# Robust Simulations of Turing Machines with Analytic Maps and Flows

Daniel S. Graça[1,2], Manuel L. Campagnolo[3,2] and Jorge Buescu[4]

[1] DM/FCT, Universidade do Algarve, C. Gambelas, 8005-139 Faro, Portugal
dgraca@ualg.pt
[2] CLC, DM/IST, Universidade Técnica de Lisboa, 1049-001 Lisboa, Portugal
[3] DM/ISA, Universidade Técnica de Lisboa, 1349-017 Lisboa, Portugal
mlc@math.isa.utl.pt
[4] CAMGSD, DM/IST, Universidade Técnica de Lisboa, 1049-001 Lisboa, Portugal
jbuescu@math.ist.utl.pt

**Abstract.** In this paper, we show that closed-form analytic maps and flows can simulate Turing machines in an error-robust manner. The maps and ODEs defining the flows are explicitly obtained and the simulation is performed in real time.

## 1 Introduction

Since the pioneering work of Turing in the 1930s, the Turing machine has become the standard paradigm for computation. With the appearance and rapid development of digital computers its role has become increasingly important. In this paper we show that the behavior of Turing machines can be embedded in robust and analytic analog systems defined on continuous spaces.

Several authors have proved that finite dimensional maps and flows can simulate Turing machines. The general approach is to associate each configuration of a Turing machine to a point of $\mathbb{R}^n$, and to show that there is a dynamical system with state space in $\mathbb{R}^n$ that embeds its evolution. It is known that Turing machines can be simulated on compact spaces, even of low dimension [1–3]. While compactness is a desirable property of dynamical systems, it is probably too strong a requirement since it is believed that no analytic map on a compact, finite dimensional space can simulate a Turing machine through a reasonable encoding [4]. However, most physical systems turn out to be analytic, at least in the classical world of Physics. Even the physical model underlying digital computers is analytic, although their behavior is idealized as discrete.

The requirement of compactness has another drawback since it prevents systems capable of simulating an arbitrary Turing machine to exhibit robustness to noise. For instance, Casey [5, 6] showed that in the presence of bounded analog noise, recurrent neural networks can only recognize regular languages. This result was later generalized in [7] to other analog discrete-time computational systems. Robustness is a critical issue in analog models since non-computable behavior might arise when the use of exact real quantities is allowed. For instance, the results of Pour-El, Richards and Zhong [8, 9] show that there is a

three-dimensional wave equation, with computable initial conditions, such that its unique solution is not computable. However, that behavior is ruled out in the presence of noise [10]. Recurrent analog neural networks are another known case where non-computable behavior can occur if real parameters are represented with infinite precision [3].

In this paper we will show that Turing machines can be simulated by finite dimensional maps and flows which are both analytic and robust. We will consider simulations on unbounded spaces. Our work is in some sense related to [11], where a constructive simulation of Turing machines using closed-form analytic maps is presented. However, in [11] it is not discussed how the presence of noise affects the computational power of the model. We prove here that any Turing machine M can be simulated by a closed-form analytic map $f_M : \mathbb{R}^3 \to \mathbb{R}^3$, even in the case where some noise is added to the initial configuration or during the evolution of the system.

The previously mentioned results show that finite dimensional maps are capable of simulating the *transition function* of an arbitrary Turing machine. In that respect, those are results about the computational power of hybrid systems, which are continuous with respect to the state space but evolve discretely in time. Another approach has been to simulate the evolution of Turing machines with continuous flows in $\mathbb{R}^n$ [12–14]. Even if it is known that those flows can be infinitely differentiable, no analytic form of iterating the map that simulates the transition function of a Turing machine had been proposed before. Furthermore, it is known that analytic differentially algebraic functions, which include most of the usual mathematical analytic functions, are not closed under iteration [15], which suggests that continuous-time computational models which are closed under iteration must contain some non-analytic functions [16]. However, since we only have to iterate functions in the vicinity of integers, we are able to show that any Turing machine M can be robustly simulated by some system of differential equations $y' = g_M(y, t)$, where $g_M$ is analytic and $t$ represents the time steps of $M$.

It is worthwhile to notice that our work can be included, in some sense, in the wider topic of stable dynamical systems. In fact, there has been a long tradition of considering only structurally stable systems [17] when modelling physical systems. The argument is that, due to measurement uncertainties, qualitative properties of a system should not change with small perturbations. Guckenheimer and Holmes [18] refer to this approach as the "stability dogma". However, recent developments in the theory of dynamical systems suggest that this is too restrictive to account for all meaningful systems [19]. In fact, one can relax the previous condition and demand stability only for those properties of interest for the system under consideration. Here, we have chosen the latter line of work: our only concern is that each system performs a simulation of a Turing machine robust to perturbations.

The paper can be outlined as follows. In Section 2 we introduce the ideas and concepts related to simulations robust to perturbations. Section 3 provides tools that will be necessary in Section 4. In Section 4, we prove (in a constructive

manner) the main results of this paper: each Turing machine can be simulated by an analytic map, or by ODEs even under the influence of (small) errors. The maps and ODEs are explicitly obtained, by using expressions involving the composition of polynomials and trigonometric functions, and only computable constants are used. We end describing some connections of this paper with previous results on continuous-time models of computation.

## 2 Simulation of Turing machines

Before stating the main results, we describe succinctly some aspects of our error-robust simulation of Turing machines. For now, we will be only concerned with discrete time simulations. Therefore we want to obtain a map that "captures" the behavior of the transition function. We will code each configuration into a triple $(x, y, z) \in \mathbb{N}^3$, and prove that the simulation still works if this triple is slightly perturbed. Without loss of generality, consider a Turing machine M using 10 symbols, the blank symbol $B = 0$, and symbols $1, 2, ...9$. Let

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

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

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

and let $q$ be the state associated to the current configuration. Then the triple $(y_1, y_2, q) \in \mathbb{N}^3$ gives the current configuration of M. We now can state the first main result of this paper as follows:[5]

**Theorem 1.** *Let $\theta : \mathbb{N}^3 \to \mathbb{N}^3$ be the transition function of a Turing machine M, under the encoding described above and let $0 < \delta < \varepsilon < 1/2$. Then $\theta$ admits an analytic extension $f_M : \mathbb{R}^3 \to \mathbb{R}^3$, robust to perturbations in the following sense: for all $f$ such that $\|f - f_M\|_\infty \leq \delta$ and for all $\bar{x}_0 \in \mathbb{R}^3$ satisfying $\|\bar{x}_0 - x_0\|_\infty \leq \varepsilon$, where $x_0 \in \mathbb{N}^3$ represents an initial configuration,*

$$\left\| f^{[j]}(\bar{x}_0) - \theta^{[j]}(x_0) \right\|_\infty \leq \varepsilon \qquad \text{for all } j \in \mathbb{N}.$$

A few remarks are in order. First, and as noticed before, we implicitly assumed that if $y$ is a halting configuration, then $\theta(y) = y$. Secondly, we notice that the upper bound $(\frac{1}{2})$ on $\varepsilon$ results from the encoding we have chosen, which is over the integers. In fact, the bound is maximal with respect to that encoding.

Incidentally, we notice that Theorem 1 can be stated using the notion of *shadowing* in dynamical systems (cf. [20, 21]), which is formally defined as below.

---

[5] We take $\|(x_1, ..., x_n)\|_\infty = \max_{1 \leq i \leq n} |x_i|$ and $\|f\|_\infty = \sup_{x \in \mathbb{R}} \|f(x)\|_\infty$, where $f$ is a real function. If $f : A \to A$ is a function, then $f^{[k]}$ denotes the $k$th iterate of $f$.

**Definition 1.** *Let $f : A \to A$ be a map, $\varepsilon > 0$, and $\{p_i\}_{i \in \mathbb{N}} \subseteq A$. Then $\{p_i\}_{i \in \mathbb{N}}$ is a $\varepsilon$-pseudo-orbit of $f$ if $|p_{i+1} - f(p_i)| < \varepsilon$ for all $i \in \mathbb{N}$. For $x \in A$, we say that $\{f^{[i]}(x)\}_{i \in \mathbb{N}}$ $\varepsilon$-shadows the pseudo-orbit $\{p_i\}_{i \in \mathbb{N}}$ if $\left| f^{[i]}(x) - p_i \right| < \varepsilon$.*

In short, we say that $\{p_i\}_{i \in \mathbb{N}}$ is a good approximation of some system whose dynamics is given by $f$, if $\{f^{[i]}(x)\}_{i \in \mathbb{N}}$ $\varepsilon$-shadows $\{p_i\}_{i \in \mathbb{N}}$. Using the previous definition, we can restate Theorem 1 by saying that the sequence $\{f_M^{[j]}(x_0)\}_{j \in \mathbb{N}}$ of configurations $\varepsilon$-shadows $\{f^{[j]}(\bar{x}_0)\}_{j \in \mathbb{N}}$. We now present the other main results.

**Theorem 2.** *Let $\theta : \mathbb{N}^3 \to \mathbb{N}^3$ be the transition function of a Turing machine $M$, under the encoding described above and let $0 < \varepsilon < 1/4$. Then there is an analytic function $z : \mathbb{R}^4 \to \mathbb{R}^3$ with the following property:*

$$\left\| z(x_0, j) - \theta^{[j]}(x_0) \right\|_\infty \leq \varepsilon$$

*for all $j \in \mathbb{N}$, where $x_0 \in \mathbb{N}^3$ represents an initial configuration.*

As a matter of fact, we will prove the following "robust" version of Theorem 2.

**Theorem 3.** *In the conditions of Theorem 2, there is an analytic function $g_M : \mathbb{R}^6 \to \mathbb{R}^6$ such that the ODE $z' = g_M(z, t)$ robustly simulates $M$ in the following sense: there is some $0 < \eta < 1/2$ such that for all $g$ satisfying $\|g - g_M\|_\infty < 1/2$, and for all $\bar{x}_0 \in \mathbb{R}^3$ satisfying $\|\bar{x}_0 - x_0\|_\infty \leq \varepsilon$, the solution $z$ of*

$$z' = g(z, t), \qquad z(0) = (\bar{x}_0, \bar{x}_0)$$

*has the following property: for all $j \in \mathbb{N}$ and for all $t \in [j, j + 1/2]$,[6]*

$$\left\| z_2(t) - \theta^{[j]}(x_0) \right\|_\infty \leq \eta.$$

## 3  Preliminary results

This section is devoted to the presentation of results that, while not very interesting on their own, will be useful when proving Theorem 1. As our first task, we introduce an analytic extension $\omega : \mathbb{R} \to \mathbb{R}$ for the function $f : \mathbb{N} \to \mathbb{N}$ defined by $f(n) = n \bmod 10$. This function will be necessary when simulating Turing machines. It will be used to read symbols written in the tape. To achieve this purpose, we can use a periodic function, of period 10, such that $\omega(i) = i$, for $i = 0, 1, ..., 9$. Then, using trigonometric interpolation (cf. [22, pp. 176-182]), one may take

$$\omega(x) = a_0 + a_5 \cos(\pi x) + \left( \sum_{j=1}^{4} a_j \cos\left( \frac{j\pi x}{5} \right) + b_j \sin\left( \frac{j\pi x}{5} \right) \right), \qquad (2)$$

---

[6] For simplicity, we denote $z$ by $(z_1, z_2)$, where $z_1, z_2 \in \mathbb{R}^3$.

where $a_0, ..., a_5, b_1, ..., b_4$ are computable coefficients that can be explicitly obtained by solving a system of linear equations.

It is easy to see that $\omega$ is uniformly continuous in $\mathbb{R}$. Hence, for every $\varepsilon \in (0, 1/2)$, there will be some $\zeta_\varepsilon > 0$ satisfying

$$\forall n \in \mathbb{N}, \; x \in [n - \zeta_\varepsilon, n + \zeta_\varepsilon] \;\; \Rightarrow \;\; |\omega(x) - n \bmod 10| \leq \varepsilon. \tag{3}$$

When simulating a Turing machine, we will also need to keep the error under control. In many cases, this will be done with the help of the error-contracting function defined by

$$\sigma(x) = x - 0.2 \sin(2\pi x).$$

The function $\sigma$ is a contraction on the vicinity of integers:

**Lemma 1.** *Let $n \in \mathbb{Z}$ and let $\varepsilon \in [0, 1/2)$. Then there is some contracting factor $\lambda_\varepsilon \in (0, 1)$ such that, for $\forall \delta \in [-\varepsilon, \varepsilon]$, $|\sigma(n + \delta) - n| < \lambda_\varepsilon \delta$.*

*Remark 1.* Throughout the remainder of this paper, we suppose that $\varepsilon \in [0, 1/2)$ is fixed and that $\lambda_\varepsilon$ is the respective contracting factor given by Lemma 1.

The function $\sigma$ will be used in our simulation to keep the error controlled when bounded quantities are involved (e.g., the actual state, the symbol being read, etc.). We will also need another error-contracting function that controls the error for unbounded quantities. This will be achieved with the help of the function $l_3 : \mathbb{R}^2 \to \mathbb{R}$, that has the property that whenever $\bar{a}$ is an approximation of $a \in \{0, 1, 2\}$, then $|l_3(\bar{a}, y) - a| < 1/y$, for $y > 0$. In other words, $l_3$ is an error-contracting map, where the error is contracted by an amount specified by the second argument of $l_3$. We start by defining a preliminary function $l_2$ satisfying similar conditions, but only when $a \in \{0, 1\}$.

**Lemma 2.** *Let $l_2 : \mathbb{R}^2 \to \mathbb{R}$ 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, for any $\bar{a}, y \in \mathbb{R}$ satisfying $|a - \bar{a}| \leq 1/4$ and $y > 0$, we get $|a - l_2(\bar{a}, y)| < 1/y$.*

**Lemma 3.** *Let $a \in \{0, 1, 2\}$ and let $l_3 : \mathbb{R}^2 \to \mathbb{R}$ be given by*

$$l_3(x, y) = l_2((\sigma^{[d+1]}(x) - 1)^2, 3y).(2l_2(\sigma^{[d]}(x)/2, 3y) - 1) + 1,$$

*where $d = 0$ if $\varepsilon \leq 1/4$ and $d = \lceil -\log(4\varepsilon)/\log \lambda_\varepsilon \rceil$ otherwise. Then for any $\bar{a}, y \in \mathbb{R}$ satisfying $|a - \bar{a}| \leq \varepsilon$ and $y \geq 2$, we have $|a - l_3(\bar{a}, y)| < 1/y$.*

The following lemma can be easily proved by induction on $n$.

**Lemma 4.** *If $|\alpha_i|, |\bar{\alpha}_i| \leq K$ for $i = 1, ..., n$ then*

$$|\alpha_1 ... \alpha_n - \bar{\alpha}_1 ... \bar{\alpha}_n| \leq (|\alpha_1 - \bar{\alpha}_1| + ... + |\alpha_n - \bar{\alpha}_n|) K^{n-1}.$$

5

## 4 Robust analytic simulations of Turing machines

In this section we show, in a constructive manner, how to simulate a Turing machine with an analytic map robust to (small) perturbations. We will first prove the following theorem.

**Theorem 4.** *Let $\theta : \mathbb{N}^3 \to \mathbb{N}^3$ be the transition function of some Turing machine. Then, given some $0 \leq \varepsilon < 1/2$, $\theta$ admits an analytic extension $h_M : \mathbb{R}^3 \to \mathbb{R}^3$ with the property that*

$$\|(y_1, y_2, q) - (\bar{y}_1, \bar{y}_2, \bar{q})\|_\infty \leq \varepsilon \quad \Rightarrow \quad \|\theta(y_1, y_2, q) - h_M(\bar{y}_1, \bar{y}_2, \bar{q})\|_\infty \leq \varepsilon. \quad (4)$$

*Proof.* We will show how to construct $h_M$ with analytic functions:

1. **Determine the symbol being read.** Let $a_0$ be the symbol being actually read by the Turing machine M. Then $\omega(y_1) = a_0$, where $\omega$ is given by (2). But what about the effect of the error present in $\bar{y}_1$? Since $|y_1 - \bar{y}_1| \leq \varepsilon$,

$$|a_0 - \omega \circ \sigma^{[l]}(\bar{y}_1)| \leq \varepsilon, \quad \text{with } l = \left\lceil \left| \frac{\log(\zeta_\varepsilon/\varepsilon)}{\log \lambda_\varepsilon} \right| \right\rceil, \quad (5)$$

   where $\zeta_\varepsilon$ is given by (3). Then pick $\bar{y} = \omega \circ \sigma^{[l]}(\bar{y}_1)$ as an approximation of the symbol being currently read. Similarly, $\omega \circ \sigma^{[l]}(\bar{y}_2)$ gives an approximation of $a_{-1}$, with error bounded by $\varepsilon$.

2. **Determine the next state.** The map that returns the next state is defined by polynomial interpolation. This can be done as follows. Let $y$ be the symbol being currently read and $q$ the current state. Recall that $m$ denotes the number of states and $k = 10$ is the number of symbols. One may take

$$q_{next} = \sum_{i=0}^{9} \sum_{j=1}^{m} \left( \prod_{\substack{r=0 \\ r \neq i}}^{9} \frac{(y - r)}{(i - r)} \right) \left( \prod_{\substack{s=1 \\ s \neq j}}^{m} \frac{(q - s)}{(j - s)} \right) q_{i,j},$$

   where $q_{i,j}$ is the state that follows symbol $i$ and state $j$. However, we are dealing with the approximations $\bar{q}$ and $\bar{y}$. Therefore, we define instead

$$\bar{q}_{next} = \sum_{i=0}^{9} \sum_{j=1}^{m} \left( \prod_{\substack{r=0 \\ r \neq i}}^{9} \frac{(\sigma^{[n]}(\bar{y}) - r)}{(i - r)} \right) \left( \prod_{\substack{s=1 \\ s \neq j}}^{m} \frac{(\sigma^{[n]}(\bar{q}) - s)}{(j - s)} \right) q_{i,j}, \quad (6)$$

   with

$$n = \left\lceil \frac{\log(10m^2 K^{m+7}(m+8))}{-\log \lambda_\varepsilon} \right\rceil, \qquad K = \max\{9.5, m + 1/2\}.$$

With this choice for $n$, the error of $\sigma^{[n]}(\bar{y})$ and $\sigma^{[n]}(\bar{q})$ is such that

$$9|y - \sigma^{[n]}(\bar{y})| + (m-1)|q - \sigma^{[n]}(\bar{q})| \leq \frac{\varepsilon}{10m^2 K^{m+7}}. \quad (7)$$

Thus, from (6), (7) and Lemma 4, we conclude that $|\bar{q}_{next} - q_{next}| \leq \varepsilon$.

6

3. **Determine the symbol to be written on the tape.** Using a similar construction, the symbol to be written, $s_{next}$, can be approximated with precision $\varepsilon$, i.e. $|s_{next} - \bar{s}_{next}| \leq \varepsilon$.

4. **Determine the direction of the move for the head.** Let $h$ denote the direction of the move of the head, where $h = 0$ denotes a move to the left, $h = 1$ denotes a "no move", and $h = 2$ denotes a move to the right. Then, again, the "next move" $h_{next}$ can be approximated by means of a polynomial interpolation as in steps 3 and 4, therefore obtaining $|h_{next} - \bar{h}_{next}| \leq \varepsilon$.

5. **Update the tape contents.** We define functions $\bar{P}_1, \bar{P}_2, \bar{P}_3$ which are intended to approximate the tape contents after the head moves left, does not move, or moves right, respectively. Let $H$ be a "sufficiently good" approximation of $h_{next}$, yet to be determined. Then, the next value of $y_1$, $y_1^{next}$, can be approximated by

$$\bar{y}_1^{next} = \bar{P}_1 \tfrac{1}{2}(1 - H)(2 - H) + \bar{P}_2 H(2 - H) + \bar{P}_3 (-\tfrac{1}{2})H(1 - H), \quad (8)$$

with

$$\bar{P}_1 = 10(\sigma^{[j]}(\bar{y}_1) + \sigma^{[j]}(\bar{s}_{next}) - \sigma^{[j]}(\bar{y})) + \sigma^{[j]} \circ \omega \circ \sigma^{[l]}(\bar{y}_2),$$

$$\bar{P}_2 = \sigma^{[j]}(\bar{y}_1) + \sigma^{[j]}(\bar{s}_{next}) - \sigma^{[j]}(\bar{y}), \quad \bar{P}_3 = \frac{\sigma^{[j]}(\bar{y}_1) - \sigma^{[j]}(\bar{y})}{10},$$

where $j \in \mathbb{N}$ is sufficiently large and $l$ is given by (5). Notice that when exact values are used, $\bar{y}_1^{next} = y_1^{next}$. The problem in this case is that $\bar{P}_1$ depends on $\bar{y}_1$, which is not a bounded value. Thus, if we simply take $\bar{H} = \bar{h}_{next}$, the error of the term $(1 - H)(2 - H)/2$ is arbitrarily amplified when this term is multiplied by $\bar{P}_1$. Hence, $\bar{H}$ must be a sharp estimate of $h_{next}$, proportional to $\bar{y}_1$. Therefore, using Lemma 3 and the definition of $y_1$, one can see that it is suffices to take

$$H = l_3(\bar{h}_{next}, 10000 (\bar{y}_1 + 1/2) + 2).$$

Using the same argument for $\bar{P}_2$ and $\bar{P}_3$, we conclude that $|\bar{y}_1^{next} - y_1^{next}| < \varepsilon$. Similarly, and for the left side of the tape, we can define $\bar{y}_2^{next}$ such that $|\bar{y}_2^{next} - y_2^{next}| < \varepsilon$,.

Finally, $h_M : \mathbb{R}^3 \to \mathbb{R}^3$ is defined by $h_M(\bar{y}_1, \bar{y}_2, \bar{q}) = (\bar{y}_1^{next}, \bar{y}_2^{next}, \bar{q}_{next})$. $\quad\square$

We shall now prove the main results of this paper.

**Proof of Theorem 1.** Let $0 \leq \delta < \varepsilon$. Then, using Theorem 4, one can find a map $h_M$ such that (4) holds. Let $i \in \mathbb{N}$ satisfy $\sigma^{[i]}(\varepsilon) \leq \varepsilon - \delta$. Define a map $f_M = \sigma^{[i]} \circ h_M$. Then, if $x_0 \in \mathbb{N}^3$ is an initial configuration,

$$\|\bar{x}_0 - x_0\|_\infty \leq \varepsilon \quad \Rightarrow \quad \|f_M(\bar{x}_0) - \theta(x_0)\|_\infty \leq \varepsilon - \delta.$$

Thus, by triangular inequality, if $\|\bar{x}_0 - x_0\|_\infty \leq \varepsilon$, then

$$\|f(\bar{x}_0) - \theta(x_0)\|_\infty \leq \|f(\bar{x}_0) - f_M(\bar{x}_0)\|_\infty + \|f_M(\bar{x}_0) - \theta(x_0)\|_\infty \leq \delta + (\varepsilon - \delta) = \varepsilon.$$

This proves the result for $j = 1$. For $j > 1$, we proceed by induction. $\square$

**Proof of Theorem 3.** (Sketch) We adapt the construction in [12] to simulate the iteration of the transition function of a TM with ODEs, using our Theorem 1 to generalize Branicky's construction to analytic and robust flows. To iterate a function $\theta$ we use a pair of functions to control the evolution of two "simulation" variables $z_1$ and $z_2$. Both simulation variables have values close to $x_0$ at $t = 0$. The first variable is iterated during half of an unit period while the second remains approximately constant (its derivative is kept close to zero by a control function that involves our error-contracting function $l_2$). Then, the first variable remains controlled during the following half unit period of time and the second variable is brought up close to it. Therefore, at time $t = 1$ both variables have values close to $\theta(x_0)$. Theorem 1 shows that there exists some analytic function robust to errors that simulates $\theta$. This allow us to repeat the process an arbitrary number of times, keeping the error under control.

We begin with some preliminary results. There exists an ODE whose solution can be as close as desired to an arbitrary fixed value $b \in \mathbb{R}$ at $t = 1/2$, for any initial condition at $t = 0$. Let $\phi : \mathbb{R} \to \mathbb{R}^+$ be some function. For an arbitrary error $\gamma > 0$ we define a perturbed version, where we allow an error $\rho \geq 0$ on $b$ and a perturbation term bounded by $\delta \geq 0$:

$$z' = -c(z - \bar{b}(t))^3 \phi(t) + E(t), \qquad \text{with} \;\; c \geq \left(2\gamma^2 \int_0^{1/2} \phi(t)dt\right)^{-1}. \quad (9)$$

where $\left|\bar{b}(t) - b\right| \leq \rho$ and $|E(t)| \leq \delta$. Using the theory of ODEs, we can conclude that $\left|z(\frac{1}{2}) - b\right| < \gamma + \rho + \delta/2$ regardless to the initial condition at $t = 0$.

For the control functions mentioned above, we use $s : \mathbb{R} \to [-\frac{1}{8}, 1]$ defined by

$$s(t) = \frac{1}{2}\left(\sin^2(2\pi t) + \sin(2\pi t)\right).$$

On $[0, 1/2]$ $s$ ranges between 0 and 1 and on $[1/2, 1]$ $s$ ranges between $-\frac{1}{8}$ and 0.

We can now present the proof of the theorem. Let $M$ be some Turing machine, let $f_M$ be a map satisfying the conditions of Theorem 4 (replacing $\varepsilon$ by $\gamma$), and let $\bar{x}_0 \in \mathbb{R}^3$ be an approximation, with error $\varepsilon$, of some initial configuration $x_0$. Take also $\delta < 1/2$ and $\gamma > 0$ such that $2\gamma + \delta/2 \leq \varepsilon < 1/2$ (we suppose, without loss of generality, that $\delta/2 < \varepsilon$). This condition will be needed later. Consider the system of differential equations $z' = g_M(z, t)$ given by

$$z_1' = c_1(z_1 - f_M \circ \sigma^{[m]}(z_2))^3 \phi_1(t, z_1, z_2), \;\; z_2' = c_2(z_2 - \sigma^{[n]}(z_1))^3 \phi_2(t, z_1, z_2) \quad (10)$$

with initial conditions $z_1(0) = z_2(0) = \bar{x}_0$, where

$$\phi_1(t, z_1, z_2) = l_2\left(\theta(t), \frac{c_1}{\gamma}(z_1 - f_M \circ \sigma^{[m]}(z_2))^4 + \frac{c_1}{\gamma} + 10\right)$$
$$\phi_2(t, z_1, z_2) = l_2\left(\theta(-t), \frac{c_2}{\gamma}(z_2 - \sigma^{[n]}(z_1))^4 + \frac{c_2}{\gamma} + 10\right).$$

Because we want to show that the ODE $z' = g_M(z, t)$ simulates M in a robust manner, we also assume that an error of amplitude not exceeding $\delta$ is added to

8

the right side of the equations in (10). Our simulation variables are $z_1, z_2$ and the control functions are $\phi_1, \phi_2$. Since $\phi_1, \phi_2$ are analytic they cannot be constant on any open interval as in [12]. However, our construction guarantees that one of the control functions is kept close to zero, while the other one reaches a value close to 1. For instance, on $[0, 1/2]$ $|s(-t)| \leq 1/8$ and, by Lemma 2, $\phi_2$ is therefore less than $\gamma(c_2 \left\| z_2 - \sigma^{[n]}(z_1) \right\|_\infty^3)^{-1}$. This guarantees that $z_2'$ is sufficiently small on $[0, 1/2]$ and, therefore,

$$\left\| z_2(\tfrac{1}{2}) - x_0 \right\|_\infty < (\gamma + \delta)/2 + \varepsilon < \tfrac{1}{2}.$$

Hence, for $m$ large enough $\|\sigma^{[m]}(z_2) - x_0\| < \gamma$. Moreover, on some subinterval of $[0, 1/2]$ $s(t)$ is close to 1 and therefore $\phi_1$ is also close to 1. Thus, the behavior of $z_1$ is given by (9) and $\left\| z_1(\tfrac{1}{2}) - \theta(x_0) \right\|_\infty < 2\gamma + \delta/2 \leq \varepsilon$.

Now, for interval $[1/2, 1]$ the roles of $z_1$ and $z_2$ are switched. One concludes that if $n \in \mathbb{N}$ is chosen so that $\sigma^{[n]}(5\gamma/2 + \delta) < \gamma$, then $\|z_2(1) - f_M(x_0)\|_\infty < 2\gamma + \delta/2 \leq \varepsilon$. We can repeat this process for $z_1$ and $z_2$ on subsequent intervals, which shows that for $j \in \mathbb{N}$, if $t \in [j, j + \tfrac{1}{2}]$ then $\|z_2(t) - \theta^{[j]}(x_0)\|_\infty \leq \varepsilon$ as claimed. $\square$

Notice that all the functions we use in the proof above are analytic. Moreover, note that if we apply the error-contracting function $\sigma$ to $z_1$ we can make the error arbitrarily small. Therefore, Theorem 3 implies Theorem 2.

## 5    Final remarks

We showed that robust analytic maps and flows can simulate Turing machines, filling some existing gaps on the literature on this subject.

There are several connections of this work and previous results on continuous-time computational models. In particular, it is not difficult to verify [23] that the function $z$ in Theorem 3 is computable by Shannon's General Purpose Analog Computer (GPAC). Moreover, according to [16] $z$ also belongs to the (analytic) subclass $[0, 1, -1, U; COMP, I]$ of Moore's real recursive functions.

We proved lower computational bounds for analytic systems robust in the sense of Theorems 1 and 3. Can we show that the computational power of those systems lies in the realm of Turing computability, in analogy with the upper bounds in [5] for compact domains? We leave this question to the reader.

# References

1. Moore, C.: Unpredictability and undecidability in dynamical systems. Phys. Rev. Lett. **64** (1990) 2354–2357
2. Koiran, P., Cosnard, M., Garzon, M.: Computability with low-dimensional dynamical systems. Theoret. Comput. Sci. **132** (1994) 113–128
3. Siegelmann, H.T., Sontag, E.D.: On the computational power of neural networks. J. Comput. System Sci. **50** (1995) 132–150
4. Moore, C.: Finite-dimensional analog computers: Flows, maps, and recurrent neural networks. In Calude, C., Casti, J., Dinneen, M., eds.: 1st International Conference on Unconventional Models of Computation - UMC'98, Springer (1998) 59–71
5. Casey, M.: The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction. Neural Comp. **8** (1996) 1135–1178
6. Casey, M.: Correction to proof that recurrent neural networks can robustly recognize only regular languages. Neural Comp. **10** (1998) 1067–1069
7. Maass, W., Orponen, P.: On the effect of analog noise in discrete-time analog computations. Neural Comput. **10** (1998) 1071–1095
8. Pour-El, M.B., Richards, J.I.: The wave equation with computable initial data such that its unique solution is not computble. Adv. Math. **39** (1981) 215–239
9. Pour-El, M.B., Zhong, N.: The wave equation with computable initial data whose unique solution is nowhere computable. Math. Log. Quart. **43** (1997) 499–509
10. Weihrauch, K., Zhong, N.: Is wave propagation computable or can wave computers beat the Turing machine? Proc. London Math. Soc. **85** (2002) 312–332
11. Koiran, P., Moore, C.: Closed-form analytic maps in one and two dimensions can simulate universal Turing machines. Theoret. Comput. Sci. **210** (1999) 217–223
12. Branicky, M.S.: Universal computation and other capabilities of hybrid and continuous dynamical systems. Theoret. Comput. Sci. **138** (1995) 67–100
13. Campagnolo, M.L., Moore, C., Costa, J.F.: An analog characterization of the Grzegorczyk hierarchy. J. Complexity **18** (2002) 977–1000
14. Mycka, J., Costa, J.F.: Real recursive functions and their hierarchy. J. Complexity **20** (2004) 835–857
15. Campagnolo, M.L., Moore, C., Costa, J.F.: Iteration, inequalities, and differentiability in analog computers. J. Complexity **16** (2000) 642–660
16. Campagnolo, M.L.: The complexity of real recursive functions. In Calude, C.S., Dinneen, M.J., Peper, F., eds.: UMC'02. LNCS 2509. Springer (2002) 1–14
17. Hirsch, M.W., Smale, S.: Differential Equations, Dynamical Systems, and Linear Algebra. Academic Press (1974)
18. Guckenheimer, J., Holmes, P.: Nonlinear Oscillations, Dynamical Systems, and Bifurcation of Vector Fields. Springer (1983)
19. Viana, M.: Dynamical systems: moving into the next century. In Engquist, B., Schmid, W., eds.: Mathematics Unlimited - 2001 and Beyond. Springer (2001) 1167–1178
20. Pilyugin, S.Y.: Shadowing in Dynamical Systems. Springer (1999)
21. Grebogi, C., Poon, L., Sauer, T., Yorke, J., Auerbach, D.: Shadowability of chaotic dynamical systems. In: Handbook of Dynamical Systems. Volume 2. Elsevier (2002) 313–344
22. Atkinson, K.E.: An Introduction to Numerical Analysis. 2nd edn. John Wiley & Sons (1989)
23. Graça, D.S., Costa, J.F.: Analog computers and recursive functions over the reals. J. Complexity **19** (2003) 644–664