Computability with Polynomial Differential Equations

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

Manuel L. Campagnolo
DM/ISA, Technical University of Lisbon, 1349-017 Lisbon, Portugal
& SQIG/Instituto de Telecomunicações, Lisbon, Portugal

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

September 16, 2011

Abstract

In this paper, we show that there are Initial Value Problems defined with polynomial ordinary differential equations that can simulate universal Turing machines in the presence of bounded noise. The polynomial ODE defining the IVP is explicitly obtained and the simulation is performed in real time.

1 Introduction

Computational models based on natural phenomena have recently attracted a significant amount of interest. However, most of those models are essentially "hybrid" since they combine smooth dynamics with non-differentiable, or at least non-analytic, "clocks" to simulate the discrete dynamics of a Turing machine (e.g. [KCG94], [Koi96], [Sie99]). This situation, however, is not satisfactory: many nonlinear mathematical models arising from classical Physics (or more generally the natural sciences), as well as many of the fundamental examples in Dynamical Systems theory, are based on systems of ordinary differential equations (ODEs) with analytic, indeed very often polynomial, right hand-sides. This is the case of now classical systems like the van der Pol equation, the Lotka-Volterra system or the Lorenz equations [GH83], [HS74].

In this paper, we explore the computational capabilities of such polynomial dynamical systems. In [GCB05] it was shown that systems defined with analytic ODEs can simulate universal Turing machines, even if some perturbation is added to the system. Here we will present a detailed proof of a stronger result: we will prove that systems defined with *polynomial* ODEs are still Turing universal, even under the influence of some perturbation.

Several authors (e.g. [Moo90] [Bra95], [SS95]) have already proved that relatively simple discrete-time systems 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 [Moo90], [KCG94], [SS95]. 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 [Moo98].

The requirement of compactness has another drawback: it prevents systems capable of simulating an arbitrary Turing machine of exhibiting robustness to noise. Indeed, Casey [Cas96], [Cas98] has shown that in the presence of bounded analog noise, recurrent neural networks can only recognize regular languages. This result was later generalized in [MO98] 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, results of Pour-El, Richards, Weihrauch, and Zhong [PER81], [PEZ97], [WZ02] show that there exists a three-dimensional wave equation, with computable initial conditions, whose unique solution is not computable. Such behavior, however, is ruled out in the presence of noise [WZ02]. Recurrent analog neural networks are another known case where non-computable behavior can occur if real parameters are represented with infinite precision [SS95].

In this paper we will show that Turing machines can be simulated by flows defined by polynomial ODEs which are robust to perturbations (in a sense to be defined later). We will consider simulations on unbounded spaces. Our work is in some sense related to [KM99], where a constructive simulation of Turing machines using closed-form analytic maps is presented. However, in [KM99] only the discrete-time case is explored, and the question of how the presence of noise affects the computational power of the model is not discussed.

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 $\mathbb{R}^n$ [Bra95], [CMC02], [MC04]. While those flows can be infinitely differentiable, it has only recently been shown that they can be analytic [GCB05]. Since precise "clocks" cannot be defined with analytic functions, the proof in [GCB05] relies on maps robust to perturbations.

In the present paper we show that those robust maps may be suitably approximated by a system of polynomial differential equations $y' = p_M(t, y)$, where p_M is a polynomial, thus allowing robust simulation of any Turing machine M.

It is worthwhile to observe that our work is closely related to the wider topic of stability in dynamical systems. In fact, there has been a long tradition of considering only structurally stable systems [HS74] when modeling physical systems. The argument is that, due to measurement uncertainties, qualitative properties of a system should not change with small perturbations. Guckenheimer and Holmes [GH83] 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 [Via01]. In fact, one can relax the notion of stability and require robustness 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 in a manner that we will precise below.

The paper may be outlined as follows. In Section 2 we introduce the ideas and concepts related to simulations robust to perturbations. Sections 3, 4 and 5 provide tools that will be necessary later. In Sections 6 and 7 we prove in a constructive way the main results of this paper: each Turing machine can be simulated by an analytic map, or by polynomial ODEs even under the influence of (small) errors. 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 briefly describe some aspects of our error-robust simulation of Turing machines. For now, we will only be concerned with discrete-time simulations. Therefore we want to obtain a map that "captures" the behavior of the transition function. We will encode each configuration as 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

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

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 (except that they are non-zero for $i = -p, n$). 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

$$\begin{aligned} y_1 &= a_0 + a_1 10 + ... + a_n 10^n, \\ y_2 &= a_{-1} + a_{-2} 10 + ... + a_{-p} 10^{p-1}, \end{aligned} \tag{1}$$

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.

Let us introduce some useful notation. Let $\|(x_1, ..., x_n)\|_\infty = \max_{1 \le i \le n} |x_i|$ and $\|f\|_\infty = \sup_{x \in \mathbb{R}} |f(x)|$. If $f : A \to A$ is a function, then $f^{[k]}$ denotes its kth iterate (if $k = 0$, the 0th iterate is simply the identity function). We say that a real or complex function f is a closed-form function if it is elementary in the sense of analysis, that is, if it is a meromorphic function defined on some open subset of $\mathbb{R}$ or $\mathbb{C}$ that is contained in an elementary extension field of the field of rational functions $\mathbb{C}(z)$ [Rit48], [Ros72] (where 'elementary' corresponds to the introduction of the complex exponential and logarithm). This corresponds to the possibility of obtaining f from the elementary functions of analysis (e.g. rational functions, sin, tan, exp,...) through finitely many compositions, inversions and field operations. In this paper we shall deal exclusively with real functions.

We may now present the first main result of this paper which states that there exists a (globally analytic) closed-form function that robustly simulates the transition functions of any Turing machine.

Theorem 1 *Let* $\psi : \mathbb{N}^3 \to \mathbb{N}^3$ *be the transition function of a Turing machine* M*, under the encoding described above and let* $0 < \delta < \varepsilon < 1/2$*. Then* ψ *admits a globally analytic closed-form extension* $f_M : \mathbb{R}^3 \to \mathbb{R}^3$*, robust to perturbations in the following sense: for all* f *such that* $\|f - f_M\|_\infty \leq \delta$*, for all* $j \in \mathbb{N}$*, 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,*

$$\left\| f^{[j]}(\bar{x}_0) - \psi^{[j]}(x_0) \right\|_\infty \leq \varepsilon.$$

A few remarks are in order. First, and as noticed before, we implicitly assumed that if y is a halting configuration, then $\psi(y) = y$. Secondly, we notice that the upper bound ($\frac{1}{2}$) on ε results from the encoding we have chosen, which is over the integers. In fact, the bound is maximal with respect to that encoding. We also remark that the proof of the previous theorem is constructive and that f can be obtained by composing the following functions: polynomials, sin, cos, and arctan.

We now present the other main results.

Theorem 2 *Let* $\psi : \mathbb{N}^3 \to \mathbb{N}^3$ *be the transition function of a Turing machine* M*, under the encoding described above and let* $0 < \varepsilon < 1/4$*. There is a polynomial* $p_M : \mathbb{R}^{m+4} \to \mathbb{R}^{m+3}$*, with* $m \in \mathbb{N}$*, and a constant* $y_0 \in \mathbb{R}^m$ *such that the ODE* $z' = p_M(t, z)$ *simulates* M *in the following sense: for all* $x_0 \in \mathbb{N}^3$*, the solution* $z(t)$ *of the IVP defined by the previous ODE plus the initial condition* (x_0, y_0)*, defined for* $t_0 = 0$*, satisfies*

$$\left\| z_1(j) - \psi^{[j]}(x_0) \right\|_\infty \leq \varepsilon,$$

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

Indeed, we will prove the following "robust" version of Theorem 2.

Theorem 3 *Given the conditions of Theorem 2, there is a PIVP function (see definition on the next section)* $f_M : \mathbb{R}^7 \to \mathbb{R}^6$ *and a constant* $y_0 \in \mathbb{R}^3$ *such that the ODE* $z' = f_M(t, z)$ *robustly simulates* M *in the following sense: for all* g *satisfying* $\|g - f_M\|_\infty < 1/2$*, there is some* $0 < \eta < 1/2$ *such that for all* $(\bar{x}_0, \bar{y}_0) \in \mathbb{R}^6$ *satisfying* $\|(\bar{x}_0, \bar{y}_0) - (x_0, y_0)\|_\infty \leq \varepsilon$*, the solution* $z(t)$ *of*

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

satisfies, for all $j \in \mathbb{N}$ *and for all* $t \in [j, j + 1/2]$*,*

$$\left\| z_1(t) - \psi^{[j]}(x_0) \right\|_\infty \leq \eta,$$

where $z \equiv (z_1, z_2)$ *with* $z_1 \in \mathbb{R}^3$ *and* $z_2 \in \mathbb{R}^3$*.*

3 Polynomial differential equations

In this section we analyze some properties of solutions of Initial Value Problems (IVPs) defined with polynomial ODEs

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

where p is a vector of polynomials. In the remainder of the paper an IVP of the form (2) will be called a *polynomial* IVP, and its solution will be termed a *PIVP function.* We note that many of the usual functions from analysis are PIVP functions, e.g. polynomials, the trigonometric functions sin, cos, tan, or the exponential exp. We note also that PIVP functions obviously belong to the class of differentially algebraic (DA) functions [28], but not the other way around: DA functions are defined by *implicit* algebraic differential equations, and so a global 'normal form' ODE for DA functions is not in general defined.

Note that by the standard existence-uniqueness theory (see e.g. [CL55], [Lef65]), an IVP associated to a system of ODEs $x' = f(t, x)$ has a unique solution whenever f is continuous and locally Lipschitz with respect to the variable x. Since polynomials are globally analytic, these conditions are automatically satisfied for polynomial IVPs.

The following result is a strengthening of the elimination theorem of Rubel for differentially algebraic functions [RS85] to the present case of polynomial IVPs.

Theorem 4 *Consider the IVP*

$$\begin{cases} x' = f(t, x), \\ x(t_0) = x_0, \end{cases} \tag{3}$$

where $f : \mathbb{R}^{n+1} \to \mathbb{R}^n$ *and each component of* f *is a composition of polynomials and PIVP functions. Then there exist* $m \geq n$*, a polynomial* $p : \mathbb{R}^{m+1} \to \mathbb{R}^m$ *and a* $y_0 \in \mathbb{R}^m$ *such that the solution of* (3) *is given by the first* n *components of* $y = (y_1, ..., y_m)$*, where* y *is the solution of the polynomial IVP*

$$\begin{cases} y' = p(t, y), \\ y(t_0) = y_0. \end{cases} \tag{4}$$

Proof. Let $f = (f_1, \ldots, f_n)$. If every f_i, $i = 1, \ldots, n$, is a polynomial there is nothing to prove, so we suppose $k \leq n$ is the first integer such that f_k is not a polynomial. For simplicity we will handle only the case where the transcendence degree of f_k over the ring of polynomials is 1; the case of higher degree will follow from iterating the construction.

Suppose then that $f_k(t, x_1, \ldots, x_n) = g(p(t, x_1, ..., x_n))$, where p is a polynomial and g is the first component of the solution y of

$$\begin{cases} y' = q(t, y), \\ y(t_0) = y_0. \end{cases} \tag{5}$$

where q is a vector of polynomials in $\mathbb{R}^l$ not independent of y (thus ensuring that g is not trivially a polynomial in t). Define new variables $\bar{y}$ by performing the change of independent variable $t \mapsto p(t, x_1, ..., x_n)$ in (5), that is,

$$\bar{y}(t) = y(p(t, x_1, \ldots, x_n)).$$

Then

$$\begin{aligned} \frac{d\bar{y}}{dt} &= \frac{dy}{dt}(p(t, x_1, \ldots, x_n)) \; p'(t, x_1, ..., x_n) \\ &= q(p(t, x_1, ..., x_n), \bar{y}) \left(\sum_{i=1}^{n} \frac{\partial p}{\partial x_i} f_i(t, x) + \frac{\partial p}{\partial t} \right), \end{aligned} \tag{6}$$

subject to the initial condition $\bar{y}(t_0) = y(p(t_0, x_1(t_0), ..., x_n(t_0)))$.

We now consider the new IVP constructed by appending to (3) the IVP (6) and replacing the terms $g(p(t, x_1, ..., x_n))$ by the new variable $\bar{y}_1$. By this procedure we have replaced the IVP in $\mathbb{R}^n$ (3) with an IVP in $\mathbb{R}^{n+l}$ substituting the first non-polynomial term of f by a polynomial and increasing the system with l new variables defined by a polynomial IVP.

If the transcendence degree of f_k over the ring of polynomials is $d > 1$ we iterate this procedure d times. In this way we have effectively eliminated all non-polynomial terms up to component k by increasing the order of the system only with polynomial equations. Repeating this procedure with the variables labeled $k+1$ up to n (whenever necessary) we end up with a polynomial IVP satisfying the stated conditions.

■

Let us illustrate this theorem with an example. Consider the IVP

$$\begin{cases} x_1' = \sin^2 x_2 & x_1(0) = 0, \\ x_2' = x_1 \cos x_2 - e^{x_1+t} & x_2(0) = 0, \end{cases} \tag{7}$$

with solution (x_1, x_2). Since sin, cos, and exp are solutions of polynomial IVPs ((sin, cos) is the solution of $z_1 = z_2$, $z_2 = -z_1$, with $z_1(0) = 0, z_2(0) = 1$ and exp is the solution of $z' = z$ with $z(0) = 1$), Theorem 4 ensures that there is a polynomial IVP, with initial condition defined for $t_0 = 0$, whose solution is formed by x_1, x_2, and possibly other components. Since the proof of the theorem is constructive, we can derive this polynomial IVP. Indeed, one has $(\sin x_2)' = (\cos x_2)x_2'$ and $(\cos x_2)' = -(\sin x_2)x_2'$ and therefore $(\sin x_2, \cos x_2)$ is the solution of the IVP

$$\begin{cases} y_3' = y_4 x_2' \\ y_4' = -y_3 x_2' \end{cases} \Leftrightarrow \begin{cases} y_3' = y_4(x_1 \cos x_2 - e^{x_1+t}) \\ y_4' = -y_3(x_1 \cos x_2 - e^{x_1+t}) \end{cases}$$
$$\Leftrightarrow \begin{cases} y_3' = y_4(x_1 y_4 - e^{x_1+t}) \\ y_4' = -y_3(x_1 y_4 - e^{x_1+t}) \end{cases}$$

with initial condition $y_3(0) = \sin x_2(0) = 0$ and $y_4(0) = 1$. It remains to eliminate the term e^{x_1+t}. But this function is the solution of the IVP

$$y_5' = y_5\,(x_1' + t') = y_5(\sin^2 x_2 + 1) = y_5(y_3^2 + 1),$$

with initial condition $y_5(0) = e^{x_1(0)+0} = 1$. Therefore the solution of (7) is formed by the first two components of the solution of the polynomial IVP

$$\begin{cases} y_1' = y_3^2 & y_1(0) = 0, \\ y_2' = y_1 y_4 - y_5 & y_2(0) = 0, \\ y_3' = y_4(y_1 y_4 - y_5) & y_3(0) = 0, \\ y_4' = -y_3(y_1 y_4 - y_5) & y_4(0) = 1, \\ y_5' = y_5(y_3^2 + 1) & y_5(0) = 1. \end{cases}$$

4 Preliminary results

This section is devoted to the presentation of auxiliary results that 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 on 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. [Atk89, pp. 176–182]), one may take

$$\omega(x) = a_0 + a_5 \cos(\pi x) + \left(\sum_{j=1}^{4} a_j \cos\left(\frac{j\pi x}{5}\right) + b_j \sin\left(\frac{j\pi x}{5}\right) \right), \qquad (8)$$

where $a_0, ..., a_5, b_1, ..., b_4$ are computable coefficients that can be explicitly obtained by solving a system of linear equations (the number of variables should be equal to the number of values to be interpolated). Actually the number of a_i's should agree with those of b_i's, yielding 11 interpolated values. Since we only need 10 such values, one can take $b_5 = 0$ and still obtain a trigonometric function ω satisfying $\omega(i) = i$, for $i = 0, 1, ..., 9$.

It is easy to see that ω is uniformly continuous in $\mathbb{R}$ (ω has period 10, is continuous in the interval $[0, 10]$ and $\omega(0) = \omega(10)$). Hence, for every $\varepsilon \in (0, 1/2)$, there will be some $\zeta_\varepsilon > 0$ satisfying

$$\forall n \in \mathbb{N},\ x \in [n - \zeta_\varepsilon, n + \zeta_\varepsilon] \quad \Rightarrow \quad |\omega(x) - n \bmod 10| \leq \varepsilon. \qquad (9)$$

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 (cf. Fig. 1)

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

The function σ is a uniform contraction in a neighborhood of integers:

Proposition 5 *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, $\forall \delta \in [-\varepsilon, \varepsilon]$, $|\sigma(n + \delta) - n| < \lambda_\varepsilon \delta$.*

Proof. It is sufficient to consider the case where $n = 0$. Because σ is odd, we only study σ in the interval $[0, \varepsilon]$. Let $g(x) = \sigma(x)/x$. This function is strictly increasing in $(0, 1/2]$. Then, noting that $g(1/2) = 1$ and $\lim_{x \to 0} g(x) = 1 - 0.4\pi \approx -0.256637$, we conclude that there exists some $\lambda_\varepsilon \in (0, 1)$ such that $|\sigma(x)| < \lambda_\varepsilon |x|$ for all $x \in [-\varepsilon, \varepsilon]$. ■

Remark 6 *For the rest of this paper we suppose that $\varepsilon \in [0, 1/2)$ is fixed and that λ_ε is the respective contracting factor given by Lemma 5. For instance, we can take $\lambda_{1/4} = 0.4\pi - 1 \approx 0.2566371$.*

The function σ 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, *e.g.* when using expressions that depend on the variables coding the tape contents. 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 $a \in \{0, 1\}$ (cf. Fig. 2).

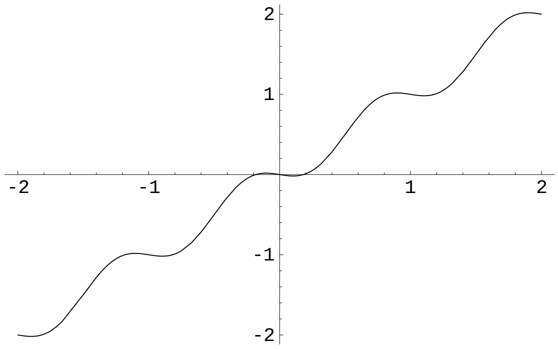

Figure 1: Graphical representation of the function σ.

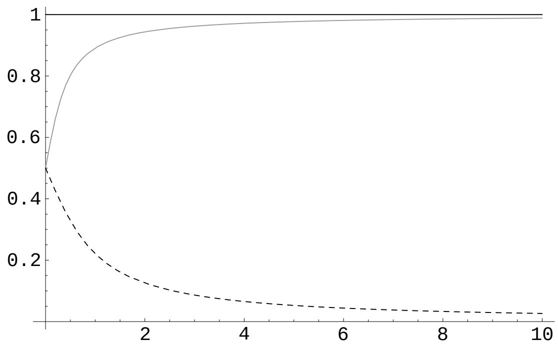

Figure 2: Graphical representation of the function l_2. The dashed line represents $l_2(0.2, y)$ while the gray line represents $l_2(1.2, y)$.

Lemma 7 $\left|\frac{\pi}{2} - \arctan x\right| < \frac{1}{x}$ *for* $x \in (0, \infty)$.

Proof. Let $f(x) = \frac{1}{x} + \arctan x - \frac{\pi}{2}$. It is easy to see that f is decreasing in $(0, \infty)$ and that $\lim_{x\to\infty} f(x) = 0$. Therefore $f(x) > 0$ for $x \in (0, \infty)$ and the result holds. ■

Lemma 8 $\left|\frac{\pi}{2} + \arctan x\right| < \frac{1}{|x|}$ *for* $x \in (-\infty, 0)$.

Proof. Take $f(x) = \frac{1}{x} + \arctan x + \frac{\pi}{2}$ and proceed as in Lemma 7. ■

Lemma 9 *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 obtain* $|a - l_2(\bar{a}, y)| < 1/y$.

Proof.

1. Consider $a = 0$. Then $\bar{a} - 1/2 \leq -1/4$ implies $|4y(\bar{a} - 1/2)| \geq y$. Therefore, by Lemma 8,

$$\left|\frac{\pi}{2} + \arctan(4y(\bar{a} - 1/2))\right| < \frac{1}{|4y(\bar{a} - 1/2)|} \leq \frac{1}{y}.$$

Moreover, multiplying the last inequality by $1/\pi$ and noting that $\frac{1}{\pi y} < \frac{1}{y}$, it follows that $|a - l_2(\bar{a}, y)| < 1/y$.

2. Consider $a = 1$. Remark that $\bar{a} - 1/2 \geq 1/4$ and proceed as above, using Lemma 7 instead of Lemma 8.

■

We denote below, for any $x \in \mathbb{R}$, $\lceil x \rceil = \min\{k \in \mathbb{Z} : k \geq x\}$.

Proposition 10 *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 $|a - l_3(\bar{a}, y)| < 1/y$.

Proof. Let us start by noticing that for all $x, y \in \mathbb{R}$ for which $l_2(x, y)$ is defined, we have $0 < l_2(x, y) < 1$. Consider the case where $a = 0$, $\bar{a} \in [-1/4; 1/4]$, i.e. $\varepsilon \leq 1/4$. Then $\left|(\sigma(\bar{a}) - 1)^2 - 1\right| < 1/4$, and by the previous lemma,

$$1 - 1/y < l_2((\sigma(\bar{a}) - 1)^2, y) < 1.$$

Similarly, we conclude

$$-1 < 2l_2(\bar{a}/2, y) - 1 < -1 + 2/y.$$

Since $y \geq 2$, this implies

$$-1 < l_2((\sigma(\bar{a}) - 1)^2, y)(2l_2(\bar{a}/2, y) - 1) < (1 - 1/y)(-1 + 2/y),$$

or

$$0 < l_2((\sigma(\bar{a}) - 1)^2, y)(2l_2(\bar{a}/2, y) - 1) + 1 < 3/y.$$

Hence, for $a = 0$, $|a - l_3(\bar{a}, y)| < 1/y$. Proceeding in the same way for $a = 1, 2$ and $\varepsilon \leq 1/4$), the same result follows.

It remains to consider the more general case $|a - \bar{a}| \leq \varepsilon$. In that case just take $d = \lceil -\log(4\varepsilon)/\log \lambda_\varepsilon \rceil$ and apply d times the function σ to $\bar{a}$, it follows that $\left|a - \sigma^{[d]}(\bar{a})\right| \leq 1/4$ and we fall back in the previous case (use $\sigma^{[d]}(\bar{a})$ instead of $\bar{a}$). ■

5 Polynomial interpolation

In this paper we use polynomial interpolation to simulate a given Turing machine. Moreover, all quantities involved in an error-free simulation are elements of $\mathbb{N}$. In particular, the states of a Turing machine will be represented by elements of $\{1, ..., m\}$ and each symbol of the tape will be considered as an element of $\{0, 1, ..., k - 1\}$, where $m, k \in \mathbb{N}$. Because we want to derive a simulation of Turing machines robust to (small) perturbations, we will not use exact values, but allow an error less than $\varepsilon > 0$ on all these quantities. Special care is thus needed to ensure that the error does not amplify along the simulation. In this section we study the propagation of errors throughout the iteration of polynomial maps defined via polynomial interpolation.

Let M be a Turing machine, and take y as the symbol being currently read and q as the current state. Consider also the following functions:

$$Q_i(x) = \prod_{\substack{j=1 \\ j\neq i}}^{m} \frac{(x-j)}{(i-j)} = \begin{cases} 0, & \text{if } x = 1, ..., i-1, i+1, ..., m, \\ 1, & \text{if } x = i, \end{cases}$$

and

$$S_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{k-1} \frac{(x-j)}{(i-j)} = \begin{cases} 0, & \text{if } x = 0, ..., i-1, i+1, ..., k-1, \\ 1, & \text{if } x = i. \end{cases}$$

Suppose that on state i and symbol j, the state of the next configuration is $q_{i,j}$. Suppose also that $|q_{i,j}| \leq N$ for some suitable constant N (*e.g.*, in this case, we may take $N = m$). Then the state that follows state q and symbol y can be given by

$$q_{next} = \sum_{i=0}^{k-1} \sum_{j=1}^{m} S_i(y) Q_j(q) q_{i,j}. \tag{10}$$

A similar procedure may be used to determine the next symbol to be written and the next move. The main problem is that we don't have access to the exact values of q or y, but rather to some approximations $\bar{q}$ and $\bar{y}$, respectively, thereby obtaining $\bar{q}_{next}$. Hence, we want to increase the precision of $\bar{q}$ and $\bar{y}$ so that $|q_{next} - \bar{q}_{next}| < \varepsilon$ (recall that ε is an upper bound for the error allowed during a computation). To achieve this precision, we only need to compute each term in the sum given in (10) with an error less than $\varepsilon/(km)$. This is verified when

$$|S_i(\bar{y})Q_j(\bar{q}) - S_i(y)Q_j(q)| < \frac{\varepsilon}{kmN}, \tag{11}$$

since this condition implies

$$|S_i(\bar{y})Q_j(\bar{q})q_{i,j} - S_i(y)Q_j(q)q_{i,j}| < \frac{\varepsilon}{km}.$$

Lemma 11 *Let $n \in \mathbb{N}$ and $(x_1, \ldots, x_n), (y_1, \ldots, y_n) \in \mathbb{R}^n$. Let $K > 0$ be such that $|x_i| \leq K, |y_i| \leq K$ for $i = 1, \ldots, n$. Then*

$$|x_1 \ldots x_n - y_1 \ldots y_n| \leq (|x_1 - y_1| + \ldots + |x_n - y_n|) K^{n-1}.$$

Proof. The Lemma is trivial for $n = 1$. Suppose it holds for $n \in \mathbb{N}$. Then

$$\begin{aligned} &|x_1 \ldots x_{n+1} - y_1 \ldots y_{n+1}| \\ &\quad \leq |x_1 \ldots x_{n+1} - x_1 \ldots x_n y_{n+1}| + |x_1 \ldots x_n y_{n+1} - y_1 \ldots y_{n+1}| \\ &\quad = |x_1 \ldots x_n| \, |x_{n+1} - y_{n+1}| + |x_1 \ldots x_n - y_1 \ldots y_n| \, |y_{n+1}| \\ &\quad \leq K^n \, |x_{n+1} - y_{n+1}| + (|x_1 - y_1| + \ldots + |x_n - y_n|) K^{n-1} K \\ &\quad = (|x_1 - y_1| + \ldots + |x_n - y_n| + |x_{n+1} - y_{n+1}|) K^n, \end{aligned}$$

which proves the statement by induction. ■

Note that (11) is satisfied if

$$\left| \prod_{\substack{r=0 \\ r\neq i}}^{k-1} (\bar{y} - r) \prod_{\substack{s=1 \\ s\neq j}}^{m} (\bar{q} - s) - \prod_{\substack{r=0 \\ r\neq i}}^{k-1} (y - r) \prod_{\substack{s=1 \\ s\neq j}}^{m} (q - s) \right| < \frac{\varepsilon}{kmN}, \tag{12}$$

(because $\left|\prod_{\substack{r=0\\r\neq i}}^{k-1}(i-r)\prod_{\substack{s=1\\s\neq j}}^{m}(j-s)\right|\geq 1$). Take $K=\max\{k-1+\varepsilon,m-1+\varepsilon\}$. Applying Lemma 11, (12) will hold for

$$(k-1)|y-\bar{y}|+(m-1)|q-\bar{q}|<\frac{\varepsilon}{kmNK^{m+k-3}}.$$

Then, in order to have $|q_{next}-\bar{q}_{next}|\leq\varepsilon$, it is sufficient to take $\bar{y}$ and $\bar{q}$ with an error less than

$$|y-\bar{y}|,|q-\bar{q}|<\frac{\varepsilon}{kmNK^{m+k-3}(k+m-2)}. \tag{13}$$

To achieve this, and supposing that $\bar{y}$ and $\bar{q}$ are initially given with an error less than ε, we only have to apply j times the error-correcting function σ so that $\sigma^{[j]}(\bar{y})$ and $\sigma^{[j]}(\bar{q})$ have greater precision than the bound in (13). This condition holds for every j satisfying

$$j\geq\left\lceil\frac{\log(kmNK^{m+k-3}(k+m-2))}{-\log\lambda_{\varepsilon}}\right\rceil,$$

where λ_{ε} is given by Proposition 5.

6 Robust simulations of Turing machines with analytic maps

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

$$\|(y_1,y_2,q)-(\bar{y}_1,\bar{y}_2,\bar{q})\|_\infty\leq\varepsilon\quad\Rightarrow\quad\|\psi(y_1,y_2,q)-h_M(\bar{y}_1,\bar{y}_2,\bar{q})\|_\infty\leq\varepsilon. \tag{14}$$

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 ω is given by (8). We must show that the effect of the error present in $\bar{y}_1$ can be controlled. Since $|y_1-\bar{y}_1|\leq\varepsilon$,

$$|a_0-\omega\circ\sigma^{[l]}(\bar{y}_1)|\leq\varepsilon,\qquad\text{with } l=\left\lceil\left|\frac{\log(\zeta_\varepsilon/\varepsilon)}{\log\lambda_\varepsilon}\right|\right\rceil, \tag{15}$$

   where ζ_ε is given by (9). Then pick $\bar{y}=\omega\circ\sigma^{[l]}(\bar{y}_1)$ as an approximation of the symbol currently being read. Similarly, $\omega\circ\sigma^{[l]}(\bar{y}_2)$ gives an approximation of a_{-1}, with error bounded by ε.

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. In the absence of error on y and q

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

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 (cf. Section 5)

$$\bar{q}_{next} = \sum_{i=0}^{9} \sum_{j=1}^{m} \left(\prod_{\substack{r=0 \\ r \neq i}}^{9} \frac{(\sigma^{[v]}(\bar{y})-r)}{(i-r)} \right) \left(\prod_{\substack{s=1 \\ s \neq j}}^{m} \frac{(\sigma^{[v]}(\bar{q})-s)}{(j-s)} \right) q_{i,j}, \qquad (16)$$

with

$$v = \left\lceil \frac{\log(10m^2 K^{m+7}(m+8))}{-\log \lambda_\varepsilon} \right\rceil, \qquad K = \max\{19/2, m - 1/2\},$$

which yields $|\bar{q}_{next} - q_{next}| \leq \varepsilon$.

3. **Determine the symbol to be written on the tape.** Using a construction similar to the previous case, the symbol to be written, s_{next}, can be approximated with precision ε, 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 2 and 3, therefore obtaining $|h_{next} - \bar{h}_{next}| \leq \varepsilon$.

5. **Update the tape contents.** In the absence of error, the "next value" of y_1, y_1^{next}, is given by polynomial interpolation as a function of y_1, y_2, s_{next} and h_{next} (recall that $a_0 = \omega(y_1)$):

$$y_1^{next} = (10(y_1 + s_{next} - \omega(y_1)) + \omega(y_2)) \frac{(1-h_{next})(2-h_{next})}{2}$$
$$+ (y_1 + s_{next} - \omega(y_1)) h_{next}(2 - h_{next}) + \frac{y_1 - \omega(y_1)}{10} \frac{h_{next}(1-h_{next})}{-2}.$$

To make the simulation robust, we define instead 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_1 be a "sufficiently good" approximation of h_{next}, yet to be determined. Then, y_1^{next} can be approximated by

$$\bar{y}_1^{next} = \bar{P}_1 \, \tfrac{1}{2}(1-H_1)(2-H_1) + \bar{P}_2 \, H_1(2-H_1) + \bar{P}_3 \, (-\tfrac{1}{2}) H_1(1-H_1) \quad (17)$$

with

$$\begin{aligned}
\bar{P}_1 &= 10(\sigma^{[d+4]}(\bar{y}_1) + \sigma^{[d+4]}(\bar{s}_{next}) - \sigma^{[d+4]}(\bar{y})) + \sigma^{[d+2]} \circ \omega \circ \sigma^{[l]}(\bar{y}_2), \\
\bar{P}_2 &= \sigma^{[d+2]}(\bar{y}_1) + \sigma^{[d+2]}(\bar{s}_{next}) - \sigma^{[d+2]}(\bar{y}), \\
\bar{P}_3 &= \tfrac{1}{10}\left(\sigma^{[d+1]}(\bar{y}_1) - \sigma^{[d+1]}(\bar{y})\right),
\end{aligned}$$

where d is given by Proposition 10 and l is given by (15), as we show below.

First, notice that when exact values are used and $H_1 = h_{next}$, one has $\bar{y}_1^{next} = y_1^{next}$. However, $\bar{P}_1$ in (17) depends on $\bar{y}_1$, which is not a bounded value. If we would simply take $H_1 = \bar{h}_{next}$, the error of the term $(1 - H_1)(2 - H_1)/2$ will be arbitrarily amplified when multiplied by $\bar{P}_1$. Hence, H_1 must be a sharp estimate of h_{next}, proportional to $\bar{y}_1$.

Our goal is to define H_1 such that it approximates h_{next} with error at most δ, i.e. $|H_1 - h_{next}| \leq \delta$. Let $P_1 = 10(y_1 + s_{next} - \omega(y_1)) + \omega(y_2)$, $P_2 = y_1 + s_{next} - \omega(y_1)$, $P_3 = (y_1 - \omega(y_1))/10$. To simplify the construction, we first suppose that $d = 0$, i.e. $\varepsilon \leq 1/4$. Since $h_{next} \in \{0, 1, 2\}$,

$$\begin{aligned}
&|y_1^{next} - \bar{y}_1^{next}| \\
&\qquad \leq |P_1 - \bar{P}_1| + |P_2 - \bar{P}_2| + |P_3 - \bar{P}_3| + |4\bar{P}_1\delta| + |7\bar{P}_2\delta| + |3\bar{P}_3\delta| \\
&\qquad < 0.112 + |4\bar{P}_1\delta| + |7\bar{P}_2\delta| + |3\bar{P}_3\delta|.
\end{aligned}$$

In the last inequality, $|P_1 - \bar{P}_1| < 0.049$, $|P_2 - \bar{P}_2| < 0.05$, $|P_3 - \bar{P}_3| < 0.013$ (note that approximate quantities have an error bounded by $1/4$; applying $\sigma^{[i]}$ to one of this term reduces the error to $\lambda_{1/4}^{i}/4$, where $i \in \mathbb{N}$). Moreover, in the interpolation terms involving h_{next}, we use the assumption that $h_{next} \in \{0, 1, 2\}$ to eliminate h_{next}. For instance $|(1 - h_{next})(2 - h_{next})| \leq 2$. Then adding up the resulting terms, we get the 4, 7, and 3 present in the equation above. Hence, $|y_1^{next} - \bar{y}_1^{next}| < 1/4$ can be achieved if $|\bar{P}_1\delta|, |\bar{P}_2\delta|, |\bar{P}_3\delta| < 0.009$. Taking $|\bar{P}_1\delta| < 0.009$, one has $|\delta| < 0.009/|\bar{P}_1|$ or equivalently, considering (1) and the the fact that $P_1 < 10^{n+2} - 1$ and $|P_1 - \bar{P}_1| < 0.049$ imply $|\bar{P}_1| < 10^{n+2}$, where n is the number of symbols written on the right half of the tape encoded by y_1, one has $|\delta| < 9 \times 10^{-n-5}$. Since $(1.2 \times 10^{n+4})^{-1} < 9 \times 10^{-n-5}$, we may assume that it should be

$$|\delta| < \frac{1}{1.2 \times 10^{n+4}}.$$

But $\bar{y}_1 > 10^n - 1/2$ implies $12000(\bar{y}_1 + 1/2) > 1.2 \times 10^{n+4}$. Therefore, we just have to take

$$|\delta| < \frac{1}{12000\,(\bar{y}_1 + 1/2)}.$$

Using a similar procedure for the inequalities $|\bar{P}_2\delta| < 0.009$ and $|\bar{P}_3\delta| < 0.009$, one reaches the same bound.

So far we have seen that to guarantee that the error $|y_1^{next} - \bar{y}_1^{next}|$ is less than ε, H_1 has to approximate h_{next} within the above bound, which depends on $\bar{y}_1$. This can be achieved with

$$H_1 = l_3(\bar{h}_{next}, 12000\,(\bar{y}_1 + 1/2) + 2),$$

as shown in Proposition 10. Notice that the extra 2 in the second argument of l_3 is only needed to ensure that $12000\,(\bar{y}_1 + 1/2) + 2 \geq 2$, as required by Proposition 10.

We can generalize this result to $\varepsilon < 1/2$ by applying d times the function σ to all the terms in the expressions of $\bar{P}_1$, $\bar{P}_2$ and $\bar{P}_3$, where d is given by Proposition 10. Therefore, $\bar{y}_1^{next}$ can be defined by (17).

Proceeding in the same manner for $\bar{y}_2^{next}$, one may take

$$\bar{y}_2^{next} = \bar{Q}_1 \frac{(1-H_2)(2-H_2)}{2} + \bar{Q}_2 H_2(2-H_2) + \bar{Q}_3 \frac{H_2(1-H_2)}{-2}, \quad (18)$$

where

$$H_2 = l_3(\bar{h}_{next}, 12000\,(\bar{y}_2 + 1/4) + 2), \quad \bar{Q}_3 = 10\sigma^{[d+4]}(\bar{y}_2) + \sigma^{[d+2]}(\bar{s}_{next}),$$
$$\bar{Q}_1 = \frac{\sigma^{[d+1]}(\bar{y}_2) - \sigma^{[d+1]} \circ \omega \circ \sigma^{[l]}(\bar{y}_2)}{10}, \qquad \bar{Q}_2 = \sigma^{[d+2]}(\bar{y}_2).$$

We then conclude that $|\bar{y}_1^{next} - y_1^{next}| < \varepsilon$ and $|\bar{y}_2^{next} - y_2^{next}| < \varepsilon$.

To finish the proof of Theorem 12, we put together the maps described above and define $h_M : \mathbb{R}^3 \to \mathbb{R}^3$ as $h_M(\bar{y}_1, \bar{y}_2, \bar{q}) = (\bar{y}_1^{next}, \bar{y}_2^{next}, \bar{q}_{next})$. ■

We shall now prove the main results of this paper.

Proof of Theorem 1. Let $0 \leq \delta < \varepsilon$. Then, using Theorem 12, one can find a map h_M such that (14) 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,

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

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

$$\begin{aligned} \|f(\bar{x}_0) - \psi(x_0)\|_\infty &\leq \|f(\bar{x}_0) - f_M(\bar{x}_0)\|_\infty + \|f_M(\bar{x}_0) - \psi(x_0)\|_\infty \\ &\leq \delta + (\varepsilon - \delta) = \varepsilon. \end{aligned}$$

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

7 Robust simulations of Turing machines with polynomial ODEs

We now show how the previous constructions can be implemented with ODEs defined by PIVP functions. By Theorem 4, this ODE is equivalent to a polynomial one.

Proof of Theorem 3. We adapt the construction in [Bra95] to simulate the iteration of the transition function of a Turing Machine with ODEs, using our Theorem 1 to generalize Branicky's construction to analytic and robust flows, defined with a polynomial IVP. In particular, since the function f_M can be obtained by composing polynomials, sin, cos, and arctan, we will show that there is an IVP $y' = h(y,t)$, $y(0) = (x_0, k_h)$ simulating in a robust manner the given Turing machine, where each component of h is defined by composing

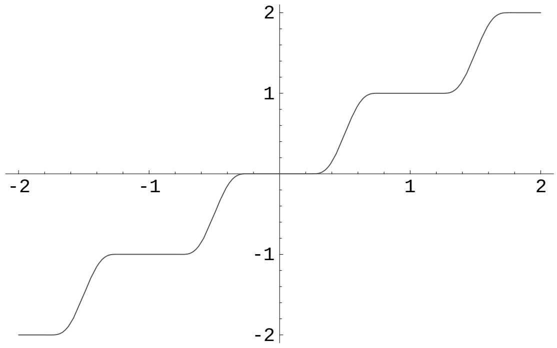

Figure 3: Graphical representation of the function r.

polynomials, sin, cos, and arctan. Then the result follows as a corollary of Theorem 4.

In a first approach, we present Branicky's idea following [Cam02b, p. 37], where an integer function can be iterated by a system of ODEs defined with functions that can be arbitrarily smooth (but still non-analytic). Before presenting the whole procedure, we need some auxiliary functions. In particular, let $\theta_j : \mathbb{R} \to \mathbb{R}$, $j \in \mathbb{N} - \{0, 1\}$ be the function defined by

$$\theta_j(x) = 0 \text{ if } x < 0, \quad \theta_j(x) = x^j \text{ if } x \geq 0.$$

This function can be seen [CMC00] as a C^{j-1} version of Heaviside's step function $\theta(x)$, where $\theta(x) = 1$ for $x \geq 0$ and $\theta(x) = 0$ for $x < 0$. Consider also the integer part function $r : \mathbb{R} \to \mathbb{R}$ defined by

$$r(0) = 0, \ r'(x - 1/4) = c_j \theta_j(-\sin 2\pi x), \tag{19}$$

where $c_j = \left(\int_0^1 \theta_j(-\sin 2\pi x) dx \right)^{-1}$. The function r satisfies $r(x) = n$, whenever $x \in [n - 1/4, n + 1/4]$, for all integer n, as illustrated in Fig. 3. Now consider the following system of ODEs

$$\begin{cases} z_1' &= \lambda_j (\tilde{f}(r(z_2)) - z_1)^3 \theta_j(\sin 2\pi t), \\ z_2' &= \lambda_j (r(z_1) - z_2)^3 \theta_j(-\sin 2\pi t), \end{cases} \tag{20}$$

where $\tilde{f} : \mathbb{R} \to \mathbb{R}$ is an arbitrary extension to the reals of function $f : \mathbb{N} \to \mathbb{N}$, $z_1(0) = z_2(0) = x_0 \in \mathbb{N}$ and $\lambda_j > 8c_j$. Notice that, if $k \in \mathbb{N}$, then for $t \in [k, k + 1/2]$, $z_2'(t) = 0$, and for $t \in [k - 1/2, k]$, $z_1'(t) = 0$ (*cf.* Fig. 4). According to the construction presented in the proof of Prop. 3.4.2 from [Cam02b], we conclude that $\left| f^{[k]}(x_0) - z_2(k) \right| < 1/4$ for all $k \in \mathbb{N}$.

Now, if we want to iterate f with analytic functions, using a system similar to (20), we cannot allow z_1' and z_2' to be 0 in half-unit intervals. Instead, we allow them to be very close to zero, which will add some errors to the system (20). Therefore, at time $t = 1$ both variables have values close to $\psi(x_0)$. But Theorem 1 shows that there exists some analytic function robust to errors that simulates ψ. This allows us to repeat the process an arbitrary number of times, keeping the error under control.

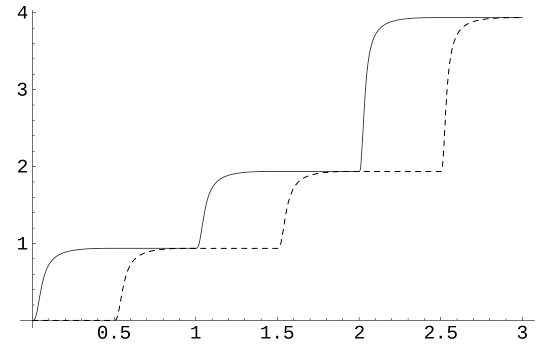

Figure 4: Simulation of the iteration of the map $f(n) = 2^n$ via ODEs. The solid line represents the variable z_1 and the dashed line represents z_2.

We begin with some preliminary results about the introduction of perturbations in (20). Let $\phi : \mathbb{R} \to \mathbb{R}_0^+$ be some function satisfying $\int_0^{1/2} \phi(t)dt > 0$. It is not difficult to see that Branicky's simulation relies on an ODE of the form

$$z' = -c(z-b)^3\phi(t), \tag{21}$$

where $c \geq (2\gamma^2 \int_0^{1/2} \phi(t)dt)^{-1}$ and $\gamma > 0$ is the "targeting error". For instance, in the equation governing z_1' in (20), $\gamma < 1/4$, $c = \lambda_j$, and $\phi(t) = \theta_j(\sin 2\pi t)$. Indeed, since (21) is a separable equation, it is easily shown that $|z(1/2) - b| < \gamma < 1/4$. However, if we want to know what happens in a perturbed version of Branicky's simulation, it is important to understand what happens in the following perturbed version of (21)

$$z' = -c(z - \bar{b}(t))^3\phi(t) + E(t), \tag{22}$$

where $\left|\bar{b}(t) - b\right| \leq \rho$ and $|E(t)| \leq \delta$, for $\rho, \delta \geq 0$, where $1/2 > \delta \geq \|g - f_M\|_\infty$ (cf. the statement of the theorem). Let $\bar{z}$ be the solution of this new ODE, with initial condition $\bar{z}(0) = \bar{z}_0$ and let z_+, z_- be the solutions of $z' = -c(z - b - \rho)^3\phi(t) + \delta$ and $z' = -c(z - b + \rho)^3\phi(t) - \delta$, with initial conditions $z_+(0) = z_{+,0}$ and $z_-(0) = z_{-,0}$ respectively, where $z_{+,0}, z_{-,0} \in \mathbb{R}$ satisfy $z_{+,0} \geq \bar{z}_0 \geq z_{-,0}$. By a standard differential inequality from the basic theory of ODEs (see e.g. [HW95], Appendix T), it follows that $z_-(t) \leq \bar{z}(t) \leq z_+(t)$ for all $t \in \mathbb{R}$. In fact, a detailed analysis of the dynamics of the equation in $[0, 1/2]$ shows that, in this interval, $|b - z_+(1/2)| \leq \gamma + \rho + \delta/2$, $|b - z_-(1/2)| \leq \gamma + \rho + \delta/2$. Therefore, we conclude that $|\bar{z}(1/2) - b| < \gamma + \rho + \delta/2$ regardless of the initial condition at $t = 0$.

Now, in order to perform Branicky's simulation, $z_1'(t)$ should be brought very close to zero whenever $t \in [1/2, 1]$. This can be done with the help of the function s 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 (ranging in $[3/4, 1]$ for $x \in [0.16, 0.34]$) and on $[1/2, 1]$ s ranges between $-\frac{1}{8}$ and 0. Therefore, we use the function

$W_0 : \mathbb{R} \times \mathbb{R}^+ \to [0,1]$ defined by

$$W_0(t,y) = l_2(s(t), y),$$

to replace $\phi(t) = \theta_j(\sin 2\pi t)$ in (20), since $\int_0^{1/2} W_0(t,y) > 3/4 \times (0.34 - 0.16) > 0$ (here we assume $y \geq 4$) and we always have $|W_0(t,y)| < 1/y$ for $t \in [1/2, 1]$ (i.e. y allows the control of the error committed when $z_1'(t)$ is brought to zero).

We can now present the proof of the theorem. Let $\gamma > 0$ be such that $2\gamma + \delta/2 \leq \varepsilon < 1/4$ (we suppose, without loss of generality, that $\delta/2 < \varepsilon$), f_M be a map satisfying the conditions of Theorem 12 (replacing ε by γ), and let $\bar{x}_0 \in \mathbb{R}^3$ be an approximation, with error ε, of some initial configuration x_0. Consider the system of differential equations $z' = g_M(t,z)$ given by

$$\begin{aligned} z_1' &= \lambda_1 (z_1 - f_M \circ \sigma^{[n_1]}(z_2))^3 \, \phi_1(t, z_1, z_2), \\ z_2' &= \lambda_2 (z_2 - \sigma^{[n_2]}(z_1))^3 \, \phi_2(t, z_1, z_2), \end{aligned} \tag{23}$$

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

$$\begin{aligned} \phi_1(t, z_1, z_2) &= l_2 \left(s(t), \tfrac{\lambda_1}{\gamma}(z_1 - f_M \circ \sigma^{[n_1]}(z_2))^4 + \tfrac{\lambda_1}{\gamma} + 10 \right), \\ \phi_2(t, z_1, z_2) &= l_2 \left(s(-t), \tfrac{\lambda_2}{\gamma}(z_2 - \sigma^{[n_2]}(z_1))^4 + \tfrac{\lambda_2}{\gamma} + 10 \right). \end{aligned}$$

Because we want to show that the ODE $z' = g_M(t,z)$ simulates M in a robust manner, we also assume that an error of absolute value not exceeding δ is added to the right hand-side of the equations in (23). Our simulation variables are z_1, z_2 and the control functions are ϕ_1, ϕ_2. Since ϕ_1, ϕ_2 are analytic they cannot be constant on any open interval as in [Bra95]. However, our construction guarantees that one of the control functions is kept close to zero, while the other reaches a value close to 1. For instance, on $[0, 1/2]$ $|s(-t)| \leq 1/8$ and, by Lemma 9 (note that, for all $x \in \mathbb{R}$, $|x|^3 \leq x^4 + 1$), ϕ_2 is less than $\gamma(\lambda_2 \left\| z_2 - \sigma^{[n_2]}(z_1) \right\|_\infty^3)^{-1}$. This ensures that $\|z_2'(t)\|_\infty \leq \gamma$ for $t \in [0, 1/2]$. Since the initial condition has error bounded by ε, z_2' in (23) is perturbed by an amount not exceeding δ, one has

$$\|z_2(t) - x_0\|_\infty < (\gamma + \delta)/2 + \varepsilon = \eta < \tfrac{1}{2} \qquad \text{for } t \in [0, 1/2].$$

Hence, for n_1 large enough (depending only on η) $\|\sigma^{[n_1]}(z_2(t)) - x_0\| < \gamma$ for $t \in [0, 1/2]$. Moreover, on $[0.16, 0.34]$, $s(t) \in [3/4, 1]$ and therefore ϕ_1 is greater or equal to $9/10$ (due to the 10 that appears in the expressions of ϕ_1 and ϕ_2). Thus, the behavior of z_1 is given by (22) and $\left\| z_1(\frac{1}{2}) - \psi(x_0) \right\|_\infty < 2\gamma + \delta/2 \leq \varepsilon$.

For the interval $[1/2, 1]$ the roles of z_1 and z_2 are reversed. Following the reasoning done for z_2 on $[0, 1/2]$, one concludes that

$$\|z_1(t) - \psi(x_0)\|_\infty < (\gamma + \delta)/2 + \varepsilon < \tfrac{1}{2} \qquad \text{for } t \in [1/2, 1],$$

and that $\|z_2(1) - f_M(x_0)\|_\infty < 2\gamma + \delta/2 \leq \varepsilon$. We can repeat this process for z_1 and z_2 on subsequent intervals, which shows that for $j \in \mathbb{N}$, if $t \in [j, j + \frac{1}{2}]$ then $\|z_2(t) - \psi^{[j]}(x_0)\|_\infty \leq \varepsilon$.

Finally, an application of Theorem 4 to the system (23) implies that z_1 and z_2 are solutions of a polynomial IVP, as claimed.

Notice that if we apply the error-contracting function σ to z_1 we can make the error arbitrarily small. Therefore, Theorem 3 implies Theorem 2.

8 Final remarks

We have shown that robust analytic maps and flows can simulate Turing machines, and that those flows can be defined by polynomial ODEs, 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 easily shown that the function z in Theorem 3 is computable by Shannon's General Purpose Analog Computer (GPAC) [Sha41], [PE74], [GC03]. Hence, it follows that GPACs can simulate Turing machines and, according to [GC03], z also belongs to the class of (analytic) real recursive functions $[0, 1, -1, K, U; COMP, I]$ (see [Cam02a]), where K is an appropriate set of constants.

Incidentally, the present work also provides undecidability results for polynomial IVPs. For instance, given a polynomial IVP and some open set on phase space, the question of knowing whether its solution passes through this open set is undecidable. To see this, just consider an IVP simulating an universal Turing machine, and an open set coding the halting state.

It has been recently shown [GZB] that fundamental quantities like the maximal interval of existence for computable IVPs defined with analytic ODEs are, in general, recursively enumerable but non-computable. However, that question is still open for polynomial IVPs.

Nonetheless, it is known that solutions of polynomial IVPs with computable initial conditions and coefficients are computable on their domain of existence [GZB]. That result, combined with the lower bounds we prove in this paper, suggest that there may be a close relation, at least from a computational point of view, between polynomial IVPs and Turing machines.

Acknowledgments. DG and MC wish to thank Carlos Lourenço and Olivier Bournez for helpful discussions. Our interest in the questions addressed in this paper was raised by past discussions with Félix Costa and Cris Moore. DG and MC were partially supported by *Fundação para a Ciência e a Tecnologia* and EU FEDER POCTI/POCI via CLC, project ConTComp POCTI/MAT/45978 /2002, grant SFRH/BD/17436/2004 (DG), and within the initiative RealNComp of SQIG - IT. Additional support to DG was also provided by the *Fundação Calouste Gulbenkian* through the *Programa Gulbenkian de Estímulo à Investigação.* JB acknowledges partial support by CAMGSD through FCT and POCI/FEDER.

References

[Atk89] K. E. Atkinson. *An Introduction to Numerical Analysis.* John Wiley & Sons, 2nd edition, 1989.

[Bra95] M. S. Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoret. Comput. Sci.*, 138(1):67–100, 1995.

[Cam02a] M. L. Campagnolo. The complexity of real recursive functions. In C. S. Calude, M. J. Dinneen, and F. Peper, editors, *Unconventional Models of Computation (UMC'02)*, LNCS 2509, pages 1–14. Springer, 2002.

[Cam02b] M. L. Campagnolo. *Computational Complexity of Real Valued Recursive Functions and Analog Circuits*. PhD thesis, Instituto Superior Técnico/Universidade Técnica de Lisboa, 2002.

[Cas96] M. Casey. The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction. *Neural Comp.*, 8:1135–1178, 1996.

[Cas98] M. Casey. Correction to proof that recurrent neural networks can robustly recognize only regular languages. *Neural Comp.*, 10:1067–1069, 1998.

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

[CMC00] M. L. Campagnolo, C. Moore, and J. F. Costa. Iteration, inequalities, and differentiability in analog computers. *J. Complexity*, 16(4):642–660, 2000.

[CMC02] M. L. Campagnolo, C. Moore, and J. F. Costa. An analog characterization of the Grzegorczyk hierarchy. *J. Complexity*, 18(4):977–1000, 2002.

[GC03] D. S. Graça and J. F. Costa. Analog computers and recursive functions over the reals. *J. Complexity*, 19(5):644–664, 2003.

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

[GH83] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcation of Vector Fields*. Springer, 1983.

[GZB] D.S. Graça, N. Zhong, and J. Buescu. Computability, noncomputability and undecidability of maximal intervals of IVPs. *to appear*.

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

[HW95] J. H. Hubbard and B. H. West. *Differential Equations: A Dynamical Systems Approach - Higher-Dimensional Systems*. Springer, 1995.

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

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

[Koi96] P. Koiran. A family of universal recurrent networks. *Theoret. Comput. Sci.*, 168(2):473–480, 1996.

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

[MC04] J. Mycka and J. F. Costa. Real recursive functions and their hierarchy. *J. Complexity*, 20(6):835–857, 2004.

[MO98] W. Maass and P. Orponen. On the effect of analog noise in discrete-time analog computations. *Neural Comput.*, 10(5):1071–1095, 1998.

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

[Moo98] 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.

[PE74] M. B. Pour-El. Abstract computability and its relations to the general purpose analog computer. *Trans. Amer. Math. Soc.*, 199:1–28, 1974.

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

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

[Rit48] J. F. Ritt. *Integration in Finite Terms.* Columbia Univ. Press, 1948.

[Ros72] M. Rosenlicht. Integration in finite terms. *Amer. Math. Monthly*, 79(9):963–972, 1972.

[RS85] L. A. Rubel and F. Singer. A differentially algebraic elimination theorem with application to analog computability in the calculus of variations. *Proc. Amer. Math. Soc.*, 94(4):653–658, 1985.

[Sha41] C. E. Shannon. Mathematical theory of the differential analyzer. *J. Math. Phys. MIT*, 20:337–354, 1941.

[Sie99] H. T. Siegelmann. *Neural Networks and Analog Computation: Beyond the Turing Limit.* Birkhäuser, 1999.

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

[Via01] M. Viana. Dynamical systems: Moving into the next century. In B. Engquist and W. Schmid, editors, *Mathematics Unlimited - 2001 and Beyond*, pages 1167–1178. Springer, 2001.

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