, it was proven in [15] that for each Turing machine M , there exists an analytic map $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ which simulates it (meaning the restriction of f to \mathbb{N}^3 is the transition function of M) with the following property:

$$\|y - x\| \leq 1/4 \quad \Rightarrow \quad \|f(y) - f(x)\| \leq 1/4 \quad (3)$$

where $x \in \mathbb{N}^3$ codes a configuration. In this paper, the analytic map f needs to be non-expanding in order to have the halting configuration as a sink; thus f must satisfy the following stronger property:

$$\|y - x\| \leq 1/4 \quad \Rightarrow \quad \|f(y) - f(x)\| \leq \lambda \cdot \|y - x\|.$$

for some $\lambda \in (0, 1)$. To achieve this purpose, we adapt the results from [15]. In the remainder of this paper we take

$$\|(x_1, \dots, x_k)\| = \|(x_1, \dots, x_k)\|_\infty = \max_{1 \leq i \leq k} |x_i|.$$

The following result will be proved in Section 4.4.

Theorem 10 *Let M be a Turing machine, and let $f_M : \mathbb{N}^3 \rightarrow \mathbb{N}^3$ be the transition function of M . Then f_M admits an analytic and computable extension $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ with the following property: there exists a constant $\lambda \in (0, 1)$ such that if $x \in \mathbb{N}^3$ is a configuration of M , then for any $y \in \mathbb{R}^3$,*

$$\|x - y\| \leq 1/4 \quad \implies \quad \|f(x) - f(y)\| \leq \lambda \cdot \|x - y\|. \quad (4)$$

In the previous proposition we have assumed that if x is a halting configuration, then $f_M(x) = x$, i.e. x is a fixed point of f .

Lemma 11 *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a map with a fixed point x_0 , and let $B(x_0, r)$ be a neighborhood of x_0 with $r > 0$. If there is a constant $\lambda \in (0, 1)$ such that for all $x \in B(x_0, r)$,*

$$\|f(x) - f(x_0)\| \leq \lambda \|x - x_0\|$$

then x_0 is a hyperbolic sink of f . In particular, no eigenvalue of $Df(x_0)$ has absolute value larger than λ .

Proof. Since f is analytic, we know (see e.g. [20, XVI, §2]) that around the fixed point x_0 ,

$$f(x) = f(x_0) + Df(x_0)(x - x_0) + \|x - x_0\| o(x - x_0) \quad (5)$$

where $o(y) \rightarrow 0$ as $y \rightarrow 0$. Let \vec{v} be any vector in \mathbb{R}^n , and let α be a positive real number. Then the last equation yields (taking $x - x_0 = \alpha\vec{v}$)

$$\begin{aligned} Df(x_0)(\alpha\vec{v}) &= f(x_0 + \alpha\vec{v}) - f(x_0) - \|\alpha\vec{v}\| o(\alpha\vec{v}) \quad \Rightarrow \\ \alpha \|Df(x_0)\vec{v}\| &\leq \|f(x_0) - f(x_0 + \alpha\vec{v})\| + \alpha \|\vec{v}\| \|o(\alpha\vec{v})\| \quad \Rightarrow \\ \|Df(x_0)\vec{v}\| &\leq \frac{\lambda \|\alpha\vec{v}\|}{\alpha} + \|\vec{v}\| \|o(\alpha\vec{v})\| \quad \Rightarrow \\ \|Df(x_0)\vec{v}\| &\leq (\lambda + \|o(\alpha\vec{v})\|) \|\vec{v}\| \end{aligned}$$

The last inequality must be true for all positive α . Since the left-hand side of the inequality does not depend on α , when $\alpha \rightarrow 0$ we get

$$\|Df(x_0)\vec{v}\| \leq \lambda \|\vec{v}\| \quad (6)$$

The last inequality implies that no eigenvalue of $Df(x_0)$ can have absolute value larger than $\lambda < 1$. In particular, this implies that the point x_0 is a hyperbolic sink for the map f . ■

Using these two results, we can now prove Theorem 8.

of Theorem 8. Let M be a universal Turing machine. Suppose, without loss of generality, that M admits only one halting configuration $s \in \mathbb{N}^3$. Since $s \in \mathbb{N}^3$, s is a computable real number. Let f_M be the transition function of M and let f be the map that simulates M according to Theorem 10. Then s is a fixed point of f and, by Theorem 10 and Lemma 11, it is a hyperbolic sink. Denote as W_{final} the basin of attraction of s . Let $x_0 \in \mathbb{N}^3$ be an initial configuration of M . We note that M halts on x_0 iff $x_0 \in W_{final}$. Moreover, from Theorem 10, any trajectory starting at a point in $B(x_0, 1/4) = \{z \in \mathbb{R}^3 \mid \|x_0 - z\| < 1/4\}$ will also converge to s if M halts on the initial configuration x_0 and will not converge to s if M does not halt on x_0 . In other words, $B(x_0, 1/4) \subseteq W_{final}$ if M halts on x_0 and $B(x_0, 1/4) \subseteq \mathbb{R}^3 - W_{final}$ if M does not halt on x_0 .

We recall that W_{final} is an open subset of \mathbb{R}^3 . Now suppose that W_{final} is a computable set. Then the distance function $d_{\mathbb{R}^3 \setminus W_{final}}$ is computable. Therefore we can compute $d_{\mathbb{R}^3 \setminus W_{final}}(x_0)$ with a precision of $1/10$ yielding some rational q . We observe that either $d_{\mathbb{R}^3 \setminus W_{final}}(x_0) = 0$ if $x_0 \notin W_{final}$, or else $d_{\mathbb{R}^3 \setminus W_{final}}(x_0) \geq 1/4$ if $x_0 \in W_{final}$. In the first case, $q \leq 1/10$, while in the second case, $q \geq 1/4 - 1/10 = 3/20$. Now we have an algorithm that solves the halting problem: on initial configuration x_0 , compute $d_{\mathbb{R}^3 \setminus W_{final}}(x_0)$ within a precision of $1/10$ yielding some rational q . If $q \leq 1/10$, then M does not halt on x_0 ; if $q \geq 3/20$, then M halts on x_0 . This is of course a contradiction to the fact that the halting problem is non-computable. Therefore W_{final} cannot be computable. ■

4.1 Brief overview of the proof

From the previous section, it is clear that what remains to be done is to prove Theorem 10. Since the result in this theorem is similar to the simulation result of TMs with analytic maps of [15], it is natural that our proof uses techniques similar to those used in that paper. The main difference is that in each iteration of the map we require that (4) holds for Theorem 10, while in [15] only the weaker condition (3) is required.

The problem in obtaining (4) from (3) is that in the construction used in [15], the error $\|f(x) - f(y)\|$ depends not only on the initial error $\|x - y\|$ but also *on the magnitude of x* . An involved construction was used in [15] to ensure that, despite this dependence, the magnitude of the error $\|f(x) - f(y)\|$ would not exceed $1/4$ (this value is fixed *a priori*).

Here we have to improve the previous construction so that $\|f(x) - f(y)\|$ is not bounded by an error given *a priori*, but rather by the dynamic quantity $\|x - y\|$. This is achieved by introducing a new function ξ_3 (see Section 4.2) which will then be used to modify the problematic step 5 (this step is at the source of the problem mentioned above) in Section 4.4.

4.2 Some special functions used in the construction

We need to construct a map f that satisfies Theorem 10. In particular, f must satisfy condition (4). This can be achieved by requiring the map to “contract” around integer values with the help of several special functions: σ , l_2 , ξ_2 , and ξ_3 . These functions are described below and all of them share the common property of being contractions either around all integers or around some particular integers.

The first special function, $\sigma : \mathbb{R} \rightarrow \mathbb{R}$, is defined by (cf. Fig. 2)

$$\sigma(x) = x - 0.2 \sin(2\pi x) \tag{7}$$

The function σ is a uniform contraction in a neighborhood of integers as the following proposition shows.

Proposition 12 ([15]) *Let $\varepsilon \in [0, 1/4]$. Then there is some contracting factor $\lambda_{1/4} \in (0, 1)$ such that for any $\delta \in [-1/4, 1/4]$, $|\sigma(n + \delta) - n| < \lambda_{1/4}\delta$ for all $n \in \mathbb{Z}$.*

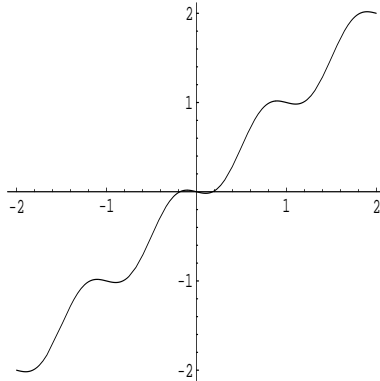


Figure 2: Graphical representation of the function σ .

For instance (see [15]), we can take $\lambda_{1/4} = 0.4\pi - 1 \approx 0.256637$. It follows from the proposition that for any $n \in \mathbb{Z}$, every point $x \in B(n, 1/4)$ will converge to n at a rate of $\lambda_{1/4}$ under the application of σ :

$$|\sigma^{[k]}(x) - n| < \lambda_{1/4}^k \delta$$

where $x = n + \delta$, $|\delta| < 1/4$, and $\sigma^{[k]}(x)$ is the k th iterate of σ . Note that the contraction rate is fixed a priori (it depends on ε) and it is the same for all $n \in \mathbb{Z}$. In some situations, this “fixed a priori” is undesirable, for we may need to specify how fast a point of $B(n, 1/4)$ should converge to n under the application of some special function. This is where the function l_2 enters. However, this comes at a cost: the “dynamic” contraction rate is only valid around the integers 0 and 1. In comparison, σ allows a static contraction rate around every integer $n \in \mathbb{Z}$.

Let us introduce the second special function l_2 . The following result was proved in [15].

Proposition 13 *Let $l_2 : \mathbb{R}^2 \rightarrow \mathbb{R}$ be given by $l_2(x, y) = \frac{1}{\pi} \arctan(4y(x - 1/2)) + \frac{1}{2}$. Suppose also that $a \in \{0, 1\}$. Then*

$$|a - l_2(\bar{a}, y)| < \frac{1}{y}$$

for any $\bar{a}, y \in \mathbb{R}$ satisfying $|a - \bar{a}| \leq 1/4$ and $y > 0$.

Thus, l_2 enjoys “dynamic” contraction rates around the integers 0 and 1. However, it also has an undesirable feature: 0 and 1 are not fixed points of l_2 . For this reason we introduce a third special function ξ_2 , which is built upon l_2 . The function ξ_2 inherits the same “dynamic” contraction rates around 0 and 1 from l_2 , but it has both 0 and 1 as its fixed points, i.e. $\xi_2(0, y) = 0$ and $\xi_2(1, y) = 1$ for $y > 0$.

We now describe the function ξ_2 . It is easy to see that if n is an integer, then

$$0 \leq \frac{\sin^2(\pi(n + \varepsilon))}{4} \leq \varepsilon \quad (8)$$

for $0 \leq \varepsilon < 1/2$. We define $\xi_2 : \mathbb{R} \times \mathbb{R}^+ \rightarrow \mathbb{R}$ by the formula:

$$\xi_2(x, y) = l_2 \left(x, \frac{4y}{\sin^2(\pi x)} \right) \quad (9)$$

where \mathbb{R}^+ is the set of all positive real numbers. For $y > 0$ and $x \in \mathbb{Z}$, $\xi_2(x, y)$ is defined to be the analytic continuation of the right-hand side of (9). It is readily seen that $\xi_2(0, y) = 0$ and $\xi_2(1, y) = 1$ for all $y > 0$.

The last special function to be defined is called ξ_3 , which is built upon ξ_2 . The new function ξ_3 extends the behavior of ξ_2 around 0 and 1 to include the number 2.

Proposition 14 *Let $\xi_3 : \mathbb{R} \times \mathbb{R}^+ \rightarrow \mathbb{R}$ be given by*

$$\xi_3(x, y) = \xi_2((\sigma(x) - 1)^2, 3y) \cdot (2\xi_2(x/2, 3y) - 1) + 1$$

If $a = 0, 1, 2$ and $|a - \bar{a}| \leq \varepsilon \leq 1/4$, then for all $y \geq 2$,

1. If $\bar{a} = a$, then $\xi_3(\bar{a}, y) = a$;
2. If $\bar{a} \neq a$, then

$$|\xi_3(\bar{a}, y) - a| \leq \frac{\varepsilon}{y}$$

Proof. It is easy to verify (1). To prove (2), let us first note $0 < \xi_2(x, y) < 1$ for all $x \in \mathbb{R}$ and $y \in \mathbb{R}^+$. Consider the case where $a = 0$ and $\bar{a} \in [-1/4; 1/4]$ (i.e. $\varepsilon = 1/4$). Then $|(\sigma(\bar{a}) - 1)^2 - 1| < 1/4$ by Proposition 12, and by Proposition 13,

$$1 - 1/y < \xi_2((\sigma(\bar{a}) - 1)^2, y) < 1$$

Similarly, we conclude

$$-1 < 2\xi_2(\bar{a}/2, y) - 1 < -1 + 2/y$$

Since $y \geq 2$, this implies

$$-1 < \xi_2((\sigma(\bar{a}) - 1)^2, y)(2\xi_2(\bar{a}/2, y) - 1) < (1 - 1/y)(-1 + 2/y)$$

or

$$0 < \xi_2((\sigma(\bar{a}) - 1)^2, y)(2\xi_2(\bar{a}/2, y) - 1) + 1 < 3/y$$

Hence, for $a = 0$, $|a - \xi_3(\bar{a}, y)| < 1/y$. The same argument applies to the cases $a = 1$ and $a = 2$. ■

4.3 Simulation of Turing machines with maps

4.3.1 Turing machines

To construct the function f defined in Theorem 10, we have to encode the behavior of a given Turing machine as the iteration of f . To do this, we need to know a bit more about TMs.

This subsection provides a more detailed description of Turing machines. It should be noted though that, to keep this subsection short, the description given here is relatively brief and adapted to the contents of this paper. It is not

intended to give the intuition behind the model nor to explain why it naturally relates to algorithms. The reader interested in such aspects is referred to the excellent introductory textbook [28]. The Definition of TM used here is not (but is equivalent to) the standard definition in the literature.

A Turing machine works on triples (configurations) from $(\Sigma \cup \{B\})^* \times (\Sigma \cup \{B\})^* \times \{1, 2, \dots, m\}$ where Σ is a finite and non-empty set (set of symbols), $B \notin \Sigma$ (B is the blank symbol), $(\Sigma \cup \{B\})^*$ denotes the set of all finite sequences of elements from $\Sigma \cup \{B\}$, and $m \geq 2$ is some integer. For example, if $\Sigma = \{0, 1\}$, then $(0, 1, 1, 0) \in \Sigma^*$, which is usually written as $0110 \in \Sigma^*$. The set $\{1, 2, \dots, m\}$ is the set of states. The initial state is 1 and the halting state is m . An input to a TM is an element $w \in \Sigma^*$; the corresponding initial configuration is $(w, B, 1)$. The Turing machine then updates the configuration

$$(a_n \dots a_0, a_{-1} \dots a_{-k}, q), \quad a_i, a_{-j} \in \Sigma \cup \{B\} \quad (10)$$

according to some fixed rule that depends only on a_0 (symbol read by the head) and the state q . Depending only on the value a_0 and q , the TM will perform the following tasks:

- (i) update the state q to a new state $q' \in \{1, 2, \dots, m\}$ (it may be $q' = q$);
- (ii) change the symbol $a_0 \in \Sigma \cup \{B\}$ to a new symbol $a'_0 \in \Sigma \cup \{B\}$ (it may be $a'_0 = a_0$);
- (iii) after performing step (ii) it may “move the head to left” (cf. [28]) yielding the new configuration

$$(a_n \dots a_1, a'_0 a_{-1} \dots a_{-k}, q')$$

or it may “move the head to right,” yielding the new configuration

$$(a_n \dots a_1 a'_0 a_{-1}, a_{-2} \dots a_{-k}, q')$$

or it may “not move the head,” yielding the configuration

$$(a_n \dots a_1 a'_0, a_{-1} a_{-2} \dots a_{-k}, q')$$

When performing step (iii), if we obtain a sequence of zero length in one of the first two components of the configuration, we replace it by the symbol B .

Configurations are updated step-by-step with these rules until the state eventually reaches the halting state m . In this case, the TM halts; we have reached a halting configuration.

One can code configurations of Turing machines as elements of \mathbb{N}^3 . It suffices to have the number 0 correspond to the symbol B and numbers $\{1, \dots, l\}$ to elements of Σ , where $l = \#\Sigma$ (cardinality of Σ). Then the configuration (10) can be seen as a triple

$$(y_1, y_2, q) \in \mathbb{N}^3 \quad (11)$$

where

$$\begin{aligned} y_1 &= a_0 + a_1 l + \dots + a_n l^n \\ y_2 &= a_{-1} + a_{-2} l + \dots + a_{-k} l^{k-1} \end{aligned} \quad (12)$$

Thus, to define a Turing machine, it suffices to know how to go from one configuration to the next one. In other words, a Turing machine M can be defined by its transition function $f_M : \mathbb{N}^3 \rightarrow \mathbb{N}^3$.

4.3.2 Determining the next action - Interpolation techniques

Let us fix a Turing machine M in the remainder of Section 4. Recall that our ultimate objective (Theorem 10) is to build an analytic and non-expanding map $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that simulates the behavior of the Turing machine M , or in other words, the restriction of f on \mathbb{N}^3 is the transition function f_M of the machine M .

We know that given a configuration (10), the computation of the next configuration only depends on a_0 and q . So, when given a configuration in the form of (11) as an input to f , we have to extract the values of a_0 and q in order to find what M (and therefore f) is supposed to do. The value q is readily available as the last component of (11). However a_0 must be extracted from the first component y_1 .

Extracting the symbol a_0 . Consider an analytic extension $\bar{\omega} : \mathbb{R} \rightarrow \mathbb{R}$ of the function $g : \mathbb{N} \rightarrow \mathbb{N}$ defined by $g(n) = n \bmod l$ (in the case where the tape alphabet of the TM has l symbols). It follows from the coding (12) that $\bar{\omega}(y_1) = a_0$, i.e., $\bar{\omega}$ extracts a_0 from y_1 . We also require $\bar{\omega}$ to be a periodic function, of period l , such that $\bar{\omega}(i) = i$, for $i = 0, 1, \dots, l-1$. The function $\bar{\omega}$ can be constructed by using trigonometric interpolation (cf. [2, pp. 176-182]), which produces an analytic as well as periodic computable function. For example, if $l = 10$, then one can define $\bar{\omega}$ as follows:

$$\bar{\omega}(x) = \alpha_0 + \alpha_5 \cos(\pi x) + \left(\sum_{j=1}^4 \alpha_j \cos\left(\frac{j\pi x}{5}\right) + \beta_j \sin\left(\frac{j\pi x}{5}\right) \right) \quad (13)$$

where

$$\begin{aligned} \alpha_0 &= 9/2, & \alpha_1 &= \alpha_2 = \alpha_3 = \alpha_4 = -1, & \alpha_5 &= -1/2 \\ \beta_1 &= -\sqrt{5 + 2\sqrt{5}}, & \beta_2 &= -\sqrt{1 + \frac{2}{\sqrt{5}}}, & \beta_3 &= -\sqrt{5 - 2\sqrt{5}}, & \beta_4 &= -\sqrt{1 - \frac{2}{\sqrt{5}}} \end{aligned}$$

The construction ensures that, over integer arguments, $\bar{\omega}$ gives exact results; but it does not guarantee that $\bar{\omega}$ is non-expanding around integers, which is needed for condition (4). To meet the non-expanding requirement, we compose $\bar{\omega}$ with the ‘‘uniform contraction’’ function σ as follows: Let K be some integer such that

$$K \geq \max_{x \in [0, l]} |\bar{\omega}'(x)| = \max_{x \in \mathbb{R}} |\bar{\omega}'(x)|$$

where $\bar{\omega}'$ is the derivative of $\bar{\omega}$, and let $k \in \mathbb{N}$ be such that $K\lambda_{1/4}^k \leq 1$, where $\lambda_{1/4}$ is given in Proposition 12 (its exact value is given right after Proposition 12). Then define

$$\omega = \bar{\omega} \circ \sigma^{[k]} \quad (14)$$

It follows that if y_1 is given by (12), then $\omega(y_1) = \bar{\omega} \circ \sigma^{[k]}(y_1) = \bar{\omega}(y_1) = a_0$ for $\sigma^{[k]}(y_1) = y_1$; thus ω extracts a_0 from y_1 . Moreover, ω has the desired non-expanding property as shown below: for any y_1 given by (12) and $y \in \mathbb{R}$,

$$\begin{aligned} |y - y_1| \leq \varepsilon \leq 1/4 &\implies \left| \sigma^{[k]}(y) - y_1 \right| \leq \varepsilon \lambda_{1/4}^k \implies \\ \left| \bar{\omega} \circ \sigma^{[k]}(y) - \bar{\omega}(y_1) \right| \leq \varepsilon K \lambda_{1/4}^k &\implies \left| \omega(y) - \omega(y_1) \right| \leq \varepsilon \end{aligned}$$

Encoding the next action to be performed. After knowing a_0 and q , we now need to encode the next action to be performed by the machine M , i.e. the new symbol to be written in the configuration, the next move to be performed, and the new state. Using Lagrange interpolation we can encode each of these performances by an analytic function.

Let $Q_j, S_i : \mathbb{R} \rightarrow \mathbb{R}$, $0 \leq i \leq l$ and $1 \leq j \leq m$, be the functions defined as follows:

$$Q_j(x) = \prod_{\substack{k=1 \\ k \neq j}}^m \frac{(x-k)}{(j-k)}, \quad S_i(x) = \prod_{\substack{k=0 \\ k \neq i}}^l \frac{(x-k)}{(i-k)}$$

Note that

$$Q_j(x) = \begin{cases} 0, & \text{if } x = 1, \dots, j-1, j+1, \dots, m \\ 1, & \text{if } x = j \end{cases} \quad \text{and}$$

$$S_i(x) = \begin{cases} 0, & \text{if } x = 0, \dots, i-1, i+1, \dots, l \\ 1, & \text{if } x = i \end{cases}$$

Suppose that on symbol i and state j , the state of the next configuration is $q_{i,j}$. Then the state that follows symbol a_0 and state q is given by

$$\overline{q_{next}}(a_0, q) = \sum_{i=0}^l \sum_{j=1}^m S_i(a_0) Q_j(q) q_{i,j} \quad (15)$$

A similar procedure can be used to determine the next symbol to be written and the next move. It is easy to see that $\overline{q_{next}}(a_0, q)$ returns the state of the next configuration if the machine M is in the state q reading the symbol a_0 . But again $\overline{q_{next}}$ may fail to be non-expanding. This problem is dealt with similarly as in the previous case. Let K be some integer such that

$$K \geq \max_{\substack{s \in [-1, l+1] \\ q \in [0, m+1]}} \|\nabla \overline{q_{next}}(s, q)\|_2$$

where $\nabla \overline{q_{next}}$ is the gradient of $\overline{q_{next}}$ and $\|\cdot\|_2$ is the Euclidean norm, and let $k \in \mathbb{N}$ be such that $K \lambda_{1/4}^k \leq 1$ (again $\lambda_{1/4}$ is given by Proposition 12). Now we define $q_{next} : \mathbb{R}^2 \rightarrow \mathbb{R}$ as follows: $q_{next}(x_1, x_2) = \overline{q_{next}}(\sigma^{[k]}(x_1), \sigma^{[k]}(x_2))$ for all $x_1, x_2 \in \mathbb{R}$. The function q_{next} satisfies the following two conditions: for all $i = 0, \dots, l$, $j = 1, \dots, m$, $q_{next}(i, j) = \overline{q_{next}}(\sigma^{[k]}(i), \sigma^{[k]}(j)) = q_{i,j}$ and it is non-expanding as shown below:

$$\|(x_1, x_2) - (i, j)\| \leq \varepsilon \leq 1/4 \implies \|(\sigma^{[k]}(x_1), \sigma^{[k]}(x_2)) - (i, j)\| \leq \varepsilon \lambda_{1/4}^k \implies$$

$$\|\overline{q_{next}}(\sigma^{[k]}(x_1), \sigma^{[k]}(x_2)) - \overline{q_{next}}(i, j)\| \leq \varepsilon K \lambda_{1/4}^k \implies \|q_{next}(x_1, x_2) - q_{next}(i, j)\| \leq \varepsilon$$

4.4 Proof of Theorem 10

We need to construct an analytic and non-expanding map f that simulates the behavior of the machine M (i.e. f is an extension of the transition function f_M of M) and satisfies the condition (4). As a first step, we construct a map $\tilde{f} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that has all the properties of the function f in Theorem 10, except that \tilde{f} satisfies the condition (4) with $\lambda = 1$ rather than the desired $\lambda \in (0, 1)$.

To remedy this deficiency, we make use of the “uniform contraction” function σ again and let $f(x_1, x_2, x_3) = \tilde{f}(\sigma(x_1), \sigma(x_2), \sigma(x_3))$ for all $x = (x_1, x_2, x_3) \in \mathbb{R}^3$. Then f would satisfy the condition (4) with $\lambda = \lambda_{1/4} = 0.4\pi - 1$. Since \tilde{f} and σ are both analytic, so is f . Moreover, if \tilde{f} simulates the machine M , then so does f because $f(n_1, n_2, n_3) = \tilde{f}(\sigma(n_1), \sigma(n_2), \sigma(n_3)) = \tilde{f}(n_1, n_2, n_3)$ for all $(n_1, n_2, n_3) \in \mathbb{N}^3$. Thus f satisfies all conditions of Theorem 10 and the proof of Theorem 10 is then complete.

The remainder of this subsection is devoted to construction of the function \tilde{f} . We assume $0 \leq \varepsilon \leq 1/4$ throughout subsection 4.4.

1. **Extract the symbol a_0 .** Let a_0 be the symbol being actually read by the machine M . Then $\omega(y_1) = a_0$, where ω is given by (14). Moreover, ω is analytic and non-expanding around integers, as we have seen.
2. **Encode the next state.** The map q_{next} returns the next state and is non-expanding around meaningful integer vectors, where an integer vector (s, q) is called meaningful if s codes a symbol and q codes a state.
3. **Encode the symbol to be written on the tape.** Similarly to the state, we can define a map $s_{next} : \mathbb{R}^2 \rightarrow \mathbb{R}$ such that $s_{next}(i, j)$ returns the next symbol to be written on the tape if the machine M is reading symbol i and is in state j . This map is non-expanding around meaningful integer vectors (i, j) .
4. **Encode the direction of the move for the head.** Let h denote the direction of the move of the head, where $h = 0$ indicates a move to the left, $h = 1$ a “no move”, and $h = 2$ a move to the right. Then, similarly to the state, we can define a map $h_{next} : \mathbb{R}^2 \rightarrow \mathbb{R}$ such that $h_{next}(i, j)$ returns the next move to be written on the tape if the machine M is reading symbol i and is in state j . This map is non-expanding around meaningful integer vectors (i, j) .
5. **Update the tape contents.** In the absence of error, the “next value” of y_1 , y_1^{next} , is given by Lagrange interpolation as a function of y_1 , y_2 , and q as follows (to simplify notation, let us use s_{next} and h_{next} to represent $s_{next}(\omega(y_1), q)$ and $h_{next}(\omega(y_1), q)$ respectively):

$$\begin{aligned} \overline{y_1^{next}}(y_1, y_2, s_{next}, h_{next}) &= (l \cdot (y_1 + s_{next} - \omega(y_1)) + \omega(y_2)) \frac{(1 - h_{next})(2 - h_{next})}{2} \\ &\quad + (y_1 + s_{next} - \omega(y_1)) h_{next} (2 - h_{next}) + \frac{y_1 - \omega(y_1)}{l} \frac{h_{next}(1 - h_{next})}{-2} \end{aligned} \tag{16}$$

When the head moves left, doesn’t move, or moves right, the first, second, or third term on the right-hand side gives the “next value” of y_1 , respectively. A function $\overline{y_2^{next}}$ giving the “next value” of y_2 can be constructed in a similar way.

When y_1 and y_2 are given by (12) and q is a state (such y_1 , y_2 , and q are called meaningful integers), $\overline{y_1^{next}}(y_1, y_2, s_{next}, h_{next})$ returns the exact value of y_1 in (12) for the next configuration. Unfortunately, there is a problem with this function: $\overline{y_1^{next}}$ is not only expanding, but also in a way harder to deal with than

in the case of $\overline{q_{next}}$. Here is why: let $A : \mathbb{R}^2 \rightarrow \mathbb{R}$ be the simple product defined by $A(x, y) = x \cdot y$. Then for any $\varepsilon, \delta > 0$, $A(x + \varepsilon, y + \delta) - A(x, y) = \varepsilon\delta + \varepsilon y + \delta x$. Thus, as x and y grow, the difference $A(x + \varepsilon, y + \delta) - A(x, y)$ also grows, even when ε and δ are held constant. Since each term on the right-hand side of (16) is a product containing y_1 , and y_1 can grow arbitrarily large, it follows that the map $\overline{y_1^{next}}$ is expanding around positive integers. Moreover, the rate of expansion is non-uniform because the rate depends on the module of y_1 . Due to this non-uniform expansion nature of $\overline{y_1^{next}}$, in comparison with the deduction of q_{next} from $\overline{q_{next}}$, more work is needed in order to build a non-expanding “next value” function y_1^{next} from $\overline{y_1^{next}}$.

So we need to build a map y_1^{next} satisfying the following two conditions: for any $y_1, y_2, s_{next}, h_{next} \in \mathbb{N}$, $\overline{y_1}, \overline{y_2}, \overline{s_{next}}, \overline{h_{next}} \in \mathbb{R}$ with $\|(y_1, y_2, s_{next}, h_{next}) - (\overline{y_1}, \overline{y_2}, \overline{s_{next}}, \overline{h_{next}})\| < \varepsilon \leq 1/4$,

$$y_1^{next}(y_1, y_2, s_{next}, h_{next}) = \overline{y_1^{next}}(\overline{y_1}, \overline{y_2}, \overline{s_{next}}, \overline{h_{next}})$$

and

$$\|y_1^{next}(y_1, y_2, s_{next}, h_{next}) - \overline{y_1^{next}}(\overline{y_1}, \overline{y_2}, \overline{s_{next}}, \overline{h_{next}})\| \leq \varepsilon \quad (17)$$

that is, y_1^{next} gives the “next value” of y_1 as $\overline{y_1^{next}}$ does and y_1^{next} is non-expanding. To build this map, we use the interpolation idea behind (16), which of course has to be improved for the reasons mentioned above.

Let us write $y_1^{next} = P_{stay} + P_{left} + P_{right}$, where $P_{stay}(y_1, y_2, s_{next}, h_{next})$ is a term that is non-zero only when $h = 1$, i.e. when the head does not move, and its value is the “next value” of y_1 in this case. The other terms P_{left} , and P_{right} are defined similarly. If we can construct the parcels $P_{left}, P_{stay}, P_{right}$ such that

$$\begin{aligned} \|P_{left}(y_1, y_2, s_{next}, h_{next}) - P_{left}(\overline{y_1}, \overline{y_2}, \overline{s_{next}}, \overline{h_{next}})\| &\leq \varepsilon/3 \\ \|P_{stay}(y_1, y_2, s_{next}, h_{next}) - P_{stay}(\overline{y_1}, \overline{y_2}, \overline{s_{next}}, \overline{h_{next}})\| &\leq \varepsilon/3 \\ \|P_{right}(y_1, y_2, s_{next}, h_{next}) - P_{right}(\overline{y_1}, \overline{y_2}, \overline{s_{next}}, \overline{h_{next}})\| &\leq \varepsilon/3 \end{aligned} \quad (18)$$

then (17) is satisfied, i.e., y_1^{next} is non-expanding. Let us show how to obtain a parcel P_{stay} with the above property. The same argument can be applied to the parcels P_{left} and P_{right} .

The parcel P_{stay} is constructed by modifying the term

$$(y_1 + s_{next} - \omega(y_1))h_{next}(2 - h_{next})$$

in (16) (we recall that this term gives the “next value” of y_1 if the head doesn’t move). We intend to define P_{stay} to be the product of two functions $C(\cdot, \cdot)$ and $D(\cdot)$, where $C(y, z) = \sigma^{[2]}(y) + \sigma^{[2]}(z) - \sigma^{[2]}(\omega(y))$ and $D(x) = x(2 - x)$. Since $0 < \varepsilon < 1/4$ and $|(y_1 + s_{next} - \omega(y_1)) - (\overline{y_1} + \overline{s_{next}} - \omega(\overline{y_1}))| \leq 3\varepsilon$, the following holds true:

$$|C(\overline{y_1}, \overline{s_{next}}) - (y_1 + s_{next} - \omega(y_1))| \leq 3\varepsilon\lambda_{1/4}^2 < \frac{\varepsilon}{5}$$

In other words, the function $C(\cdot, \cdot)$ is non-expanding. Thus, to guarantee that the product $C(\cdot, \cdot)D(\cdot)$ is non-expanding, it suffices to find the condition that ensures non-expansiveness of the function D . To this end, we observe that

$$\max_{x \in [-1, 3]} |D'(x)| = 4$$

and it then follows that $|D(w) - D(h_{next})| \leq 4\delta$ provided $|w - h_{next}| \leq \delta \leq 1/4$ (recall that $h_{next} \in \{0, 1, 2\}$). In particular, since $|C(\overline{y_1}, \overline{s_{next}})| \leq \overline{y_1} + l$ and $|D(h_{next})| \leq 1$, the following estimate holds:

$$\begin{aligned} & |C(\overline{y_1}, \overline{s_{next}})D(w) - C(y_1, s_{next})D(h_{next})| \leq \tag{19} \\ & |C(\overline{y_1}, \overline{s_{next}})(D(w) - D(h_{next}))| + |D(h_{next})(C(\overline{y_1}, \overline{s_{next}}) - C(y_1, s_{next}))| \leq \\ & (\overline{y_1} + l)4\delta + \frac{\varepsilon}{5} \end{aligned}$$

Therefore, if we can compute some $w = \theta(\overline{h_{next}}, \overline{y_1})$ satisfying $|w - h_{next}| \leq \delta$ with

$$(\overline{y_1} + l)4\delta \leq \frac{\varepsilon}{10}, \quad \text{or equivalently,} \quad \delta \leq \frac{\varepsilon}{40(\overline{y_1} + l)} \tag{20}$$

then we can define P_{stay} as follows:

$$P_{stay}(\overline{y_1}, \overline{y_2}, \overline{s_{next}}, \overline{h_{next}}) = C(\overline{y_1}, \overline{s_{next}})D(\theta(\overline{h_{next}}, \overline{y_1}))$$

From (19) and (20) it follows that P_{stay} satisfies the second condition of (18); thus P_{stay} is non-expanding.

It therefore remains to define a function θ that has the following property:

$$|\theta(\overline{h_{next}}, \overline{y_1}) - h_{next}| \leq \frac{\varepsilon}{40(\overline{y_1} + l)}$$

From Proposition 14 and the facts that $|\overline{h_{next}} - h_{next}| \leq \varepsilon$ and $h_{next} \in \{0, 1, 2\}$, it becomes clear that we can make use of the special function ξ_3 to get the desired function θ as follows:

$$\theta(\overline{h_{next}}, \overline{y_1}) = \xi_3(\overline{h_{next}}, 40(\overline{y_1} + l))$$

With θ defined as above, P_{stay} is non-extending. Moreover, we observe that

$$P_{stay}(y_1, y_2, s_{next}, h_{next}) = C(y_1, s_{next})D(\theta(h_{next}, y_1)) = \overline{y_1^{next}}(y_1, y_2, s_{next}, h_{next})$$

since $\sigma^{[2]}(n) = n$ for any $n \in \mathbb{N}$ and $\xi_3(h_{next}, 40(y_1 + l)) = h_{next}$; thus the first condition imposed on y_1^{next} is also satisfied.

We now have all the pieces needed to build the map $\tilde{f} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ mentioned at the beginning of this section. It is defined as follows:

$$\begin{aligned} \tilde{f}(y_1, y_2, q) = & (y_1^{next}(y_1, y_2, s_{next}(\omega(y_1), q)), h_{next}(\omega(y_1), q)), \\ & y_2^{next}(y_1, y_2, s_{next}(\omega(y_1), q)), h_{next}(\omega(y_1), q)), \\ & q_{next}(\omega(y_1), q)) \end{aligned}$$

It is also easy to see that \tilde{f} and therefore f are both computable (they are defined by composing standard computable functions and some of their analytic continuations, and therefore are computable; see [26], [7]).

5 The continuous-time case

Now that we have proved Theorem 8, we proceed to prove Theorem 9. To construct the system $y' = g(y)$ of Theorem 9, the idea is to embed the discrete-time system defined in Theorem 8 in a continuous-time system, as we noted in

Section 3.1. We strongly suggest that the reader use Section 3.1 as a guide for the next sections. Section 5.1 is basically material from [5], [13], [11], [12], which is used as a building block for Section 5.2 which is material from [15]. Those sections will provide enough background to prove the completely new results of Sections 5.3, 5.4, and 5.5.

5.1 Simulations of Turing machines with ODEs - non-analytic case

In this subsection we show how to iterate a map from integers to integers with smooth (but non-analytic) ODEs. By a smooth ODE we mean an ODE

$$y' = g(t, y) \quad (21)$$

where g is of class C^k for $1 \leq k \leq \infty$. This construction will be refined later (Subsection 5.2) to include the analytic case. For the non-analytic case, we make use of the construction presented by Branicky in [5], but following the approach of [13], [11], [12], [15], which allows iteration of a map with a smooth ODE. We say that an ODE $y' = g(t, y)$ iterates a map f if $|y(k, y_0) - f^{[k]}(y_0)| < \gamma$, $k \in \mathbb{N}$, for some $\gamma > 0$, where $y(t, y_0)$ is the solution to the initial-value problem $y' = g(t, y)$ and $y(0) = y_0$. Note that if we iterate the map given in Theorem 10 with a smooth ODE, then this ODE simulates a TM simultaneously.

The construction that allows one to iterate a map with a smooth ODE is given below (Construction 17). It is preceded by two auxiliary results (Constructions 15 and 16). For simplicity, the constructions are presented on \mathbb{R} .

The first construction, Construction 15, presents an ODE whose trajectories target a given value at a specified time, whatever the initial states. This is needed later, for example, to update y_A to the targeted value $f(y_B)$ at time $t = 1/2$, as the reader may recall from Section 3.1.

Notation 15 Consider a point $b \in \mathbb{R}$ (the target), some $\gamma > 0$ (the targeting error), and times $t_0 \geq 0$ (departure time) and t_1 (arrival time), with $t_1 > t_0$. Then we obtain an initial-value problem (IVP) defined with an ODE (21), where $g : \mathbb{R}^2 \rightarrow \mathbb{R}$, such that the solution y satisfies

$$|y(t_1) - b| < \gamma \quad (22)$$

independent of the initial condition $y(t_0) \in \mathbb{R}$.

Let $\phi : \mathbb{R} \rightarrow \mathbb{R}_0^+$ be some function satisfying $\int_{t_0}^{t_1} \phi(t) dt > 0$ and consider the following ODE

$$y' = c(b - y)^3 \phi(t) \quad (23)$$

where $c > 0$. We note that $\mathbb{R}_0^+ = [0, \infty)$ is contained in the interval of existence of any solution to (23), regardless of the initial states. There are two cases to consider: (i) $y(t_0) = b$, (ii) $y(t_0) \neq b$. In the first case, the solution is given by $y(t) = b$ for all $t \in \mathbb{R}$ and (22) is trivially satisfied. For the second case, note that (23) is a separable equation, which can be explicitly solved as follows:

$$\begin{aligned} \frac{1}{(b - y(t_1))^2} - \frac{1}{(b - y(t_0))^2} &= 2c \int_{t_0}^{t_1} \phi(t) dt \implies \\ \frac{1}{2c \int_{t_0}^{t_1} \phi(t) dt} &> (b - y(t_1))^2 \end{aligned} \quad (24)$$

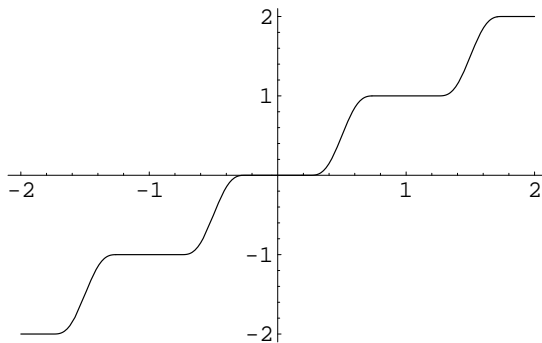


Figure 3: Graphical representation of the function r .

Hence, (22) is satisfied if c satisfies $\gamma^2 \geq (2c \int_{t_0}^{t_1} \phi(t) dt)^{-1}$ i.e., if

$$c \geq \frac{1}{2\gamma^2 \int_{t_0}^{t_1} \phi(t) dt} \quad (25)$$

Notice that, in the construction above, there is an approximation error γ when approaching the target. This error can be removed using the function r defined in the following construction.

Notation 16 We obtain an IVP defined with an ODE (21), where $g : \mathbb{R}^2 \rightarrow \mathbb{R}$, such that the solution r satisfies the condition below (cf. Fig. 3):

$$r(x) = j \quad \text{whenever } x \in [j - 1/4, j + 1/4] \text{ for all } j \in \mathbb{Z} \quad (26)$$

This particular function $r : \mathbb{R} \rightarrow \mathbb{R}$ is needed for the following reason. Suppose that, in Construction 15, $0 < \gamma < 1/4$ and $b \in \mathbb{N}$. Then $r(y(t_1)) = b$, i.e., r corrects the error present in $y(t_1)$ when approaching an integer value b .

Now the construction. First let $\theta_k : \mathbb{R} \rightarrow \mathbb{R}$, $k \in \mathbb{N} - \{0, 1\}$, be the function defined by

$$\theta_k(x) = 0 \text{ if } x \leq 0, \quad \theta_k(x) = x^k \text{ if } x > 0$$

For $k = \infty$ define

$$\theta_k(x) = 0 \text{ if } x \leq 0, \quad \theta_k(x) = e^{-\frac{1}{x}} \text{ if } x > 0$$

These functions can be seen [13] as a C^{k-1} version of Heaviside's step function $\theta(x)$, where $\theta(x) = 1$ for $x \geq 0$ and $\theta(x) = 0$ for $x < 0$.

With the help of θ_k , we define a "step function" $s : \mathbb{R} \rightarrow \mathbb{R}$, which matches the identity function on the integers, as follows:

$$\begin{cases} s'(x) = \lambda_k \theta_k(-\sin 2\pi x) \\ s(0) = 0 \end{cases}$$

where

$$\lambda_k = \frac{1}{\int_{1/2}^1 \theta_k(-\sin 2\pi x) dx} > 0$$

For $x \in [0, 1/2]$, $s(x) = 0$ since $\sin 2\pi x \geq 0$. On $(1/2, 1)$, s strictly increases and satisfies $s(1) = 1$. Using the same argument for $x \in [j, j+1]$, for all integer j , we conclude that $s(x) = j$ whenever $x \in [j, j+1/2]$. Then let $r : \mathbb{N} \rightarrow \mathbb{N}$, $r(x) = s(x+1/4)$. It is easy to see that r satisfies the condition (26). Formally, we should note that, for each $k \in \mathbb{N} \cup \{\infty\} - \{0, 1\}$, we get a different function r , but they all have the same fundamental property (26). So we choose to omit the reference to the index k when defining r (this won't present any problems in later results).

Now that we can reach a target and remove approximation errors, we can iterate maps with ODEs.

Notation 17 Iterate a map $f_M : \mathbb{N} \rightarrow \mathbb{N}$ with a smooth ODE (21).

Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be an arbitrary smooth extension of f_M , and consider the IVP defined with the smooth ODE

$$\begin{cases} z_1' = c_{j,1}(f(r(z_2)) - z_1)^3 \theta_j(\sin 2\pi t) \\ z_2' = c_{j,2}(r(z_1) - z_2)^3 \theta_j(-\sin 2\pi t) \end{cases} \quad (27)$$

and the initial conditions

$$\begin{cases} z_1(0) = x_0 \\ z_2(0) = x_0 \end{cases}$$

where $x_0 \in \mathbb{N}$. Roughly speaking, the special feature concerning system (27) is that one component is “active,” while the other is “dormant” and serves as a memory. Succinctly, in the time interval $[0, 1/2]$, $z_2(t)$ is constant and takes the value x_0 , while the component z_1 is being updated to the value $f_M(x_0)$. In the interval $[1/2, 1]$, the behaviors of z_1 and z_2 are switched: z_2 becomes “active”, while z_1 is “dormant” (taking value $\simeq f_M(x_0)$). In this interval, z_2 will approach the value of z_1 , i.e. $f_M(x_0)$. Therefore, at time $t = 1$, $z_1(t) \simeq f_M(x_0)$ and $z_2(t) \simeq f_M(x_0)$. Then the procedure is repeated for the interval $[1, 2]$ resulting in $z_1(2) \simeq f_M^{[2]}(x_0)$ and $z_2(2) \simeq f_M^{[2]}(x_0)$. Repeating the process, one obtains $z_1(j) \simeq f_M^{[j]}(x_0)$ and $z_2(j) \simeq f_M^{[j]}(x_0)$ for all $j \in \mathbb{N}$ (cf. Fig. 1, where $y_A = z_1$ and $y_B = z_2$).

Let us look at this in more detail. First, we select the parameters used in Construction 15 as follows: $\gamma \leq 1/4$, $t_0 = 0$, $t_1 = 1/2$, $\phi = \phi_1$, where $\phi_1(t) = \theta_j(\sin 2\pi t)$, and $c_{j,1} = c$ given by (25). Using these parameters in (27), $z_2'(t) = 0$ for $t \in [0, 1/2]$; thus the first equation of (27) becomes

$$z_1' = c(b - z_1)^3 \phi(t)$$

where $b = f(x_0) = f_M(x_0)$. It follows from Construction 15 that $|z_1(1/2) - f_M(x_0)| < \gamma \leq 1/4$. Next, for $t \in [1/2, 1]$, $z_1'(t) = 0$, and Construction 16 ensures that $r(z_1(t)) = f_M(x_0)$ (z_1 “remembers” the value of $f_M(x_0)$ for $t \in [1/2, 1]$). If we make use of Construction 15 again with the new set of parameters: $t_0 = 1/2$, $t_1 = 1$, $\phi(t) = \phi_2(t) = \theta_j(-\sin 2\pi t)$, and $c_{j,2} = c$ given by (25), then the second equation of (27) becomes

$$z_2' = c(b - z_2)^3 \phi(t)$$

where $b = f_M(x_0)$. Hence, one has $|z_2(1) - f_M(x_0)| < \gamma \leq 1/4$, which implies that $r(z_2(1)) = f_M(x_0)$. We now continue to the time interval $[1, 2]$. For $t \in [1, 3/2]$, $z_2'(t) = 0$, and Construction 16 ensures that $f(r(z_2(3/2))) = f_M^{[2]}(x_0)$.

Since both $\sin 2\pi t$ and $-\sin 2\pi t$ are periodic with period one, it follows that the above procedure can be repeated for time intervals $[j, j + 1]$, $j \in \mathbb{N}$ (cf. Fig. 1). Moreover, one has, for any given $x_0 \in \mathbb{N}$,

$$r(z_2(t)) = f_M^{[j]}(x_0) \text{ whenever } t \in [j, j + 1/2]$$

for all $j \in \mathbb{N}$. In this sense, (27) simulates the iteration of the function $f_M : \mathbb{N} \rightarrow \mathbb{N}$. Since f is an extension of f_M , we have $f^{[j]}(x_0) = f_M^{[j]}(x_0)$ for any $x_0 \in \mathbb{N}$. For this reason, we also say that (27) simulates the iteration of f (over \mathbb{N}). In the case where f_M is the transition function of a Turing machine M , the ODE system (27) is called a simulation of the machine M .

The construction can be easily extended to the more general case $h : \mathbb{N}^k \rightarrow \mathbb{N}^k$ for $k \geq 1$. We then obtain an ODE with $2k$ equations, where each component h_1, \dots, h_k of h is simulated by a pair of equations.

5.2 Simulations of Turing machines with ODEs - analytic case

The previous section shows how to iterate a map from \mathbb{N} to \mathbb{N} with smooth (but non-analytic) ODEs. In particular, we can iterate the transition function of a given TM with some smooth ODE. However, the ODE built in Construction 17 to iterate the function f is not analytic, even if the function f itself is analytic, for the construction uses the non-analytic functions θ_j in a crucial way. Worse still, it is believed that, in general, one cannot iterate analytic functions with analytic ODEs. However, if an analytic function f satisfies the conditions set in Theorem 10, in particular Condition (4), then we show in this subsection that f can be iterated by an analytic ODE.

The main idea underlying the construction goes as follows. If we want to iterate the transition function $f_M : \mathbb{N}^3 \rightarrow \mathbb{N}^3$ of a Turing machine M with analytic ODEs by using a system similar to (27), we cannot allow z'_1 and z'_2 to be 0 in half-unit time intervals. Instead, we allow them to be very close to zero, which will add some errors to system (27). In general situations, this error may cause the trajectory of the ODE starting at some $x_0 \in \mathbb{N}^3$ to depart from $f_M^{[k]}(x_0)$ in the long run, and thus disqualify the ODE as an iterator of f_M . However, since f , the extension of f_M , satisfies (10), it simulates M robustly in the presence of errors. This allows us to repeat the process arbitrarily many times and still maintain $z_1(j) \simeq f^{[j]}(x_0)$ for all $j \in \mathbb{N}$. Let us re-analyze the constructions of Section 5.1 from this new perspective.

Convention. From now on we fix an analytic function $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that satisfies Theorem 10 and simulates a universal Turing machine M (thus the restriction of f on \mathbb{N}^3 is the transition function f_M of M). We note that the function f also satisfies Theorem 8.

Studying the perturbed targeting equation (cf. Construction 15). Because the iteration procedure relies on the basic ODE (23), we need to study the following perturbed version of (23)

$$z' = c(\bar{b}(t) - z)^3 \phi(t) \tag{28}$$

where $|\bar{b}(t) - b| \leq \rho$. This ‘‘perturbed’’ $\bar{b}(t)$ accounts for the possibility that $z_1(t)$ and $z_2(t)$ may not be fixed in the half-unit time interval where they should be

“dormant.” As in (27), we take the departure time to be $t_0 = 0$, the arrival time to be $t_1 = 1/2$, and $\phi : \mathbb{R} \rightarrow \mathbb{R}_0^+$ satisfying $\int_0^{1/2} \phi(t) dt > 0$, where c satisfies (25) and $\gamma > 0$ is the targeting error without perturbation, that is, $|z(1/2) - b| < \gamma$ if z is the solution to the “unperturbed” equation $z' = c(b - z)^3 \phi(t)$. Let \bar{z} be the solution of this new ODE (28) with initial condition $\bar{z}(0) = \bar{z}_0$ and let z_+, z_- be the solutions of $z' = c(b + \rho - z)^3 \phi(t)$ and $z' = c(b - \rho - z)^3 \phi(t)$, respectively, with initial conditions $z_+(0) = z_-(0) = \bar{z}_0$. Since $\phi(t)$ is a nonnegative function, it is clear that, for all $(t, z) \in \mathbb{R}^2$, the following holds:

$$c(b - \rho - z)^3 \phi(t) \leq c(\bar{b}(t) - z)^3 \phi(t) \leq c(b + \rho - z)^3 \phi(t) \quad (29)$$

From (29) and a standard differential inequality from the basic theory of ODEs (see e.g. [18, Appendix T]), it follows that $z_-(t) \leq \bar{z}(t) \leq z_+(t)$ for all $t \in \mathbb{R}_0^+$. So if we have an upper bound on z_+ and a lower bound on z_- , we immediately get bounds for \bar{z} .

With γ being the targeting error, we have the following estimates from Construction 15:

$$|b + \rho - z_+(1/2)| < \gamma \text{ and } |b - \rho - z_-(1/2)| < \gamma$$

which in turn implies that

$$b - \rho - \gamma < z_-(1/2) \leq \bar{z}(1/2) \leq z_+(1/2) < b + \rho + \gamma$$

or equivalently

$$|\bar{z}(1/2) - b| < \rho + \gamma \quad (30)$$

In other words, the targeting error in the presence of perturbation is at most $\rho + \gamma$, if the target value is perturbed at most up to an amount ρ .

Removing θ_j from (27). We must remove the function θ_j from the right-hand side of (27) as well as in the function r (see Construction 16). Since f is robust to perturbations in the sense that f satisfies Condition (4) of Theorem 10, we no longer need the corrections performed by r . (For simplicity we assume that $f : \mathbb{R} \rightarrow \mathbb{R}$ instead of $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$.) On the other hand, we cannot simply drop the functions $\theta_j(\pm \sin 2\pi t)$ in (27). We need to replace $\phi(t) = \theta_j(\sin 2\pi t)$ by an analytic function $\zeta : \mathbb{R} \rightarrow \mathbb{R}$ with the following ideal behavior:

- (i) ζ has period 1;
- (ii) $\zeta(t) = 0$ for $t \in [1/2, 1]$;
- (iii) $\zeta(t) \geq 0$ for $t \in [0, 1/2]$ and $\int_0^{1/2} \zeta(t) dt > 0$.

Of course, Conditions (ii) and (iii) are incompatible for analytic functions (it is well-known that if a real analytic function is 0 in a non-empty open interval, then it must be 0 everywhere). Instead, we approach ζ using a function ζ_ϵ , $\epsilon > 0$. This function should satisfy the following conditions:

- (ii') $|\zeta_\epsilon(t)| \leq \epsilon$ for $t \in [1/2, 1]$;
- (iii') $\zeta_\epsilon(t) \geq 0$ for $t \in [0, 1/2]$ and $\int_0^{1/2} \zeta_\epsilon(t) dt > I > 0$, where I is independent of ϵ .

Our idea to define such a function ζ_ϵ is to make use of the function l_2 introduced in Proposition 13: Let

$$\zeta_\epsilon(t) = l_2(\vartheta(t), 1/\epsilon) \quad (31)$$

where $\epsilon > 0$ is the precision up to which ζ_ϵ should approximate 0 in the interval $[1/2, 1]$ and $\vartheta : \mathbb{R} \rightarrow \mathbb{R}$ is an analytic periodic function of period 1 satisfying the following conditions:

- (a) $|\vartheta(t)| \leq 1/4$ for $t \in [1/2, 1]$;
- (b) $\vartheta(t) \geq 3/4$ for $t \in (a, b) \subseteq (0, 1/2)$, $a < b$.

We note that Proposition 13 and Condition (a) ensure that $|\zeta_\epsilon(t)| < \epsilon$ for $t \in [1/2, 1]$, and thus ensures (ii'), while Proposition 13 and Condition (b) guarantee that $|\zeta_\epsilon(t)| > 1 - \epsilon$ for $t \in (a, b)$, which in turn implies $\int_0^{1/2} \zeta_\epsilon(t) \geq (1 - \epsilon)(b - a) > 3(b - a)/4$ for $\epsilon < 1/4$, thus (iii') is satisfied. We note that, for all $(t, x) \in \mathbb{R}^2$, $l_2(t, x) > 0$ and thus $\zeta_\epsilon(t) \geq 0$ for all $t \in \mathbb{R}$. It is not difficult to see that one may select $\vartheta : \mathbb{R} \rightarrow \mathbb{R}$ as

$$\vartheta(t) = \frac{1}{2}(\sin^2(2\pi t) + \sin(2\pi t)) \quad (32)$$

since this function satisfies both Conditions (a) and (b) (e.g. $a = 0.16$ and $b = 0.34$). Now we make the replacement: replace $\theta_j(\sin 2\pi t)$ by the analytic function $\zeta_\epsilon(t) = l_2(\vartheta(t), 1/\epsilon)$, where ϑ is given by (32). Similarly, we replace $\theta_j(-\sin 2\pi t)$ by the analytic function $\zeta_\epsilon(-t)$. For this particular function ϑ defined by (32), the constant satisfying (25) may be selected as

$$c \geq \frac{1}{2\gamma^2 \frac{3(0.34-0.16)}{4}}$$

which is independent of ϵ .

Performing Construction 17 with analytic functions. We are now ready to perform a simulation of the Turing machine M (or an iteration of the transition function $f_M : \mathbb{N}^3 \rightarrow \mathbb{N}^3$ of M) with a system similar to (27), but using only analytic functions. For readability, let us assume for now that $f_M : \mathbb{N} \rightarrow \mathbb{N}$ and $f : \mathbb{R} \rightarrow \mathbb{R}$ (instead of $f_M : \mathbb{N}^3 \rightarrow \mathbb{N}^3$ and $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$). Choose a targeting error $\gamma > 0$ and suppose $\epsilon > 0$ is the error resulting from perturbation such that

$$2\gamma \leq \epsilon < 1/8, \quad (33)$$

and consider the following system of ODEs

$$\begin{cases} z_1' = c_1(f \circ \sigma^{[k]}(z_2) - z_1)^3 \zeta_{\epsilon_1}(t) \\ z_2' = c_2(\sigma^{[n]}(z_1) - z_2)^3 \zeta_{\epsilon_2}(-t) \end{cases} \quad (34)$$

with initial conditions $z_1(0) = z_2(0) = \bar{x}_0$, where $|x_0 - \bar{x}_0| \leq \epsilon$, $x_0 \in \mathbb{N}$, and σ is the error-contracting function defined in (7). The four constants c_1, c_2, k , and n and the two functions ϵ_1 and ϵ_2 remain to be defined.

We would like for (34) to satisfy the following property: on $[0, 1/2]$,

$$|z_2'(t)| \leq \gamma \quad (35)$$

This can be achieved by setting $\epsilon_2(t) = \gamma/K(t)$, where $K(t) = c_2^{4/3}(\sigma^{[n]}(z_1) - z_2)^4 + 1$. We note that $|x|^3 \leq x^4 + 1$ for all $x \in \mathbb{R}$ and $\vartheta(-t) < \frac{1}{4}$ for all $t \in [n, n + \frac{1}{2}]$, $n \in \mathbb{N}$. Thus, from Proposition 13,

$$0 \leq \zeta_{\epsilon_2}(-t) = \zeta_{\epsilon_2(t)}(-t) = l_2(\vartheta(-t), 1/\epsilon_2(t)) \leq \epsilon_2(t), \quad t \in [n, n + \frac{1}{2}], \quad n \in \mathbb{N}$$

which further implies that, for all $n \in \mathbb{N}$ and $t \in [n, n + \frac{1}{2}]$,

$$\begin{aligned} |z_2'(t)| &= |c_2(\sigma^{[n]}(z_1) - z_2)^3 \zeta_{\epsilon_2}(-t)| \\ &\leq |c_2(\sigma^{[n]}(z_1) - z_2)^3| \cdot |\epsilon_2(t)| \\ &= |c_2(\sigma^{[n]}(z_1) - z_2)^3| \frac{\gamma}{c_2^{4/3}(\sigma^{[n]}(z_1) - z_2)^4 + 1} \\ &< \gamma \end{aligned}$$

Similarly, if we set $\epsilon_1(t) = \gamma/[c_1^{4/3}(f \circ \sigma^{[n]}(z_2) - z_1)^4 + 1]$, then we have $|z_1'(t)| < \gamma$ for all $t \in [n + \frac{1}{2}, n + 1]$, $n \in \mathbb{N}$. We still need to define the two constants c_1 and c_2 which must satisfy (25). To do so, we first observe that $\epsilon_2(t) \leq \gamma < \frac{1}{16}$ (see (33)) for $t \in [n, n + \frac{1}{2}]$, $n \in \mathbb{N}$. Then it follows from (32) and the discussions immediately preceding and following (32) that

$$\int_0^{1/2} \zeta_{\epsilon_2(t)}(-t) dt > (1 - \epsilon_2(t))(0.34 - 0.16) > (1 - \frac{1}{16})(0.34 - 0.16) > 0.16$$

Thus if we choose c_2 such that

$$c_2 \geq \frac{1}{2\gamma^2(0.16)}$$

then c_2 satisfies (25). We choose c_1 in a similar way. The two integers k and n are chosen such that $k = n$ and $|\sigma^{[k]}(\epsilon + \frac{\gamma}{2})| < \gamma$.

Next we proceed to show that system (34) iterates the function f . First we note that, from the given initial condition $|z_2(0) - x_0| < \epsilon$ with $x_0 \in \mathbb{N}$, it follows from (35) that for all $t \in [0, 1/2]$,

$$|z_2(t) - x_0| \leq |z_2(t) - z_2(0)| + |z_2(0) - x_0| < \gamma t + \epsilon \leq \frac{\gamma}{2} + \epsilon$$

Then from our choice of k , $|\sigma^{[k]}(z_2(t)) - x_0| < \gamma$ for all $t \in [0, 1/2]$, which further implies that $|f \circ \sigma^{[k]}(z_2(t)) - f(x_0)| < \lambda\gamma < \gamma$, $t \in [0, 1/2]$ (see Theorem 10). Now from the study of the perturbed targeting equation (28), if we let $\phi(t) = \zeta_{\epsilon_1}(t)$ and the perturbation error $\rho = \gamma$ in (28), then we reach the conclusion that the solution z_1 exists on $[0, 1/2]$ and

$$|z_1(1/2) - f(x_0)| < 2\gamma \leq \epsilon \tag{36}$$

Proceeding to the next half-time interval $[1/2, 1]$, the roles of z_1 and z_2 are switched. On $[1/2, 1]$, $|z_1'(t)| \leq \gamma$. Then it follows from $|z_1'(t)| \leq \gamma$ and (36) that

$$|z_1(t) - f(x_0)| \leq \epsilon + \gamma/2 \leq 1/4 \text{ for all } t \in [1/2, 1]$$

Thus from the choice of n , $|\sigma^{[n]}(z_1(t)) - f(x_0)| < \gamma$ for all $t \in [1/2, 1]$. Again, by making use of the perturbed targeting equation (28) with $\phi(t) = \zeta_{\epsilon_1}(t)$, we obtain

$$|z_2(1) - f(x_0)| < 2\gamma \leq \epsilon$$

Repeating the above procedure for intervals $[1, 2]$, $[2, 3]$, etc., we conclude that for all $j \in \mathbb{N}$ and for all $t \in [j, j + 1/2]$,

$$|z_1(t) - f^{[j]}(x_0)| \leq \epsilon + \frac{\gamma}{2} \leq 1/4 \tag{37}$$

Moreover, z_1 is defined as the solution of an analytic ODE.

From the construction above, it is easy to see that the following ODE system (38) iterates $f_M : \mathbb{N}^3 \rightarrow \mathbb{N}^3$:

$$\begin{cases} y'_1 = c_1(f_1(\sigma^{[k]}(v_1), \sigma^{[k]}(v_2), \sigma^{[k]}(v_3)) - y_1)^3 \zeta_{\epsilon_1}(t) \\ y'_2 = c_1(f_2(\sigma^{[k]}(v_1), \sigma^{[k]}(v_2), \sigma^{[k]}(v_3)) - y_2)^3 \zeta_{\epsilon_1}(t) \\ y'_3 = c_1(f_3(\sigma^{[k]}(v_1), \sigma^{[k]}(v_2), \sigma^{[k]}(v_3)) - y_3)^3 \zeta_{\epsilon_1}(t) \\ v'_1 = c_2(\sigma^{[n]}(y_1) - v_1)^3 \zeta_{\epsilon_2}(-t) \\ v'_2 = c_2(\sigma^{[n]}(y_2) - v_2)^3 \zeta_{\epsilon_2}(-t) \\ v'_3 = c_2(\sigma^{[n]}(y_3) - v_3)^3 \zeta_{\epsilon_2}(-t) \end{cases} \quad (38)$$

where $(y_1, y_2, y_3), (v_1, v_2, v_3) \in \mathbb{R}^3$ and $f = (f_1, f_2, f_3)$, $f_i : \mathbb{R}^3 \rightarrow \mathbb{R}$, $1 \leq i \leq 3$.

We need to build several new systems similar to (38) but with ζ_{ϵ_1} and ζ_{ϵ_2} replaced by other functions. For simplicity, we will not present their fully expanded forms as in (38), instead we will use systems similar to (34) to represent the corresponding fully expanded systems. When doing so, the expression $(f \circ \sigma)(x)\phi(t)$ is used for $f(\sigma(x_1)\phi(t), \sigma(x_2)\phi(t), \sigma(x_3)\phi(t))$ if $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, $\sigma, \phi : \mathbb{R} \rightarrow \mathbb{R}$, $x \in \mathbb{R}^3$, and $t \in \mathbb{R}$.

5.3 The halting configuration is a sink

In the previous subsection we have shown how to iterate the map f with an analytic ODE on \mathbb{R}^6 . Since f simulates the Turing machine M , so does the ODE. Let us continue to assume that the machine M has a unique halting configuration. To prove Theorem 9, we need to show that this halting configuration of M corresponds to a hyperbolic sink of the analytic ODE simulating M .

Note that the results of Section 5.2 do not guarantee the existence of such a hyperbolic sink. What is shown there is that if $x_0 \in \mathbb{N}^3$ encodes an initial configuration of M and $x_h \in \mathbb{N}^3$ corresponds to the halting configuration of M , then M halts on x_0 iff

$$\|z(0) - (x_0, x_0)\| \leq 1/8 \implies \exists k_0 \in \mathbb{N} \text{ such that } \forall t \geq k_0, \|z(t) - (x_h, x_h)\| \leq 1/4$$

where z is the solution of (34) with the initial condition $z(0) \in \mathbb{R}^6$. In other words, we know that, provided that M halts on x_0 , a trajectory starting sufficiently close to (x_0, x_0) will eventually be in the ball $B((x_h, x_h), 1/4)$, but it may just wander there and never converge to (x_h, x_h) . Since we need to make (x_h, x_h) a sink (hyperbolicity will be dealt with in the next subsection), we have to modify system (34). There are several issues to be addressed.

- In general, an ODE (21) describes a dynamical system if it is autonomous, i.e. if g in (21) does not depend on t (see e.g. [17]), which is not the case for system (34). It is true that (21) can be converted into an autonomous system by writing

$$\begin{cases} y' = g(z, y) \\ z' = 1 \end{cases}$$

But in a system like this, if it has a hyperbolic sink α , the “time” variable z must be finite at α . Such a system cannot simulate a universal Turing machine, for the set of halting times of a universal Turing machine is not bounded. To deal with this problem, an intuitive fix is to introduce a new variable $u = e^{-t}$.

Then $u' = -u$ has an equilibrium point at $u = 0$ and u converges exponentially fast to 0 as $t \rightarrow \infty$. The analytic ODE (34) can then be rewritten as follows:

$$\begin{cases} z'_1 = c_1(f \circ \sigma^{[k]}(z_2) - z_1)^3 \zeta_{\epsilon_1}(-\ln u) \\ z'_2 = c_2(\sigma^{[n]}(z_1) - z_2)^3 \zeta_{\epsilon_2}(\ln u) \\ u' = -u \end{cases} \quad (39)$$

It is clear that the new system (39) still simulates the machine M as system (34) does. However, a new problem occurs with the introduction of u : system (39) has no fixed point, let alone a sink. The problem is obvious: the system is not defined at the only possible equilibrium point $(x_h, x_h, 0)$ since $\zeta_{\epsilon_1}(-\ln u)$ and $\zeta_{\epsilon_2}(\ln u)$ are not defined at $u = 0$. Moreover, the problem cannot be easily solved by extending $\zeta_{\epsilon_1}(-\ln u)$ and $\zeta_{\epsilon_2}(\ln u)$ as $u \rightarrow 0^+$, for both ζ_{ϵ_1} and ζ_{ϵ_2} are periodic functions. The strategy we use to fix this problem is to replace $\zeta_{\epsilon_1}(-\ln u)$ and $\zeta_{\epsilon_2}(\ln u)$ by other functions such that the resulting system not only still simulates the machine M but is also analytic; in particular, it is analytic at $(x_h, x_h, 0)$. More precisely, our idea is to substitute $\zeta_{\epsilon_1}(-\ln u)$ ($\zeta_{\epsilon_2}(\ln u)$ as well) by a function $\chi(u)$ that acts distinctively depending upon whether or not the machine M has halted: as long as M has not yet halted, $\chi(u)$ behaves roughly like $\zeta_{\epsilon_1}(-\ln u)$ (this ensures that the modified system with $\chi(u)$ replacing $\zeta_{\epsilon_1}(-\ln u)$ still simulates the machine M); once M has halted, the behavior of $\chi(u)$ is switched so that $\chi(u)$ is defined as well as analytic at $u = 0$ and the modified system converges to $(x_h, x_h, 0)$ hyperbolically. The switch of behavior of $\chi(u)$ is controlled by a variable in (39) that codes the current state of M in the simulation. For example, one can use the third component v_3 of $z_2 \in \mathbb{R}^3$, which codes the state of the machine M . If the states are given by the values $1, \dots, m$, with m corresponding to the halting state, then the condition $v_3(t) \leq m - 3/4$ implies that the machine M hasn't yet halted, while the condition $v_3(t) \geq m - 1/4$ implies that M has halted (the error of $1/4$ here is due to the perturbation error allowed in the simulation). We note that if M never halts on an input (x_0, x_0) , then system (39) with the initial value $(x_0, x_0, 1)$ will not converge to $(x_h, x_h, 0)$. In this case, it does not matter whether or not the system is defined at $u = 0$ (i.e. $t = \infty$).

- So we want a function $\chi(u)$ to replace $\zeta_{\epsilon_1}(-\ln u)$ that has two distinct behaviors depending upon the value of the variable v_3 . For this purpose, we add a new equation with a new variable τ to system (39) which is defined by

$$\tau' = \frac{d\tau}{dt} = -\alpha u^{\alpha+1} \quad (40)$$

where ideally $\alpha = -1$ before the machine M halts, and $\alpha = 1$ after M halts. Our aim is to make the point $(x_h, x_h, 0, 0)$ a hyperbolic sink of system (39) augmented with equation (40), where x_h is the halting configuration of M . We observe that before M halts, $\tau' = 1$, and so $\tau(t) = t + \tau(0)$ (thus $\tau(u) = -\ln u + \tau(0)$). After M halts, $\tau' = -u^2$; thus τ is defined and analytic at $u = 0$. Thus if we define $\chi(u)$ as follows:

$$\chi(u) = \zeta_{\epsilon_1}(\tau(u))$$

then this $\chi(u)$ will meet our requirement of changing behavior according to whether or not the machine M has halted. Unfortunately, there is a shortfall: defined in such a way, $\chi(u)$ is not analytic due to the jump in α . For this reason,

we modify the definition of α such that $\alpha \simeq -1$ before the machine M halts, $\alpha \simeq 1$ after M halts, and α is analytic in v_3 . Precisely, we define

$$\alpha = \frac{5}{2}\xi_2(v_3 - (m - 1), 10) - \frac{5}{4}$$

(the 10 is rather arbitrary) where v_3 is the third component of $z_2 \in \mathbb{R}^3$ that codes the state of the machine M . Since $\xi_2(\cdot, 10)$ is analytic, so is α . It is not difficult to show (see [15]) that for any $x < \frac{1}{4}$ and $y > 0$, if $|l_2(x, y)| < \frac{1}{y}$, then $|\xi_2(x, y)| < \frac{1}{y}$. But, as a consequence of Proposition 13, one can easily derive that $|l_2(x, y)| < \frac{1}{y}$ for any $x < \frac{1}{4}$ and $y > 0$, which in turn implies that

$$|\xi_2(x, y)| < \frac{1}{y} \text{ for any } x < \frac{1}{4} \text{ and } y > 0 \quad (41)$$

Since we assume that states are coded by integers $1, \dots, m$ with m the halting state, it follows that, until M halts (which may never happen), $0 < \xi_2(v_3 - (m - 1), 10) \leq 1/10$. One cycle after M has halted (this ensures v_3 has enough time to update its value to $\simeq m$), one has $9/10 < \xi_2(v_3 - (m - 1), 10) \leq 1$. Thus

$$\alpha \in \begin{cases} \left(-\frac{5}{4}, -1 \right] & \text{if } M \text{ hasn't halted} \\ \left[1, \frac{5}{4} \right) & \text{one cycle after } M \text{ has halted} \end{cases} \quad (42)$$

• If we simply define τ by (40) with analytic α , we will not be able to show that system (39) augmented with Equation (40) has $(x_h, x_h, 0, 0)$ as a hyperbolic sink. This is because we need to compute the Jacobian of the augmented system at $(x_h, x_h, 0, 0)$, which includes the derivative of $u^{\alpha+1}$ with respect to v_3 , since this variable appears in the expression for α . But $\partial u^{\alpha+1}/\partial v_3 = u^{\alpha+1} \ln u \frac{\partial \alpha}{\partial v_3}$, which is not defined at $u = 0$. For this reason, we have to modify the definition of τ yet again, using the following equation

$$\tau' = 2(u + \xi_2(v_3 - (m - 1), \tau + 1))^{\alpha+1} \xi_2(m - v_3, 10) - \xi_2(v_3 - (m - 1), 10 + 10\tau^2)\tau \quad (43)$$

We note that the right-hand side of (43) is a function of u and τ . Since the derivative of this function with respect to v_3 containing $\ln(u + 1)$ (see the calculation below) and the function $\xi_2(x, y)$ is only defined for $y > 0$, the function in the right-hand side of (43) is defined only for $u > -1$ and $\tau > -1$.

The behavior of τ , defined by (43), is remarkably similar to its previous form defined in (40) as we shall see below. The parcel $(u + \xi_2(v_3 - (m - 1), \tau + 1))^{\alpha+1}$ in the product of the right-hand side of (43) ensures that, in the halting configuration, the derivative of $(u + \xi_2(v_3 - (m - 1), \tau + 1))^\alpha$ with respect to v_3 is well defined, and thus fixes the problem raised above (1 is added to τ for the reason that $\xi_2(x, y)$ is defined only for $y > 0$ but $\tau = 0$ at $(x_h, x_h, 0, 0)$). We recall that ξ_2 is an analytic function on $\mathbb{R} \times \mathbb{R}^+$, thus $\frac{\partial \xi_2}{\partial v_3}$ and $\frac{\partial \alpha}{\partial v_3}$ are well-defined, in particular at the halting configuration (where $v_3 = m$). For the parcel $(u + \xi_2(v_3 - (m - 1), \tau + 1))^\alpha$, we have

$$\begin{aligned} & \frac{\partial}{\partial v_3} [(u + \xi_2(v_3 - (m - 1), \tau + 1))^\alpha] \\ &= (u + \xi_2(v_3 - (m - 1), \tau + 1))^\alpha \ln(u + \xi_2(v_3 - (m - 1), \tau + 1)) \cdot \frac{\partial \alpha}{\partial v_3} \\ &+ \alpha(u + \xi_2(v_3 - (m - 1), \tau + 1))^{\alpha-1} \frac{\partial}{\partial v_3} (u + \xi_2(v_3 - (m - 1), \tau + 1)) \end{aligned}$$

which is also well-defined at the halting configuration, where $u + \xi_2(v_3 - (m - 1), \tau + 1) = u + \xi_2(m - (m - 1), \tau + 1) = u + \xi_2(1, \tau + 1) = u + 1$ and thus $\ln(u + \xi_2(v_3 - (m - 1), \tau + 1)) = \ln(u + 1)$ is well-defined at $u = 0$. The other term ensures the “switching of behavior” according to whether or not M has halted.

• We now have the following modified system

$$\begin{cases} z'_1 = c_1(f \circ \sigma^{[k]}(z_2) - z_1)^3 \zeta_{\epsilon_1}(\tau) \\ z'_2 = c_2(\sigma^{[m]}(z_1) - z_2)^3 \zeta_{\epsilon_2}(-\tau) \\ u' = -u \\ \tau' = 2(u + \xi_2(v_3 - (m - 1), \tau + 1))^{\alpha+1} \xi_2(m - v_3, 10) - \xi_2(v_3 - (m - 1), 10 + 10\tau^2)\tau \end{cases} \quad (44)$$

We note that the modified system is defined on $\mathbb{R}^6 \times (-1, +\infty) \times (-1, +\infty)$.

In the remainder of this subsection we prove that (I) $z_{\text{equilibrium}} = (z_f, z_f, 0, 0)$ is an equilibrium point of this new system, where

$$z_f = (0, 0, m)$$

is the halting configuration of the machine M (i.e. $z_f = x_h$); (II) For any trajectory $(z_1(t), z_2(t), u(t), \tau(t))$ of (44), if there exists a $t_0 > 0$ such that $(z_1(t_0), z_2(t_0), u(t_0), \tau(t_0)) \in U_f$, where

$$U_f = B(z_f, 1/8) \times B(z_f, 1/8) \times (-1, 2) \times (-1, 5)$$

is an open neighborhood of the equilibrium point $(z_f, z_f, 0, 0)$, then $(z_1(t), z_2(t), u(t), \tau(t)) \in U_f$ for all $t \geq t_0$ and $(z_1(t), z_2(t), u(t), \tau(t)) \rightarrow (z_f, z_f, 0, 0)$ as $t \rightarrow \infty$. Combining (I) and (II) we conclude that $(z_f, z_f, 0, 0)$ is a sink of system (44).

To show that (I) holds, that is, $(z_f, z_f, 0, 0)$ is an equilibrium point of system (44), it suffices to observe that $f \circ \sigma^{[k]}(z_f) = f(\sigma^{[k]}(0), \sigma^{[k]}(0), \sigma^{[k]}(m)) = z_f$ (according to our convention stated right before subsection 5.3) and $\xi_2(m - v_3, 10) = \xi_2(0, 10) = 0$ at $z_f = (0, 0, m)$. The first equality follows from (7) and the fact that $f(z_f) = f_M(z_f) = z_f$ (since f extends f_M , f_M is the transition function of M , and $z_f = (0, 0, m)$ is the halting configuration of M). The second equality is derived from (9). Thus (I) holds.

We now consider (II). We need to show that if $(z_1(t_0), z_2(t_0), u(t_0), \tau(t_0)) \in U_f$ for some $t_0 > 0$, then $(z_1(t), z_2(t), u(t), \tau(t)) \in U_f$ for all $t \geq t_0$ and $(z_1(t), z_2(t), u(t), \tau(t)) \rightarrow (z_f, z_f, 0, 0)$ as $t \rightarrow \infty$. Since $u' = -u$, $u(t) = u(0)e^{-t}$. It is clear that $|u(t)| \leq |u(0)|$ for any $t \geq 0$ and $u(t) \rightarrow 0$ as $t \rightarrow \infty$ for any $u(0) \in (-1, 2)$ (thus any $t > 0$ can be used as t_0). However, to prove the rest of (II), the functions ζ_{ϵ_1} and ζ_{ϵ_2} in (44) must be nonzero if $|v_3 - m| < 1/4$ for technical reasons. Thus we need to refine the two functions ζ_{ϵ_1} and ζ_{ϵ_2} yet again. Recall that $\zeta_\epsilon(t) = l_2(\vartheta(t), \frac{1}{\epsilon})$, where $\vartheta(t) = \frac{1}{2}(\sin^2(2\pi t) + \sin(2\pi t))$ (see (31) and (32)). We redefine ζ_{ϵ_1} and ζ_{ϵ_2} by choosing a different function for $\vartheta(t)$; the new $\vartheta(t)$ is defined as follows:

$$\vartheta(t) = \frac{1}{2}(\sin^2(2\pi t) + \sin(2\pi t)) + l_2(v_3 - (m - 1), t + 10)$$

where, as already mentioned, v_3 is the third component of $z_2 \in \mathbb{R}^3$ that codes the state of the machine M . (Again the number 10 is rather arbitrary.) Recall that the states of M are coded by integers $1, 2, \dots, m$ with m being the halting state. For any v_3 satisfying $|v_3 - m| < 1/4$, we have $|v_3 - (m - 1) - 1| < 1/4$,

thus it follows that $|l_2(v_3 - (m-1), t+10) - 1| < \frac{1}{t+10}$ (see Proposition 13). Therefore, $\vartheta(t) \geq -\frac{1}{8} + 1 - \frac{1}{t+10} = \frac{7}{8} - \frac{1}{t+10} \geq \frac{3}{4}$ (note that the minimum of $\frac{1}{2}(\sin^2(2\pi t) + \sin(2\pi t))$ is $-\frac{1}{8}$), which implies that $1 - \vartheta(t) < \frac{1}{4}$, and it follows again from Proposition 13 that $|1 - \zeta_\epsilon| = |1 - l_2(\vartheta(t), \frac{1}{\epsilon})| \leq \epsilon$.

Thus, assuming that $0 < \epsilon_1, \epsilon_2 \leq 1/10$ in ζ_{ϵ_1} and ζ_{ϵ_2} (if this is not the case, for the ‘‘old’’ precision ϵ_i , substitute the improved accuracy $\epsilon_i/(10\epsilon_i + 10)$, then this new precision will be less than ϵ_i and also less than $1/10$), we conclude that ζ_{ϵ_1} and ζ_{ϵ_2} , defined in (31) but using the new function ϑ (and perhaps a new precision ϵ_1 or ϵ_2), will satisfy

$$9/10 \leq \zeta_{\epsilon_1}(\tau), \zeta_{\epsilon_2}(\tau) < 1$$

In particular

$$\frac{9}{10} \leq \zeta_{\epsilon_1}(\tau)|_{z_{\text{equilibrium}}}, \zeta_{\epsilon_2}(\tau)|_{z_{\text{equilibrium}}} < 1 \quad (45)$$

We are now ready to prove the rest of (II). Let $(z_1(t), z_2(t), u(t), \tau(t))$ be a trajectory of (44) such that $z_1(t), z_2(t) \in B(z_f, 1/8)$ for some $t_0 > 0$. Then from the subsection Performing Construction 17 with Analytic Functions above, it follows that $z_1(t), z_2(t) \in B(z_f, 1/4)$ for all $t \geq t_0$. In the following we show that, for any $0 < \delta \leq 1/4$, if $z_1, z_2 \in B(z_f, \delta)$ for all $t \geq t_\delta$ for some $t_\delta > 0$, then there exists some $t_{\delta/2} > t_\delta$ such that $z_1, z_2 \in B(z_f, \frac{\delta}{2})$ for all $t \geq t_{\delta/2}$. As a consequence, it is readily seen that the part of (II) concerning z_1 and z_2 is true.

Let us prove the result for z_1 . The same argument applies to z_2 . Assume that $z_1, z_2 \in B(z_f, \delta)$ for all $t \geq t_\delta$ for some $t_\delta > 0$. First we recall that, from Proposition 12, if $z_2 \in B(z_f, \delta)$, then $|\sigma^{[k]}(z_2) - z_f| < \lambda_{1/4}^k \delta < \frac{\delta}{3}$, which further implies that

$$|f(\sigma^{[k]}(z_2)) - z_f| = |f(\sigma^{[k]}(z_2)) - f(z_f)| < \lambda_{1/4}^k \delta < \frac{\delta}{3} \quad (46)$$

(see Theorem 10). Let us denote $z_1 = (y_1, y_2, y_3)$, $z_2 = (v_1, v_2, v_3)$, and $f = (f_1, f_2, f_3)$, where $y_i, v_i \in \mathbb{R}$ and $f_i : \mathbb{R}^3 \rightarrow \mathbb{R}$, $i = 1, 2, 3$. Without loss of generality we prove the result component-wise for the first component, that is, we show that there is a $t_{\delta/2} > t_\delta$ such that $|y_1(t)| < \frac{\delta}{2}$ for all $t > t_{\delta/2}$ (recall that $z_f = (0, 0, m)$ and thus $|f_1(\sigma^{[k]}(z_2))| < \frac{\delta}{3}$ for all $t > t_\delta$ by (46) and the assumption). There are two cases to be considered. Case 1: If there exists a time $\tilde{t} > t_\delta$ such that $y_1(\tilde{t}) \in B(0, \delta/2)$, then we only need to show that $y_1(t) \in B(0, \delta/2)$ for all $t > \tilde{t}$. For any $t > \tilde{t}$, if $y_1 = y_1(t) > \frac{\delta}{3}$, then $f_1(\sigma^{[k]}(z_2)) - y_1 < 0$ and so $y_1' = c_1(f_1(\sigma^{[k]}(z_2)) - y_1)^3 \zeta_{\epsilon_1} < 0$ for $\zeta_{\epsilon_1} > \frac{9}{10}$. Thus y_1 will be decreasing until it reaches the value $f_1(\sigma^{[k]}(z_2))$. In other words, for any $t > \tilde{t}$, y_1 cannot surpass $\frac{\delta}{2}$. Similarly, if $y_1 = y_1(t) < -\frac{\delta}{3}$, then $f_1(\sigma^{[k]}(z_2)) - y_1 > 0$, which further implies that $y_1' = c_1(f_1(\sigma^{[k]}(z_2)) - y_1)^3 \zeta_{\epsilon_1} > 0$. Therefore y_1 will be increasing until it reaches $f_1(\sigma^{[k]}(z_2))$, which implies that y_1 cannot decrease below $-\frac{\delta}{2}$. We have now proved that $y_1(t) \in B(0, \frac{\delta}{2})$ for all $t > t_{\delta/2}$ with $t_{\delta/2} = \tilde{t}$. Case 2: If for all $t > t_\delta$, $y_1 \notin B(0, \frac{\delta}{2})$. Without loss of generality, let us assume that there is some $t > t_\delta$ such that $y_1(t) > \frac{\delta}{2}$. Then, since $|f_1(\sigma^{[k]}(z_2))| < \frac{\delta}{3}$ for all $t > t_\delta$, we have $y_1' = c_1(f_1(\sigma^{[k]}(z_2)) - y_1)^3 \zeta_{\epsilon_1} < 0$. It follows that y_1 will be decreasing, passing the value $\frac{\delta}{2}$, until it reaches $f_1(\sigma^{[k]}(z_2))$. In other words, there will be a time \tilde{t} such that $y_1(\tilde{t}) \in B(0, \frac{\delta}{2})$. Then it follows from the first

case that for any $t > \tilde{t}$, $y_1(t) \in B(0, \frac{\delta}{2})$. This is a contradiction. Therefore, the second case is invalid.

It remains to show that if there exists some $t_0 > 0$ such that $(z_1(t_0), z_2(t_0), u(t_0), \tau(t_0)) \in U_f$, then $\tau(t) \in (-1, 5)$ for all $t > t_0$ and $\tau(t) \rightarrow 0$ as $t \rightarrow \infty$. We recall that $\tau' = 2(u + \xi_2(v_3 - (m-1), \tau + 1))^{\alpha+1} \xi_2(m - v_3, 10) - \xi_2(v_3 - (m-1), 10 + 10\tau^2)\tau$. On the other hand, by the assumption and the proof in the previous paragraph, we have $v_3 \rightarrow m$ as $t \rightarrow \infty$. Thus, as $t \rightarrow \infty$, $\xi_2(m - v_3, 10) \rightarrow \xi_2(0, 10) = 0$ and $|\xi_2(v_3 - (m-1), 10 + 10\tau^2) - 1| < \frac{1}{10+10\tau^2}$ for sufficiently large t satisfying $|(v_3(t) - (m-1)) - 1| < \frac{1}{4}$ (see Proposition 13), or in other words, for sufficiently large t

$$1 - \frac{1}{10} \leq 1 - \frac{1}{10 + 10\tau^2} < \xi_2(v_3 - (m-1), 10 + 10\tau^2) < 1 + \frac{1}{10 + 10\tau^2} \leq 1 + \frac{1}{10}$$

Therefore, for sufficiently large t ,

$$\tau' \simeq -a\tau \text{ for some positive number } a$$

which implies that the part of (II) concerning τ is true. The proof for (II) is then complete. (We note that the time t_0 in the assumption $(z_1(t_0), z_2(t_0), u(t_0), \tau(t_0)) \in U_f$ may be different in the previous proofs for the three parts of (II) concerning u , z_1 and z_2 , and τ . It suffices to pick the maximum t_0 among the three.)

We have thus shown that the equilibrium point $(z_f, z_f, 0, 0)$ is indeed a sink of system (44). In subsection 5.5 we will show that $(z_f, z_f, 0, 0)$ is a hyperbolic sink.

5.4 The system (44) still simulates the Turing machine M

In this subsection we show that system (44) with the (fixed) initial conditions $u(0) = 1$ and $\tau(0) = 4$ will still simulate the machine M before M halts, including the case that M never halts. We only need to study the system for $t \geq 0$.

First we show that if M hasn't yet halted, then

$$1 < \tau'(t) \leq 5e^{\frac{5}{4}t} \quad (47)$$

for all $t > 0$. If M hasn't yet halted, then it follows from (42) that $\alpha \in (-\frac{5}{4}, -1]$. Moreover, since M has not halted, $m - v_3 \geq 3/4$, which leads to $1 - (m - v_3) \leq 1/4$, and thus $\xi_2(m - v_3, 10) \in [9/10, 1)$ by Proposition 13. Knowing that $\xi_2(v_3 - (m-1), \tau + 1) > 0$, $0 < \xi_2(x, y) < 1$ for all $x \in \mathbb{R}$ and $y > 0$ (by definition of ξ_2), $u(t) = e^{-t}$ (since $u' = -u$ and $u(0) = 1$) and $\tau > -1$, it follows from (43) that

$$\begin{aligned} \tau' &\leq \frac{2(u + \xi_2(v_3 - (m-1), 1 + \tau))\xi_2(m - v_3, 10)}{(u + \xi_2(v_3 - (m-1), 1 + \tau))^{-\alpha}} + \xi_2(v_3 - (m-1), 10 + 10\tau^2) \\ &< \frac{2(1+1)}{u^{-\alpha}} + 1 \leq \frac{2}{u^{\frac{5}{4}}} + 1 = 4e^{\frac{5}{4}t} + 1 \leq 5e^{\frac{5}{4}t} \end{aligned}$$

Next we show that $\tau'(t) > 1$ for $t > 0$. It can be verified easily that under the assumption that $\tau(0) = 4$, $\tau(t) > 0$ for all $t > 0$. Then from (41) it follows

that $1/\tau \geq 1/(\tau + 1) \geq \xi_2(v_3 - (m - 1), \tau + 1) > 0$ and $1/\tau^2 \geq \xi_2(v_3 - (m - 1), 10 + 10\tau^2) > 0$. Using (43) again, we get

$$\begin{aligned} \tau' &= \frac{2\xi_2(m - v_3, 10)}{(u + \xi_2(v_3 - (m - 1), 1 + \tau))^{-\alpha-1}} - \xi_2(v_3 - (m - 1), 10 + 10\tau^2)\tau \\ &\geq \frac{2\frac{9}{10}}{(u + \frac{1}{\tau})^{-\alpha-1}} - \frac{1}{\tau^2}\tau \geq \frac{9/5}{(u + \frac{1}{\tau})^{-\alpha-1}} - \frac{1}{\tau} \geq \frac{9/5}{(1 + \frac{1}{\tau})^{1/4}} - \frac{1}{\tau} \geq \frac{9/5}{1 + \frac{1}{\tau}} - \frac{1}{\tau} \end{aligned}$$

We note that the last expression in the above estimate is an increasing function of τ . Since $\tau(0) = 4$, one has

$$\tau'(0) > \frac{9/5}{1 + \frac{1}{4}} - \frac{1}{4} = \frac{36}{25} - \frac{1}{4} > 1$$

We conclude that $\tau'(t) > 1$ for all $t > 0$. From (47) we know that, before M halts, τ grows with $\tau'(t) > 1$, which implies that $\tau \rightarrow +\infty$ as $t \rightarrow +\infty$ if M never halts.

The next question is whether the simulation of M is still being faithfully carried out by the modified system (44) until M halts. We recall that the machine M is simulated by system (34) in the sense that $|z_1(j) - f^{[j]}(x_0)| \leq \eta$ and $|z_2(j) - f^{[j]}(x_0)| \leq \eta$ for some $0 < \eta < 1/4$ and all $j \in \mathbb{N}$ (see (36) and (37)), where $x_0 \in \mathbb{N}^3$ codes the initial configuration. Thus the time needed to complete a cycle (i.e. from one configuration to the next) is one unit. Now with τ replacing t as input to the ‘‘clocking functions’’ ζ_{ε_1} and ζ_{ε_2} in the modified system (44), we face a new problem: Since τ grows at a faster rate than t , $1 < \tau'(t) \leq 5e^{\frac{5}{4}t}$, the time needed to complete a cycle becomes shorter. So we need to analyze the effect of this speeded-up phenomenon.

As we have seen in Subsections 5.1 and 5.2, in each iteration ‘‘cycle,’’ $[j, j+1]$, of system (34), one of the two variables, z_1, z_2 , is ‘‘dormant’’ while the other is active and is being updated within a $\frac{1}{4}$ -vicinity of the target $f^{[j]}(x_0)$ during the time period $[j, j + \frac{1}{2}]$; then, during the later half $[j + \frac{1}{2}, j + 1]$ of the cycle, the dormant variable becomes active and is being updated within the $\frac{1}{4}$ -vicinity of the same target while the active one stays put. The only problem that may arise with a dormant variable is when the cycle is too long and thus the dormant may accumulate too much error. However, since we now have shorter cycles, this problem won’t occur. The problem here is of another type. With system (44), the iteration cycles no longer have the uniform length $(j+1) - j = 1$, $j \in \mathbb{N}$, as in (34), but become shorter and shorter as $t \rightarrow \infty$ (possibly exponentially shorter in the worst-case scenario) because τ moves faster than t does as (47) shows. Thus there may not be enough time to update the active variable within a $\frac{1}{4}$ -vicinity of the target. Let us look at this in more detail. We note that, in system (34), the reason that z_1 and z_2 can be updated successively on all intervals $[j, j + 1]$ (until M halts) is because both functions ζ_{ε_1} and ζ_{ε_2} , as functions of t , have period 1; thus it is possible to pick a constant c_1 such that, for all $j \in \mathbb{N}$, the target-estimate

$$\frac{1}{16} \geq \frac{1}{2c_1 \int_0^{1/2} \zeta_{\varepsilon_1}(t) dt} > (f^{[j]}(x_0) - z_1(j + 1/2))^2$$

holds (see (24)), which in turn implies that $|f^{[j]}(x_0) - z_1(j + 1/2)| \leq \frac{1}{4}$ for all j . (Similarly one can select a constant c_2 satisfying the target-estimate for

z_2 .) Now with the speeded-up system, the half-cycles $[T_j, T_{j+1}]$ may decrease without a lower bound, and thus it becomes impossible to pick a constant c such that the left-hand side of the target-estimate

$$\frac{1}{16} \geq \frac{1}{2c \int_{T_j}^{T_{j+1}} \zeta_{\epsilon_1}(\tau(t)) dt} > (f^{[j]}(x_0) - z_1(T_{j+1}))^2 \quad (48)$$

holds for all j . We note that the right-hand side of the above inequality is always true (see the derivation of (24)). Of course, we cannot select a constant c_j for each j . Instead, we solve the problem by multiplying $\phi(t)$ by a function $\chi(t)$ so that

$$\int_{t_0}^{t_1} \phi(t)\chi(t) dt \geq \int_0^{1/2} \phi(t) dt \quad (49)$$

where $\phi(t)$ corresponds to ζ_{ϵ_1} or ζ_{ϵ_2} in system (44) and $[t_0, t_1]$ is an arbitrary half-cycle, $0 < t_1 - t_0 < \frac{1}{2}$. The underlying idea here is that, to compensate for the loss in time, we increase the magnitude of the function $\phi(t)$ to $\phi(t)\chi(t)$ so that the integrals $\int_{t_0}^{t_1} \phi(t)\chi(t) dt$ are uniformly bounded below by $\int_0^{1/2} \phi(t) dt$ for all cycles; thus the target-estimate (48) would hold for all cycles with the constant c being c_1 or c_2 as in system (34). Or more intuitively, one may notice that the function $\phi(t)$ is a function with periodic pulses which switch between $\simeq 0$ and $\simeq 1$. The value of $\int_j^{j+1/2} \phi(t) dt$, $j \in \mathbb{N}$, is the area under the j th active pulse of $\phi(t)$. The problem with $\phi(\tau(t))$ is that the durations of the pulses get shorter as t increases. But we would like to continue to use $\int_0^{1/2} \phi(t) dt$ as a lower bound for the area under each active pulse in the speeded-up system. In order to achieve this, we simply use the pulses with increasing heights; that is, if the duration of each pulse decreases by at most a factor of $\chi(t)$, then we increase the height of that pulse accordingly and thus maintain the area under each pulse at the level of at least $\int_0^{1/2} \phi(t) dt$.

Now the details. To obtain $\chi(t)$, we need to know how small $t_1 - t_0$ can be. Recall from (47) that $\tau'(t) \leq 5e^{\frac{5}{4}t}$. Since $\tau(t_1) - \tau(t_0) = 1/2$ (ζ_{ϵ_2} switches from $\simeq 1$ to $\simeq 0$ and vice-versa when its argument increments by $1/2$), one has

$$\begin{aligned} \frac{1}{2} = \tau(t_1) - \tau(t_0) &= \int_{t_0}^{t_1} \tau'(t) dt \leq 5 \int_{t_0}^{t_1} e^{\frac{5}{4}t} dt = 4 \left(e^{\frac{5}{4}t_1} - e^{\frac{5}{4}t_0} \right) \Rightarrow \\ 4e^{\frac{5}{4}t_0} \left(e^{\frac{5}{4}\Delta t} - 1 \right) &\geq \frac{1}{2} \Rightarrow \Delta t \geq \frac{4}{5} \ln \left(1 + \frac{1}{8} e^{-\frac{5}{4}t_0} \right) \end{aligned}$$

where $\Delta t = t_1 - t_0$. It can be proved that

$$a(t) = \frac{4}{5} \ln \left(1 + \frac{1}{8} e^{-\frac{5}{4}t} \right) > \frac{e^{-\frac{5}{4}t}}{15} = b(t)$$

for $t \geq 0$ (by showing that $a(0) - b(0) > 0$, $\lim_{t \rightarrow \infty} \frac{a(t)}{b(t)} = \frac{3}{2} > 1$, and $a(t) - b(t)$ has a unique critical point on $(0, \infty)$ at $t = \frac{4 \ln 2}{5}$, which gives the maximum $\frac{4}{5} \ln \frac{17}{16} - \frac{1}{30}$). We omit these straightforward calculations. Hence

$$\Delta t > \frac{e^{-\frac{5}{4}t}}{15}$$

Therefore it is sufficient to multiply ϕ (i.e. ζ_{ϵ_1} and ζ_{ϵ_2}) by $\chi(t) = 15e^{\frac{5}{4}t}$ to obtain (49). However, since we desire an autonomous system and t is replaced by τ in (44), we need to express $\chi(t)$ in terms of τ . This can easily be done by replacing t with τ : from (47) and $\tau(0) = 4$ (see the beginning of this section), one easily concludes that one can take

$$\chi(\tau) = 15e^{2\tau} \quad (50)$$

since $\chi(\tau) \geq 15e^{\frac{5}{4}t}$. Because both ζ_{ϵ_1} and ζ_{ϵ_2} are multiplied by χ , the error present in z'_1 and z'_2 when these variables are dormant is also multiplied by the same factor. To cancel this effect, we need to get ζ_{ϵ_1} and ζ_{ϵ_2} closer to 0 in the same proportion. This is done by using $\frac{\zeta_{\epsilon_1}}{\chi(\tau)}$ and $\frac{\zeta_{\epsilon_2}}{\chi(\tau)}$ in (44) instead of ζ_{ϵ_1} and ζ_{ϵ_2} . After replacing ζ_{ϵ_1} and ζ_{ϵ_2} by $\chi\frac{\zeta_{\epsilon_1}}{\chi(\tau)}$ and $\chi\frac{\zeta_{\epsilon_2}}{\chi(\tau)}$ in system (44), it is not difficult to see that the system now simulates the machine M , using Construction 17. We also note that $(z_f, z_f, 0, 0)$ remains a sink of system (44). For the sake of readability, we will denote $\frac{\zeta_{\epsilon_1}}{\chi(t)}$ and $\frac{\zeta_{\epsilon_2}}{\chi(t)}$ simply as ζ_{ϵ_1} and ζ_{ϵ_2} , respectively.

5.5 The sink is hyperbolic

Now that we have shown that $z_{equilibrium}$ is a sink of system (44), it remains to prove that $z_{equilibrium}$ is hyperbolic. It suffices to show that the Jacobian of (44) at $z_{equilibrium}$ only admits eigenvalues with negative real parts.

However there is yet another problem with our system (44). We note that the first two equations in system (44) rely on a certain type of targeting equations (see (23)), which in essence can be reduced to

$$z' = -z^3 \quad (51)$$

and for which the equilibrium point 0 is not hyperbolic. Therefore, instead of choosing an equation of type (23) or, more generally, of type (28), we choose an equation with the format

$$z' = (c(\bar{b}(t) - z)^3 + (\bar{b}(t) - z)) \phi(t)$$

Since $\bar{b}(t) - z$ always has the same sign as $c(\bar{b}(t) - z)^3$, it is not difficult to see that adding the term $(\bar{b}(t) - z)$ will not alter the constructions of the previous sections. Thus the following (final) system

$$\begin{cases} z'_1 = (c_1(f \circ \sigma^{[k]}(z_2) - z_1)^3 + (f \circ \sigma^{[k]}(z_2) - z_1))\chi(\tau)\zeta_{\epsilon_1}(\tau) \\ z'_2 = (c_2(\sigma^{[n]}(z_1) - z_2)^3 + (\sigma^{[n]}(z_1) - z_2))\chi(\tau)\zeta_{\epsilon_2}(-\tau) \\ u' = -u \\ \tau' = 2(u + \xi_2(v_3 - (m-1), \tau+1))^\alpha \xi_2(m - v_3, 10) - \xi_2(v_3 - (m-1), 10 + 10\tau^2)\tau \end{cases} \quad (52)$$

will still simulate our universal Turing machine M and have $z_{equilibrium}$ as a sink as the previous system (44) does.

Now we show that that $z_{equilibrium}$ is a hyperbolic sink of system (52) by computing the eigenvalues of the Jacobian of (52) at $z_{equilibrium}$. Note that in (52) the components z_1 and z_2 actually belong to \mathbb{R}^3 . Fully expanding the

system (52), one obtains

$$\begin{cases} y_1' = (c_1(h_1(v_1, v_2, v_3) - y_1)^3 + (h_1(v_1, v_2, v_3) - y_1))\chi(\tau)\zeta_{\epsilon_1}(\tau) \\ v_1' = (c_2(\sigma^{[n]}(y_1) - v_1) + (\sigma^{[n]}(y_1) - v_1))\chi(\tau)\zeta_{\epsilon_2}(-\tau) \\ y_2' = (c_1(h_2(v_1, v_2, v_3) - y_2)^3 + (h_2(v_1, v_2, v_3) - y_2))\chi(\tau)\zeta_{\epsilon_1}(\tau) \\ v_2' = (c_2(\sigma^{[n]}(y_2) - v_2)^3 + (\sigma^{[n]}(y_2) - v_2))\chi(\tau)\zeta_{\epsilon_2}(-\tau) \\ y_3' = (c_1(h_3(v_1, v_2, v_3) - y_3)^3 + (h_3(v_1, v_2, v_3) - y_3))\chi(\tau)\zeta_{\epsilon_1}(\tau) \\ v_3' = (c_2(\sigma^{[n]}(y_3) - v_3)^3 + (\sigma^{[n]}(y_3) - v_3))\chi(\tau)\zeta_{\epsilon_2}(-\tau) \\ u' = -u \\ \tau' = 2(u + \xi_2(v_3 - (m - 1), \tau + 1))^\alpha \xi_2(m - v_3, 10) - \xi_2(v_3 - (m - 1), 10 + 10\tau^2)\tau \end{cases} \quad (53)$$

where $z_1 = (y_1, y_2, y_3)$, $z_2 = (v_1, v_2, v_3)$, and $h = (f_1 \circ \sigma^{[k]}, f_2 \circ \sigma^{[k]}, f_3 \circ \sigma^{[k]}) = (h_1, h_2, h_3)$. Since $h(z_f) = f(\sigma^{[k]}(0), \sigma^{[k]}(0), \sigma^{[k]}(m)) = z_f$ (see the proof of property (I) in section 5.3), it follows that, in (53), at $z_{equilibrium} = (z_f, z_f, 0, 0)$,

$$\begin{aligned} h_i(v_1, v_2, v_3) &= y_i \\ \sigma^{[n]}(y_i) &= v_i \end{aligned}$$

for $i = 1, 2$, and 3 , which reduces many terms to 0 . Moreover, since the map $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is contracting near $z_f = (0, 0, m)$ (see Theorem 10), so is h (with a contraction factor bounded in absolute value by $0 \leq \lambda < 1$), thus

$$\|Dh(z_f)(z - z_f)\|_\infty \leq \lambda \|z - z_f\|_\infty \quad (54)$$

Now pick $z - z_f = (1/4, 0, 0)$. Since

$$Dh(z_f)(z - z_f) = \frac{1}{4} \begin{pmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_2}{\partial x_1} & \frac{\partial h_3}{\partial x_1} \end{pmatrix}$$

using (54) and the sup-norm on both sides, we get

$$\begin{aligned} \frac{\lambda}{4} &= \lambda \|z - z_f\|_\infty \geq \|Dh(z_f)(z - z_f)\|_\infty \geq \frac{1}{4} \left| \frac{\partial h_i}{\partial x_1} \right| \\ \Rightarrow \quad \lambda &\geq \left| \frac{\partial h_i}{\partial x_1} \right| \end{aligned}$$

for $i = 1, 2$, and 3 . Picking $z - z_f = (0, 1/4, 0)$ and $z - z_f = (0, 0, 1/4)$, and proceeding similarly, we reach the conclusion that all partial derivatives of h are bounded in absolute value by λ at the point z_h . Now we define $\sigma_i = \frac{\partial \sigma^{[n]}(y_i)}{\partial y_i}$. Notice that for $n \geq 1$,

$$|\sigma_i| \leq \lambda_{1/4} = 0.4\pi - 1 \approx 0.256637 \quad (55)$$

from Proposition 12 and the proof of Lemma 11. Therefore the Jacobian matrix

A of (53) at the point $z_e = z_{equilibrium}$ is

$$\begin{bmatrix} -\chi(\tau)\zeta_{\epsilon_1}(\tau)|_{z=z_e} & \frac{\partial h_1}{\partial v_1}\chi(\tau)\zeta_{\epsilon_1}(\tau)|_{z=z_e} & 0 & \frac{\partial h_1}{\partial v_2}\chi(\tau)\zeta_{\epsilon_1}(\tau)|_{z=z_e} \\ \sigma_1 \chi(\tau)\zeta_{\epsilon_2}(-\tau)|_{z=z_e} & -\chi(\tau)\zeta_{\epsilon_2}(-\tau)|_{z=z_e} & 0 & 0 \\ 0 & \frac{\partial h_2}{\partial v_1}\chi(\tau)\zeta_{\epsilon_1}(\tau)|_{z=z_e} & -\chi(\tau)\zeta_{\epsilon_1}(\tau)|_{z=z_e} & \frac{\partial h_2}{\partial v_2}\chi(\tau)\zeta_{\epsilon_1}(\tau)|_{z=z_e} \\ 0 & 0 & \sigma_2 \chi(\tau)\zeta_{\epsilon_2}(-\tau)|_{z=z_e} & -\chi(\tau)\zeta_{\epsilon_2}(-\tau)|_{z=z_e} \\ 0 & \frac{\partial h_3}{\partial v_1}\chi(\tau)\zeta_{\epsilon_1}(\tau)|_{z=z_e} & 0 & \frac{\partial h_3}{\partial v_2}\chi(\tau)\zeta_{\epsilon_1}(\tau)|_{z=z_e} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{\partial h_1}{\partial v_3}\chi(\tau)\zeta_{\epsilon_1}(\tau)|_{z=z_e} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{\partial h_2}{\partial v_3}\chi(\tau)\zeta_{\epsilon_1}(\tau)|_{z=z_e} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\chi(\tau)\zeta_{\epsilon_1}(\tau)|_{z=z_e} & \frac{\partial h_3}{\partial v_3}\chi(\tau)\zeta_{\epsilon_1}(\tau)|_{z=z_e} & 0 & 0 \\ \sigma_3 \chi(\tau)\zeta_{\epsilon_2}(-\tau)|_{z=z_e} & -\chi(\tau)\zeta_{\epsilon_2}(-\tau)|_{z=z_e} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & \beta & 0 & -\xi_2(1, 10) \end{bmatrix}$$

where $\beta \in \mathbb{R}$ is well defined (we do not need the explicit value of this partial derivative, it suffices to know its existence). To show that $z_{equilibrium}$ is a hyperbolic sink, we just need to prove that all eigenvalues of the above matrix have negative real part. Suppose, otherwise, that A admits an eigenvalue μ with nonnegative real part. Let $x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) \in \mathbb{C}^8$ be an eigenvector of A associated to μ . Since x is an eigenvector, $x \neq 0$. We also note that, from (50), $\chi(\tau)|_{z=z_e} = \chi(0) = 5$. Then, from the equation $Ax = \mu x$ and (45), one gets the following equations

$$\begin{aligned} \chi(\tau)\zeta_{\epsilon_1}(\tau)|_{z=z_e} \left(-x_1 + \frac{\partial h_1}{\partial v_1} \Big|_{z=z_e} x_2 + \frac{\partial h_1}{\partial v_2} \Big|_{z=z_e} x_4 + \frac{\partial h_1}{\partial v_3} \Big|_{z=z_e} x_6 \right) &= \mu x_1 \implies \\ \frac{1}{1 + \frac{\mu}{\chi(\tau)\zeta_{\epsilon_1}(\tau)|_{z=z_e}}} \left(\frac{\partial h_1}{\partial v_1} \Big|_{z=z_e} x_2 + \frac{\partial h_1}{\partial v_2} \Big|_{z=z_e} x_4 + \frac{\partial h_1}{\partial v_3} \Big|_{z=z_e} x_6 \right) &= x_1 \implies \\ \left| \frac{\partial h_1}{\partial v_1} \Big|_{z=z_e} x_2 + \frac{\partial h_1}{\partial v_2} \Big|_{z=z_e} x_4 + \frac{\partial h_1}{\partial v_3} \Big|_{z=z_e} x_6 \right| &\geq |x_1| \implies \\ \lambda (|x_2| + |x_4| + |x_6|) &\geq |x_1| \end{aligned} \tag{56}$$

We also obtain from $Ax = \mu x$ and (55)

$$\begin{aligned} \chi(\tau)\zeta_{\epsilon_2}(\tau)|_{z=z_e} (\sigma_1 x_1 - x_2) &= \mu x_2 \implies \\ \sigma_1 x_1 &= \left(1 + \frac{\mu}{\chi(\tau)\zeta_{\epsilon_2}(-\tau)|_{z=z_e}} \right) x_2 \implies \\ \lambda_{1/4} |x_1| &\geq |x_2| \end{aligned} \tag{57}$$

Similarly the following can be derived:

$$\begin{cases} \lambda(|x_2| + |x_4| + |x_6|) \geq |x_3| \\ \lambda(|x_2| + |x_4| + |x_6|) \geq |x_5| \end{cases} \quad \text{and} \quad \begin{cases} \lambda_{1/4}|x_3| \geq |x_4| \\ \lambda_{1/4}|x_5| \geq |x_6| \end{cases} \quad (58)$$

Then it follows from (56), (57), and (58) that

$$\begin{cases} \lambda\lambda_{1/4}(|x_1| + |x_3| + |x_5|) \geq \lambda(|x_2| + |x_4| + |x_6|) \geq |x_1| \\ \lambda\lambda_{1/4}(|x_1| + |x_3| + |x_5|) \geq \lambda(|x_2| + |x_4| + |x_6|) \geq |x_3| \\ \lambda\lambda_{1/4}(|x_1| + |x_3| + |x_5|) \geq \lambda(|x_2| + |x_4| + |x_6|) \geq |x_5| \end{cases}$$

Adding the inequalities, we get

$$3\lambda\lambda_{1/4}(|x_1| + |x_3| + |x_5|) \geq |x_1| + |x_3| + |x_5|$$

which implies (note that $0 < 3\lambda_{1/4} < 1$)

$$\lambda(|x_1| + |x_3| + |x_5|) \geq |x_1| + |x_3| + |x_5|$$

Since $|\lambda| < 1$, the above holds true only if $x_1 = x_2 = x_3 = 0$. Moreover, from (57) and (58) one also concludes $x_2 = x_4 = x_6 = 0$. Thus the equation $Ax = \mu x$ is reduced to

$$\begin{cases} -x_7 = \mu x_7 \\ -\xi_2(1, 10)x_8 = \mu x_8 \end{cases}$$

Since $0 < \xi_2(1, 10) < 1$ and μ has nonnegative real part, the system above is satisfied only if $x_7 = x_8 = 0$. In other words, any eigenvector associated to the eigenvalue μ is a zero vector, which is clearly a contradiction. Therefore, all eigenvalues of A have negative real part, i.e. $z_{equilibrium}$ is a hyperbolic sink for the ODE (52).

We note that all functions used in system (52) are computable (they are defined by composing usual functions of analysis with some of their analytic continuations, and therefore are computable; see [26], [7]).

5.6 Proof of Theorem 9

In the previous subsections 5.4 and 5.5, we have shown that the universal Turing machine M can be simulated by the ODE (52) which has a hyperbolic sink at $z_{equilibrium} = (z_f, z_f, 0, 0) \in \mathbb{N}^8$. The lemma below follows from the results in those sections.

Lemma 18 *Suppose that $x_0 \in \mathbb{N}^3$ codes an initial configuration of M simulated by system (52), and let $z_0 = (x_0, x_0, 1, 4)$. Then for every point $z \in \mathbb{R}^8$ satisfying $\|z - z_0\| < 1/16$, one has:*

1. If M halts on x_0 , then the trajectory starting at z converges to the hyperbolic sink $z_{equilibrium}$.
2. If M does not halt on x_0 , then the trajectory starting at z does not converge to $z_{equilibrium}$.

We now proceed to the proof of Theorem 9.

of Theorem 9. Let $g : \mathbb{R}^6 \times (-1, +\infty) \times (-1, +\infty) \rightarrow \mathbb{R}^8$ be the function in the right-hand side of system (52) and let $s = (z_f, z_f, 0, 0)$. Then g is analytic. As

proved in previous subsections, s is a hyperbolic sink of system (52), where z_f corresponds to the unique halting configuration of the universal Turing machine M simulated by system (52). Let us denote the basin of attraction of s as W_{final} . It then follows that, for any $z_0 = (x_0, x_0, 1, 4)$ with $x_0 \in \mathbb{N}^3$ being an initial configuration of M , M halts on x_0 iff $z_0 \in W_{final}$. Moreover, from Lemma 18, any trajectory starting at a point inside $B(z_0, 1/16)$ will either converge to s if M halts on x_0 or not converge to s if M does not halt on x_0 . In other words, $B(z_0, 1/16)$ is either inside W_{final} (if M halts on x_0) or inside $\mathbb{R}^8 - W_{final}$ (if M does not halt on x_0).

Let \mathfrak{R} denote $\mathbb{R}^6 \times (-1, +\infty) \times (-1, +\infty)$. Now suppose that W_{final} is a computable open subset of \mathfrak{R} . Then the distance function $d_{\mathfrak{R} \setminus W_{final}}$ is computable. Therefore, for any $x_0 \in \mathbb{N}^3$, we can compute $d_{\mathfrak{R} \setminus W_{final}}(z_0)$ with a precision of $1/40$ which yields some rational q , where $z_0 = (x_0, x_0, 1, 4)$. We note that either $d_{\mathfrak{R} \setminus W_{final}}(z_0) = 0$ (if $z_0 \notin W_{final}$) or $d_{\mathfrak{R} \setminus W_{final}}(z_0) \geq 1/16$ (if $z_0 \in W_{final}$). In the first case, $q \leq 1/40$, while in the second case, $q \geq \frac{1}{16} - \frac{1}{40} = \frac{3}{80}$. The following algorithm then solves the Halting problem, which is absurd: on initial configuration x_0 , compute $d_{\mathfrak{R} \setminus W_{final}}(z_0)$, $z_0 = (x_0, x_0, 1, 4)$, with a precision of $1/40$ yielding some rational q . If $q \leq 1/40$ then M does not halt on x_0 ; if $q \geq 3/80$, then M halts on x_0 . Therefore W_{final} cannot be computable. \blacksquare

6 Conclusion

We have shown that the basins of attraction of even very regular systems, namely the basins of attraction of hyperbolic asymptotically stable equilibrium points (hyperbolic sinks) of analytic dynamical systems are, in general, non-computable.

Notice that these basins of attractions are in a sense “robustly non-computable”: perturbing the dynamical system will not break the non-computability of the basin of attraction, indicating that non-computability is not necessarily a “fragile” property.

The “robustness of non-computability” stems from the well-known fact (see e.g. [17, Proposition of p. 305]) that a dynamical system with a hyperbolic sink x_0 can be C^1 -perturbed; the perturbation will produce a nearby hyperbolic sink \tilde{x}_0 that is the only attractor within a vicinity of x_0 . Although we have not done it in this paper, for reasons of space, one can in principle C^1 -perturb the dynamical system of Theorem 8 (at least for the discrete-time case) and still be able to carry out the simulation of a Turing machine, as it was done in [15]. The halting configuration in the perturbed system will then be \tilde{x}_0 , and the basin of attraction of \tilde{x}_0 will continue to be non-computable for reasons similar to those pointed out in the proof of Theorem 8.

As an open problem, let us notice that the discrete-time dynamical system with a non-computable basin of attraction of Theorem 8 is defined on \mathbb{R}^3 only because it was the case more suitable for our method. It would be interesting to know whether basins of attractions of hyperbolic sinks can still be non-computable in \mathbb{R}^2 . Moreover, since it is believed that no analytic function on a compact, finite-dimensional space can simulate a Turing machine through a reasonable input and output encoding [23], the technique presented in the present paper does not allow us to infer what happens for basins of attraction

of compact dynamical systems.

A second open problem. In [10] the authors showed that Julia sets are, in general, not computable; these non-computable Julia sets can be boundaries of basins of attraction of hyperbolic sinks. However, on the other hand, Julia sets defined by hyperbolic polynomials are computable, in a very efficient manner (in polynomial time) [8], [27]. Note that a hyperbolic polynomial defines a hyperbolic system (hyperbolic globally) while a hyperbolic sink is a local property. It would be very exciting to show that global hyperbolicity is not a barrier to non-computability. Since the analytic systems with computable hyperbolic sinks constructed in this paper fail to be globally hyperbolic, the following open problem presents itself: Are basins of attraction of computable analytic and hyperbolic systems computable?

Acknowledgments. D. Graça was partially supported via R&D Unit 50008, financed by the applicable financial framework (FCT/MEC through national funds and when applicable co-funded by FEDER - PT2020 partnership agreement).

References

- [1] V. I. Arnold and A. Avez. *Ergodic Problems of Classical Mechanics*. W.A. Benjamin, 1968.
- [2] K. E. Atkinson. *An Introduction to Numerical Analysis*. John Wiley & Sons, 2nd edition, 1989.
- [3] A. F. Beardon. *Iteration of Rational Functions*. Springer, 1991.
- [4] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer, 1998.
- [5] M. S. Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoret. Comput. Sci.*, 138(1):67–100, 1995.
- [6] V. Brattka. The emperor’s new recursiveness: the epigraph of the exponential function in two models of computability. In M. Ito and T. Imaoka, editors, *Words, Languages & Combinatorics III*, Kyoto, Japan, 2000. ICWLC 2000.
- [7] 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.
- [8] M. Braverman. Computational complexity of euclidean sets: hyperbolic Julia sets are poly-time computable. In V. Brattka, L. Staiger, and K. Weihrauch, editors, *Proc. 6th Workshop on Computability and Complexity in Analysis (CCA 2004)*, volume 120 of *Electron. Notes Theor. Comput. Sci.*, pages 17–30. Elsevier, 2005.

- [9] M. Braverman and S. Cook. Computing over the reals: foundations for scientific computing. *Notices Amer. Math. Soc.*, 53(3):318–329, 2006.
- [10] M. Braverman and M. Yampolsky. Non-computable Julia sets. *J. Amer. Math. Soc.*, 19(3):551–578, 2006.
- [11] M. Campagnolo and C. Moore. Upper and lower bounds on continuous-time computation. In I. Antoniou, C. Calude, and M. Dinneen, editors, *2nd International Conference on Unconventional Models of Computation - UMC'2K*, pages 135–153. Springer, 2001.
- [12] M. L. Campagnolo. *Computational Complexity of Real Valued Recursive Functions and Analog Circuits*. PhD thesis, Instituto Superior Técnico/Universidade Técnica de Lisboa, 2002.
- [13] M. L. Campagnolo, C. Moore, and J. F. Costa. Iteration, inequalities, and differentiability in analog computers. *J. Complexity*, 16(4):642–660, 2000.
- [14] G. Chesi. Estimating the domain of attraction for uncertain polynomial systems. *Automatica*, 40(11):1981–1986, 2004.
- [15] D. S. Graça, M. L. Campagnolo, and J. Buescu. Computability with polynomial differential equations. *Adv. Appl. Math.*, 40(3):330–349, 2008.
- [16] D.S. Graça, N. Zhong, and J. Buescu. Computability, noncomputability and undecidability of maximal intervals of IVPs. *Trans. Amer. Math. Soc.*, 361(6):2913–2927, 2009.
- [17] M. W. Hirsch and S. Smale. *Differential Equations, Dynamical Systems, and Linear Algebra*. Academic Press, 1974.
- [18] J. H. Hubbard and B. H. West. *Differential Equations: A Dynamical Systems Approach — Higher-Dimensional Systems*. Springer, 1995.
- [19] P. Koiran and C. Moore. Closed-form analytic maps in one and two dimensions can simulate universal Turing machines. *Theoret. Comput. Sci.*, 210(1):217–223, 1999.
- [20] S. Lang. *Calculus of Several Variables*. Springer, 3rd edition, 1987.
- [21] L. G. Matallana, A. M. Blanco, and J. a. Bandoni. Estimation of domains of attraction: A global optimization approach. *Mathematical and Computer Modelling*, 52(3-4):574–585, 2010.
- [22] C. Moore. Generalized shifts: unpredictability and undecidability in dynamical systems. *Nonlinearity*, 4(2):199–230, 1991.
- [23] C. Moore. Finite-dimensional analog computers: Flows, maps, and recurrent neural networks. In C. Calude, J. Casti, and M. Dinneen, editors, *1st International Conference on Unconventional Models of Computation - UMC'98*, pages 59–71. Springer, 1998.
- [24] P. Odifreddi. *Classical Recursion Theory*, volume 1. Elsevier, 1989.
- [25] L. Perko. *Differential Equations and Dynamical Systems*. Springer, 3rd edition, 2001.

- [26] M. B. Pour-El and J. I. Richards. *Computability in Analysis and Physics*. Springer, 1989.
- [27] R. Rettinger. A fast algorithm for julia sets of hyperbolic rational functions. In V. Brattka, L. Staiger, and K. Weihrauch, editors, *Proc. 6th Workshop on Computability and Complexity in Analysis (CCA 2004)*, volume 120 of *Electron. Notes Theor. Comput. Sci.*, pages 145–157. Elsevier, 2005.
- [28] M. Sipser. *Introduction to the Theory of Computation*. Course Technology, 2nd edition, 2005.
- [29] S. Smale. Differentiable dynamical systems. *Bull. Amer. Math. Soc.*, 73:747–817, 1967.
- [30] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, (Ser. 2–42):230–265, 1936.
- [31] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. a correction. *Proc. London Math. Soc.*, (Ser. 2–43):544–546, 1937.
- [32] K. Weihrauch. *Computable Analysis: an Introduction*. Springer, 2000.
- [33] N. Zhong. Computational unsolvability of domain of attractions of nonlinear systems. *Proc. Amer. Math. Soc.*, 137:2773–2783, 2009.