Decidability and Undecidability in Dynamical Systems

Emmanuel Hainry

Decidability and Undecidability in Dynamical Systems

E H

Nancy Université, LORIA, Université Henri Poincaré, Nancy, France

Emmanuel.Hainry@loria.fr

A computing system can be modelized in various ways: one being in analogy with transfer functions, this is a function that associates to an input and optionally some internal states, an output ; another being focused on the behaviour of the system, that is describing the sequence of states the system will follow to get from this input to produce the output. This second kind of system can be defined by dynamical systems. They indeed describe the "local" behaviour of a system by associating a configuration of the system to the next configuration. It is obviously interesting to get an idea of the "global" behaviour of such a dynamical system. The questions that it raises can be for example related to the reachability of a certain configuration or set of configurations or to the computation of the points that will be visited infinitely often. Those questions are unfortunately very complex: they are in most cases undecidable. This article will describe the fundamental problems on dynamical systems and exhibit some results on decidability and undecidability in various kinds of dynamical systems.

Keywords: Dynamical Systems, Reachability, ω-limit set, Skolem-Pisot problem, General Purpose Analog Computer.

1 Introduction

Dynamical systems are a very generic model that can describe a dynamical process. Dynamical systems consist of just two things: a space where the system will evolve and the dynamics map that describes this evolution. It is then very easy to follow the evolution of a system as at each time its behaviour is defined but on the long term, there is no direct way to foresee where the system will end up.

The study of dynamical systems is made particularly interesting by its versatility: phenomena from many different domains can be described using them. Examples come from mathematics [20], physics, biology [27] or of course models of computation; the famous Lorenz' attractor [24] is an example of a dynamical system describing a meteorological phenomenon; simulating Turing machines with dynamical systems was for example done in [26, 22]. Unfortunately, if the fact that it is possible to simulate Turing machines in dynamical systems makes them a powerful model, it also means that we will encounter undecidability when studying them.

The canonical problem that is defined for dynamical systems are reachability problems (reachability of a point or reachability of a region of space). Some other questions are also of importance, for example knowing the set of points that get ultimately reached (the ω-limit set) can be important in some systems. Reachability is in general undecidable, but in some specific systems it becomes decidable. Linear dynamical systems are a candidate for a decidable world as point-to-point reachability can be proved decidable in them [21], however, it is still open whether the Skolem-Pisot problem (which consists in reachability of a hyperplane for a linear dynamical system) is decidable. Some results are known but they don't yet enlighten the whole decision problem. As an example of recent developments, [18] shows that in small dimensions (up to 5), the problem is decidable, and [6] shows that this problem is NP-hard which gives a lower bound on its complexity. As this problem also arises in a continuous context it would be interesting to study the continuous Skolem-Pisot problem for continuous-time linear dynamical systems. Considering a continuous space may make the study of this problem easier than in a discrete space, indeed if two points on the two different sides of the aimed hyperplane are reached, continuity (and the intermediate values theorem) implies that the hyperplane will also be reached.

The (point to point) reachability problem, which is undecidable in the general case, has been shown undecidable for various very restricted classes of dynamical systems, such as Piecewise Constant Derivative systems [1] where the dynamics are really simple as it consists of a sharing of the space into regions where the derivative will be constant. Other results on the subject of reachability and undecidability of problems in hybrid systems are studied in [1, 2, 3, 7].

Section 2 defines dynamical systems and introduces various examples of systems from different domains that can be described using dynamical systems. It also presents the canonical problems for dynamical systems. Section 3 presents basic undecidability results on dynamical systems. It will recall that all basic problems are undecidable in general and even undecidable for some quite restricted classes of dynamical systems. Section 4 however gives some decidability results. Clearly, those decidability results concern constrained dynamical systems, but those systems are not devoid of interest as they can be used to represent some natural problems. Section 5 will detail some open problems and some recent results in characterizing which part of them are problematic. This helps to draw the line between decidability and undecidability in dynamical systems.

2 Definitions and examples

There exist two types of dynamical systems: those that work in discrete time and those that work in continuous time. The first allow to modelize discrete phenomena such as computing using Turing machines or similar models, recurrence schemes, for example linear recurrent sequences or synchronous models. The second allows to continuous phenomena, and hence is used to simulate physical phenomena such as celestial bodies trajectories under gravitation laws or meteorological phenomena but also computing devices that work in non-discrete time. We will here define those two kinds of dynamical systems and present a few examples that should show the variety of applications that this model has.

2.1 Discrete-time dynamical systems

Definition 1 (Discrete-time dynamical systems) *A discrete time dynamical system is a tuple (X, f) where X is the configuration space (usually $\mathbb{R}^n$ or $\mathbb{Z}^n$) and $f : X \to X$ is a map that defines the dynamics of the system through the equation $y(n+1) = f(y(n))$*

Definition 2 (Trajectory) *Given an initial point $x_0 \in X$, there exist a unique trajectory described by*

$$\{y(n); y(0) = x_0 \text{ and } \forall n \in \mathbb{N}, y(n+1) = f(y(n))\}$$

We will now give three examples of discrete-time dynamical systems that hint at their interest in mathematics and computer science, and also give intuition on the complexity that can be hidden in very simple descriptions. This is however merely a scratch at the surface of the diversity of this model.

2.1.1 Linear recursive sequence

A linear recursive sequence is an sequence (a_n) that satisfies an equation of the form

$$\forall n \in \mathbb{N}, a_{n+d} = c_1 a_{n+d-1} + c_2 a_{n+d-2} + \cdots + c_d a_n$$

where the c_i are constants. The sequence usually belongs to $\mathbb{Z}^{\mathbb{N}}$ or $\mathbb{R}^{\mathbb{N}}$. Giving the initial vector $\{a_0, a_1, \ldots, a_{d-1}\}$ determines a unique sequence.

A classical example of such a linear recursive sequence is Fibonacci's sequence that satisfies $a_{n+2} = a_{n+1} + a_n$ with $a_0 = a_1 = 1$.

A linear recursive sequence of integers can be expressed as discrete-time dynamical systems with $X = \mathbb{Z}^d$ and $f : Y \mapsto \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ c_d & c_{d-1} & \cdots & c_2 & c_1 \end{pmatrix} Y$. The initial condition is then written as a vertical matrix $Y_0 = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{d-1} \end{pmatrix}$. It is easy to verify that $\forall n \in \mathbb{N}, \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} Y_n = a_n$. Note that the matrix that appears in function f is well known: it is a transpose companion matrix.

2.1.2 Turing machine

It is possible to simulate the behaviour of Turing machines using discrete dynamical systems. The space on which we will work will be the set of configurations of the machine (state, tapes and head) and the dynamics map will describe the transitions of the machine.

Let us consider a Turing machine with one tape, on alphabet $\Sigma = \{0, 1, ..., 9\}$, with states in $\mathbb{N}$, and with transition function $\delta : \mathbb{N} \times \Sigma \to \mathbb{N} \times \Sigma \times \{\triangleleft, \bullet, \triangleright\}$ where $\triangleleft$ denotes a move of the head to the left; $\bullet$ the fact that the head does not move; $\triangleright$ a move to the right.

In the configuration space, we want to encode the whole bi-infinite tape and the position of the head. We will for that cut the tape at the head and consider the left part of the tape as a number and the right part as another number:

$$X = \mathbb{N} \times \mathbb{N} \times \mathbb{N}$$

with the first integer representing the state; the second, the left part of the tape beginning at the head; the third, the right part of the tape, from right to left, ending just right of the head.

For example, if the tape is $\boxed{}\boxed{1}\boxed{3}\boxed{2}\boxed{5\star}\boxed{0}\boxed{1}\boxed{}$ with the star denoting the position of the head, the configuration will be $(s, 1325, 10)$.

The dynamics map f can be defined in the following way:

$$\begin{aligned} f : (n, a, b) \quad \mapsto \quad & (n', a', b') \text{ where} \\ & \text{Let } \sigma = a \bmod 10 \text{ ; consider } \delta(n, \sigma) = (n', \sigma', \tau) \\ & \begin{cases} \text{if } \tau = \bullet & a' = a - \sigma + \sigma' \text{ ; } b' = b \\ \text{if } \tau = \triangleright & a' = a/10 \text{ ; } b' = 10b + \sigma' \\ \text{if } \tau = \triangleleft & b' = b/10 \text{ ; } a' = 10(a - \sigma + \sigma') + (b \bmod 10) \end{cases} \end{aligned}$$

2.1.3 Syracuse (or Collatz) sequence

The Syracuse sequence (also called Collatz' sequence) is a famous example of a simply defined sequence whose general behaviour is not totally understood.

Its domain is $X = \mathbb{N}$, its dynamics is defined by the following function:

$$\begin{cases} S(n+1) = \frac{S(n)}{2} & \text{if } S(n) \text{ is even} \\ S(n+1) = 3S(n) + 1 & \text{if } S(n) \text{ is odd.} \end{cases}$$

For any tested natural initial, the sequence reaches 1. It is however not proved that all such defined sequences reaches 1 (and enters the cycle $(1, 4, 2)$).

2.1.4 Hénon map

The Hénon map is a discrete-time dynamical system. It is one of the most studied examples of dynamical systems that exhibit chaotic behavior. The Hénon map takes a point (x, y) in the plane and maps it to a new point through the following dynamics:

$$\begin{cases} x_{n+1} & = y_n + 1 - ax_n^2 \\ y_{n+1} & = bx_n. \end{cases}$$

For the canonical Hénon map, the values chosen are $a = 1.4$ and $b = 0.3$.

The Hénon map exhibits a strange attractor behaviour similar to Lorenz's one (see 2.2.3) in a simpler way.

2.2 Continuous-time dynamical system

Definition 3 (Continuous-time dynamical system) *A continuous-time dynamical system is a tuple (X, f) where X is the configuration state (usually $\mathbb{R}^n$) and $f : X \to X$ defines the dynamics through the differential equation*

$$y'(t) = f(y(t)).$$

Definition 4 (Trajectory) *A trajectory from an initial point y_0 of a continuous-time dynamical system is the set $\{y(t); t \in \mathbb{R}^+\}$ where $y(0) = y_0$ and such that y satisfies the differential equation.*

This kind of dynamical systems allows to modelize many kinds of phenomena. Especially physical phenomena that are essentially continuous and biological phenomena, but also continuous computing processes. We will here give some examples related to continuous computation, astronomy and meteorological phenomena.

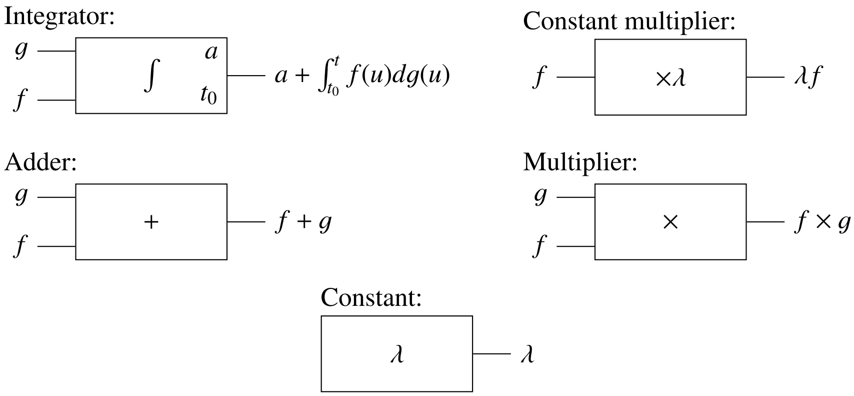

Figure 1: The different types of units used in a G

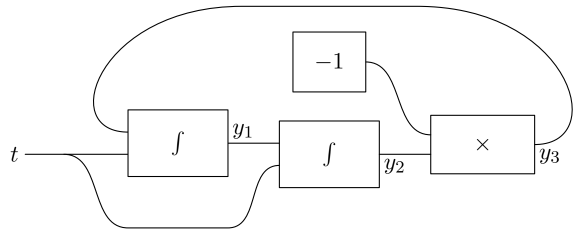

Figure 2: Generating cos and sin with a G . We have $y_1 = \cos$, $y_2 = \sin$, $y_3 = -\sin$

2.2.1 General Purpose Analog Computer (G)

The G , defined by Shannon [28], may be seen as a circuit that connects blackboxes able to realize functions such as addition, multiplication or integration. As with electronics circuits, it is allowed to create loops in a G , which permits to generate a circuit that has to solve complex differential equations, but some interconnections are forbidden as they yield problematic functions[14].

The different basic blocks that can be used in a G are described in Figure 1. The functions generated by G s include polynomials, exponential, the usual trigonometrical functions, their inverses. For example, the G drawn in Figure 2 generates functions cos and sin.

Since Shannon, work has been done to precisely characterize what can be generated by G s. The following characterization was shown in [14]:

Proposition 5 *A scalar function $f : \mathbb{R} \to \mathbb{R}$ can be generated by a* G *iff it is a* *component of a solution of the system*

$$y' = p(y, t), \tag{1}$$

where p is a vector of polynomials. A function $f : \mathbb{R} \to \mathbb{R}^k$ is generated by a G *iff all of its components are.*

The G is hence a polynomial dynamical system.

2.2.2 n-body problem

Newton's laws of gravitation describe how celestial bodies are attracted by each other. When considering two isolated celestial bodies, their trajectories can be described through a differential equation that is obtained by stating that the system is isolated and hence the gravitation forces are the only ones involved in the motion.

Applying the basic laws of mechanics, we would like to set the space as

$$X = \left(\mathbb{R}^3\right)^n$$

and the dynamics as

$$x_i'' = -G \sum_{j \neq i} \frac{m_i m_j}{\left((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2\right)^{\frac{3}{2}}} (x_i - x_j)$$

with the same for y and z. Unfortunately, this is not a true dynamical system as it relies on second order derivatives. To circumvent this problem, we will double the size of the space, considering the speeds to be part of the configuration (which makes sense as the speeds will also be part of the initial conditions). We will consider

$$X = \left(\mathbb{R}^3\right)^{2n}$$

and assume the positions are the

$$x, y, z$$

with odd indexes and the corresponding speeds, the even indexes:

$$\begin{array}{rcl} x_{2i}' & = & -G \sum_{j \neq i} \frac{m_i m_j}{\left((x_{2i-1} - x_{2j-1})^2 + (y_{2i-1} - y_{2j-1})^2 + (z_{2i-1} - z_{2j-1})^2\right)^{\frac{3}{2}}} (x_{2i-1} - x_{2j-1}) \\ x_{2i-1}' & = & x_{2i} \end{array}$$

We have then a dynamical system that only uses rational fractions. It is hence quite a simple dynamical system. We know because of Kepler that if there are two bodies, the trajectory of one relatively to the other will be a conic curve: either an ellipse, either a parabola, either a hyperbola. It is moreover simple to decide which it is and hence to know if the two bodies will stay close or will diverge. However, we also know since Poincaré that as soon as there are 3 celestial bodies, there is no simpler general equation and for example we cannot decide in general if the system will indefinitely grow or not.

Although simple, this system yields undecidable questions.

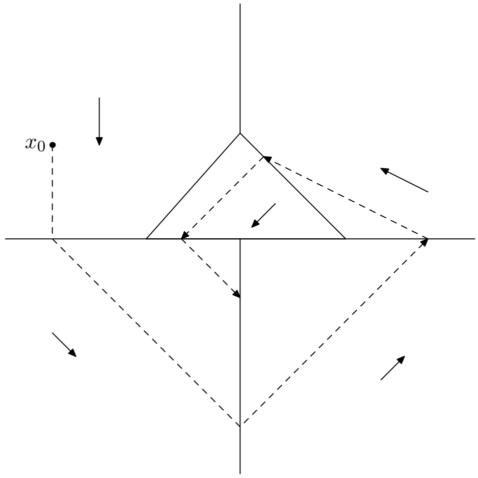

Figure 3: A PCD system

2.2.3 Lorenz attractor

Convection flows, air motions can be modelised using differential equations. Edward Lorenz indeed used differential equations to depict meteorological phenomena [24]. The following is the canonical equation that Lorenz introduced to describe convection in the atmosphere.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}' = \begin{pmatrix} 10(y-x) \\ 28x - y - xz \\ xy - \frac{8}{3}z \end{pmatrix}$$

A trajectory of this system exhibits a chaotic behaviour: the Lorenz attractor. Its shape is a bit like two discs on which the trajectory goes into circles and sometimes decides to change to the other disc.

This system is a polynomial (of degree 2) dynamical system, but although it is a simple system, the long term behaviour is difficult to grasp.

2.2.4 Piecewise Constant Derivative (PCD)

Piecewise constant derivative systems are dynamical systems where the space is $\mathbb{R}^d$ and the transition map is piecewise constant. The pieces being defined by parallelotopes.

For example, figure 3 shows a PCD in $\mathbb{R}^2$, the speed vectors (the derivatives) in each region and a trajectory originating from a point x_0.

PCD are a continuous-time dynamical system, but the interesting parts of its behaviour are the single points when the trajectory reaches a frontier between two regions and changes direction. There is hence a discrete time behaviour associated with a continuous-time trajectory of the system.

We will see in section 3.2 that this system which is really weak is in fact able to simulate a Turing machine and hence yields undecidable problems.

2.2.5 Time-differentiable Petri nets

Time-differentiable Petri nets (TDPN) are a generalization of Petri nets with a speed control for transitions. For a precise description, refer to [15]. TDPN are in fact continuous-time dynamical systems whose dynamics map is simply a linear expression with minimum operator. It is hence more powerful than a simple linear dynamical system, but only slightly more powerful.

We will recall in section 3.3 that it is possible to simulate 2 counter machines with TDPN. This means that this system also presents undecidable problems even though it is just linear differential equations with the adjunction of a minimum operation.

2.3 Problems

The questions that are asked to those systems depend on the application considered. For our examples, we would like to decide if the Turing machine simulated by a specific dynamical system halts, we would like to know if a linear recursive sequence reaches 0, if the celestial bodies involved in a n-body problem will collide...

Those various questions usually correspond to a few simple problems on dynamical systems: reachability (of a point or of a region) and ω-limit (some kind of ultimate reachability).

Definition 6 (Reachability) *Given $A \in \mathbb{R}^{n\times n}$, $X_0 \in \mathbb{R}^n$, $Y \in \mathbb{R}^n$, the system is said to reach Y from X_0 if there exists $t \in \mathbb{R}$ such that $X(t) = Y$ with X the trajectory defined with the dynamics A and the initial point X_0.*

Definition 7 (ω-limit points) *Given a trajectory X, a point Y is an ω-limit point of X if there is a diverging increasing sequence $(t_n) \in \mathbb{R}^{\mathbb{N}}$ such that $Y = \lim_{n\to+\infty} X(t_n)$.*

Definition 8 (ω-limit sets) *The ω-limit set of a dynamical system is the set of its ω-limit points: $\omega(X) = \cap_n \overline{\cup_{t>n} X(t)}$, where $\overline{A}$ is the closure of the set A.*

Problem 9 (Reachability problem) *Given a trajectory X defined from $A \in \mathbb{K}^{n\times n}$ and $X_0 \in \mathbb{K}^n$, a point $Y \in \mathbb{K}^n$, decide whether Y can be reached from X_0.*

Problem 10 (ω-limit set) *Given a dynamical system, compute a representation of its ω-limit set.*

The classical Skolem-Pisot problem originally consists in determining if a linear recurrent sequence has a zero. It can however be defined as a hyperplane reachability problem.

Problem 11 (Skolem-Pisot problem) *Given a trajectory X, given $C \in \mathbb{K}^n$ defining an hyperplane*[1] *of $\mathbb{K}^n$, decide if $\exists t \in \mathbb{R}$ such that $C^T X(t) = 0$? In other words, does the trajectory X intersect the hyperplane defined by C?*

The problems we will consider will be those for which the field $\mathbb{K}$ is in fact the set of rational numbers $\mathbb{Q}$.

2.4 Prerequisites

We will now present some mathematical notions and the way we will represent them in order to study computability and decidability of the problems that we defined.

2.4.1 Linear continuous-time dynamical systems

The dynamics of a linear dynamical system are described by a linear differential equation. To describe such a system, we take a matrix of real numbers which will represent the dynamics and a vector of reals that is the initial point. We use here classical definitions and notations that can be found in [19].

Definition 12 (Linear continuous-time dynamical system) *Given a matrix $A \in \mathbb{R}^{n\times n}$ and a vector $X_0 \in \mathbb{R}^n$. We define X as the solution of the following Cauchy problem:*

$$\begin{cases} X' & = AX \\ X(0) & = X_0. \end{cases}$$

X is called a trajectory of the system.

2.4.2 Polynomials

Let us now recall a few notations, mathematical tools and algorithms on polynomials. In the following, we use a field $\mathbb{K}$ that is a subfield of $\mathbb{C}$.

Definition 13 (Ring of polynomials) *We denote $\mathbb{K}[X]$ the ring of one variable polynomials with coefficients in $\mathbb{K}$. A polynomial can be written as $P(X) = \sum_{i=1}^{n} a_i X^i$, with $a_i \in \mathbb{K}$ and $a_n \neq 0$. The integer n is the degree of P.*

Definition 14 (Roots of a polynomial) *The set $Z(P)$ of roots of a polynomial P is defined as $Z(P) = \{x \in \mathbb{C}; P(x) = 0\}$*

[1]The hyperplane defined by C is the set of points Y such that $C^T Y = \begin{bmatrix} 0 \end{bmatrix}$

Definition 15 (Algebraic numbers) *The set of roots of polynomials with coefficients in* $\mathbb{Q}$ *is the set of algebraic numbers.*

An algebraic number can be represented uniquely by the minimal polynomial it nulls (minimal in $\mathbb{Q}[X]$ for the division) and a ball containing only one root of the polynomial. Note that the size of the ball can be chosen using only the values of the coefficients of the polynomial as [25] shows a bound on the distance between roots of a polynomial from its coefficient.

Definition 16 (Representation of an algebraic number) *An algebraic number* α *will be represented by* $(P, (a, b), \rho)$ *where* P *is the minimal polynomial of* α*,* $a + \mathrm{i}b$ *is an approximation of* α *such that* $|\alpha - (a + \mathrm{i}b)| < \rho$ *and* α *is the only root of* P *in the open ball* $\mathcal{B}(a + \mathrm{i}b, \rho)$.

It can be shown that given the representations of two algebraic numbers α and β, the representations of $\alpha + \beta$, $\alpha - \beta$, $\alpha\beta$ and α/β can be computed. See [8, 9] for details.

We will also need specific results on algebraic numbers that come from [4, 11].

Proposition 17 (Baker) *Given* $\alpha \in \mathbb{C} - \{0\}$*,* α *and* e^{α} *are not both algebraic numbers.*

Theorem 18 (Gelfond-Schneider) *Let* α *and* β *be two algebraic numbers. If* $\alpha \notin \{0, 1\}$ *and* $\beta \notin \mathbb{Q}$*, then* α^{β} *is not algebraic*

2.4.3 Matrices

Definition 19 (Characteristic polynomial) *Given a matrix* $A \in \mathbb{K}^{n \times n}$*, its characteristic polynomial is* $\chi_A(X) = \det(XI_n - A)$

Definition 20 (Exponential of a matrix) *Given a matrix* A*, its exponential denoted* $\exp(A)$ *is the matrix*

$$\sum_{i=1}^{+\infty} \frac{1}{i!} A^i.$$

Note that the exponential is well defined for all real matrices.

All matrices can be put in Jordan form, which allows to compute easily the exponential. To find more about Jordan matrices and blocks, the reader may consult [19] or [23].

Definition 21 (Jordan block) *A Jordan block is a square matrix of one of the two following forms*

$$\begin{bmatrix} \lambda & & & \\ 1 & \lambda & & \\ & \ddots & \ddots & \\ & & 1 & \lambda \end{bmatrix}; \qquad \begin{bmatrix} B & & & \\ I_2 & B & & \\ & \ddots & \ddots & \\ & & I_2 & B \end{bmatrix} \text{ with } B = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \text{ and } I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Definition 22 (Jordan form) *A matrix that contains Jordan blocks on its diagonal is said to be in Jordan form.*

$$\begin{bmatrix} D_1 & 0 & \cdots & 0 \\ 0 & D_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & D_n \end{bmatrix}$$

Proposition 23 ([23]) *Any matrix $A \in \mathbb{R}^{n\times n}$ is similar to a matrix in Jordan form. In other words,*

$$\exists P \in GL(\mathbb{R}^{n\times n}) \text{ and } J \text{ in Jordan form such that } A = P^{-1}JP.$$

3 Undecidability results

3.1 Undecidability for polynomial dynamical systems

Many biological phenomena can be modelized using polynomial dynamical systems rather than linear dynamical systems. A famous example comes from meteorological systems which were described by Lorenz in [24]. Lorenz' attractor has a quite chaotic behaviour which gives the intuition that the reachability problem in polynomial dynamical systems is not decidable. Other polynomial differential systems yields fractal basins of attraction. In other words, this dynamical systems has exactly two ω-limit points depending on the initial point and, the set of starting points that will lead to the first of those attractors is a fractal, for example a Julia set.

In those systems, from already known results, we can infer that the Skolem-Pisot problem and the reachability problem are undecidable.

Theorem 24 *The Skolem-Pisot problem is undecidable for polynomial dynamical systems.*

*P*roof: From [13], we know that it is possible to simulate a Turing machine using a polynomial differential system. The halt of the Turing machine is then equivalent to the system reaching the hyperplane $z = q_f$ which stands for the halting state. This is an instance of the Skolem-Pisot problem. □

Theorem 25 *Reachability is undecidable for polynomial dynamical systems.*

*P*roof: Let us modify the Turing machine of the previous proof so that from the halting state, the machine erases its tape then enters a special state. Simulating this machine by the same mechanism from [13], the dynamical system reaches the point representing blank tapes and special state if and only if the original machine halts. This means we can translate any instance of the halting problem into a reachability in polynomial differential systems problem. □

3.1.1 Degree two polynomials

Theorem 26 *Reachability is undecidable for degree two polynomial dynamical systems.*

To prove this, we will show that in fact, the simulation of a Turing machine described previously can be written in a system of degree two, as any polynomial dynamical system can be simulated by a degree two polynomial dynamical system.

Lemma 27 *Any polynomial dynamical system can be rewritten in a degree two polynomial dynamical system*

This lemma is a direct consequence of the following technical lemma:

Lemma 28 *Any polynomial dynamical system of degree $k > 2$ with l components can be rewritten in a polynomial dynamical system of degree $k - 1$ with $l + l^2$ components.*

*P*roof: Consider a polynomial dynamical system of degree $k > 2$ with l components:

$$\begin{cases} x_1' &= p_1(x_1, x_2, \cdots, x_l) \\ & \vdots \\ x_l' &= p_l(x_1, x_2, \cdots, x_l) \end{cases}$$

To reduce the degree, we will replace some quadratic factors by new components. We will in fact create l^2 new components figuring the various $x_i x_j$. For any $i, j \in \{1, ..., l\}$, let $y_{i,j}$ be a new component whose derivative will be chosen as $x_i' x_j + x_i x_j'$. We can write

$$y_{i,j} = p_i(x_1, x_2, \cdots, x_l)x_j + x_i p_j(x_1, x_2, \cdots, x_l).$$

Note that those equations are polynomials of degree at most $k + 1$. In each equation of degree k or $k + 1$, let us replace one quadratic factor of the highest degree monomial by the corresponding $y_{i,j}$. Now all equations describing the x_i' are of degree at most $k - 1$, some equations describing the y_i' may still be of degree k, as $k \geq 3$, there are still quadratic factors consisting only of x_i in those equations. Those factors can then be replaced by a y_i and the whole system is now of degree $k - 1$. □

Note that this construction gives in fact slightly better results than what we claim in the lemma: it is possible to choose which quadratic factors to replace and hence not generate all the $y_{i,j}$; also, if we replace all $x_i x_j$ in all the equations, we get a system which is of degree at most $\left\lceil \frac{k+1}{2} \right\rceil$.

3.2 Undecidability for PCD

PCD seem to be a very weak system in which it is easy to understand the overall behaviour of the system. In fact, in low dimension, it is indeed not as powerful as Turing machines, but with high dimensions, it is possible to simulate 2-stack push-down automata and hence Turing machines. It has indeed been shown [1] that for PCD of dimension 3 the reachability problem is undecidable, but that for PCD of dimension 2, it becomes decidable.

Theorem 29 *[1, Theorem 6] For PCD of dimension 2, reachability is decidable.*

Theorem 30 *[1, Theorem 18] Every 2-stack push-down automata (and hence every Turing machine) can be simulated by a 3-dimensional PCD.*

Corollary 31 *[1, Corollary 19] For PCD of dimension 3, reachability is undecidable.*

3.3 Undecidability for lin-min systems

Haddad, Recalde and Silva present in [15] two simulations of two counter machines by TDPN. It means that Turing machines can be simulated by TDPN and hence that deciding the condition equivalent to the halt of Turing machines is impossible for Turing machines.

Theorem 32 *[15, Theorem 1] Given a two counter machine $\mathcal{M}$, one can build a TDPN $\mathcal{D}$, with a constant number of places, whose size is linear w.r.t. the machine, whose associated ODE has dimension 6 and such that $\mathcal{D}$ robustly simulates $\mathcal{M}$.*

As a corollary, they obtain the following undecidability result:

Proposition 33 *[15, Proposition 1] Let $\mathcal{D}$ be a TDPN whose associated ODE has dimension at least 6, m_0, m_1 be markings, p be a place and $k \in \mathbb{N}$ then:*

- *the problem whether there is a τ such that the trajectory starting at m_0 fulfills $m(\tau)(p) = k$ is undecidable.*
- *the problem whether there is a τ such that the trajectory starting at m_0 fulfills $m(\tau)(p) \geq k$ is undecidable.*
- *the problem whether there is a τ such that the trajectory starting at m_0 fulfills $m(\tau)(p) \geq m_1$ is undecidable.*

Those undecidable problems are in fact reachability of a region problems, like Skolem-Pisot's problem is reachability of a hyperplane for linear dynamical systems.

Proposition 34 *[15, Proposition 2] Let $\mathcal{D}$ be a TDPN whose associated ODE has dimension at least 8, m_0 be a marking. Then the problem whether the trajectory m starting at m_0 is such that $\lim_{r \to +\infty} m(\tau)$ exists is undecidable.*

This problem (the steady-state problem) is related to the ω-limit set, as it answers questions about ultimate reachability in the system.

4 Decidability results

4.1 Decidability for Discrete-time linear dynamical systems

Discrete-time linear dynamical systems are, as their name suggest DTDS in which the dynamics map is a linear function. Linear recursive sequence are an example of Discrete-time linear dynamical systems.

In this system, the reachability problem is decidable. It has indeed been shown in [21] that it can be computed in polynomial time whether a given point will or will not be reached by such a linear dynamical system.

4.2 Decidability for linear dynamical systems

This section is devoted to proving the two main theorems of this article: theorem 35 and theorem 36 which appeared in [16, 17]. They respectively show that the ω-limit set of linear dynamical systems is computable; and that the reachability problem for linear dynamical systems is decidable. We hence have a computability result and a decidability result similar to the result on discrete-time linear dynamical systems.

Theorem 35 *Given a linear dynamical system, its ω-limit set is computable and is a semi-algebraic set.*

*P*roof: In this proof, we will work with a matrix in Jordan form as it is always possible to put the matrix in this form as shown below. Note that the Jordan matrix will have algebraic coefficients and not only rational ones.

Building the Jordan form of a matrix implies knowing its eigenvalues, for that we need to compute the roots of the characteristic polynomial of the matrix.

This consists in the following steps that are classical:

- computing the characteristic polynomial;
- factorizing the polynomial in $\mathbb{Q}[X]$ (section A.1);
- computing the roots (section A.2);
- jordanizing the matrix (section A.3).

We now consider the following system (which is only a base change rewriting of the original system): $A = \begin{bmatrix} D_1 & 0 & \cdots & 0 \\ 0 & D_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & D_k \end{bmatrix}$ with the D_i being Jordan blocks.

The solution of the Cauchy system $\begin{cases} X' = AX \\ X(0) = X_0 \end{cases}$ is formally $X(t) = \exp(tA)X_0$.

We then need to compute the exponential of tA. It is easy to check that

$$\exp(tA) = \begin{bmatrix} \exp(tD_1) & & & \\ & \exp(tD_2) & & \\ & & \ddots & \\ & & & \exp(tD_k) \end{bmatrix}$$

Let us now compute the ω-limit set Ω for the different possible cases.

- If one eigenvalue has a positive real part, then

  $$\Omega = \emptyset.$$

  Indeed, this component diverges towards $+\infty$ hence no real point will be a limit of a sub-trajectory.

- If one eigenvalue has a null real part and a multiplicity greater than 1,

  $$\Omega = \emptyset.$$

  Indeed, the second component related to this eigenvalue will diverge to $+\infty$ due to the t term in the exponential matrix.

- If all eigenvalues have negative real part, all the components will converge to 0, regardless of the multiplicity of the eigenvalues, hence

  $$\Omega = \{0^k\}.$$

- If all eigenvalues are non positive reals, then all the components corresponding to negative eigenvalues will converge to 0 as in the third case, the components corresponding to a null eigenvalue will either be constant either diverge to $+\infty$ if the multiplicity is greater than 1. Hence, either

  $$\Omega = \{(..., x_{0_i}, 0, ...)\}.$$

  either

  $$\Omega = \emptyset.$$

- Otherwise we have complex eigenvalues of null real part and multiplicity 1, and we may have other eigenvalues, either 0 with multiplicity 1 (whose component will be constant), either eigenvalues with negative real part (that will converge to 0). Only the complex eigenvalues with null real part are of interest, so let us consider only them for now.

  We have eigenvalues $\mathrm{i}b_1$, $-\mathrm{i}b_1$, ..., $\mathrm{i}b_n$, $-\mathrm{i}b_n$, with the b_i being real algebraic numbers. There are two cases to consider: either the family $(b_1, b_2, ..., b_n)$ is $\mathbb{Q}$ linearly independent, either it is not.

– Let us assume the $(b_1, ..., b_n)$ is $\mathbb{Q}$ linearly independent. In this case, the trajectory will not be periodic but instead will be dense in the set of points whose projections on each (x_{2k+1}, x_{2k+2}) are the circles defined by $x_{2k+1}^2 + x_{2k+2}^2 = x_{2k+1}{}_0^2 + x_{2k+1}{}_0^2$. Indeed, it is trivial if $n = 1$. Let us consider it true for $n = k$. It means that for any given point $(\alpha_{1_1}, \alpha_{1_2}, ..., \alpha_{k_1}, \alpha_{k_2}, \alpha_{k+1_1}, \alpha_{k+1_2})$ of that set, there exists a sequence of times $(t_i)_{i\in\mathbb{N}}$ such that

$$\|(x_1(t_i), ..., x_{2k}(t_i)) - (\alpha_{1_1}, ..., \alpha_{k_2})\| < \frac{1}{2^i}.$$

We can similarly, for any α build a sequence of times $(t_j)_{j\in\mathbb{N}}$ such that

$$\|(x_{2k+1}(t_j), x_{2k+2}(t_j)) - (\alpha_{k+1_1}, \alpha_{k+1_2})\| < \frac{1}{2^j}.$$

Indeed, there exists a number t_0 such that

$$(x_{2k+1}(t_0), x_{2k+2}(t_0)) = (\alpha_{k+1_1}, \alpha_{k+1_2}).$$

So choosing $t_j = t_0 + 2j\pi$ verifies this constraint. As x are continuous functions, those inequalities are true for neighbourhoods V_i, V_j of those t_i, t_j. As b_{k+1} is not a linear combination of the $b_1, ..., b_k$, for all i_0, j_0, there exist $i' > i_0$ and $j' > j_0$ such that $V_{i'} \cap V_{j'} \neq \emptyset$. If we take $t^\star_\phi(i_0) \in V_{i'} \cap V_{j'}$, then we have

$$\|(x_1(t^\star), ..., x_{2k+2}(t^\star)) - (\alpha_{1_1}, ..., \alpha_{k+1_2})\| < \frac{1}{2^{i_0+1}}.$$

Hence we have exhibited a sequence that converges towards the said point. Finally,

$$\Omega = \{(x_1, ..., x_n); \forall i, x_{2i+1}^2 + x_{2i+2}^2 = x_{0_2 i+1}^2 + x_{0_2 i+2}^2\}.$$

– Let us assume there exists $\alpha_1, ..., \alpha_n \in \mathbb{Q}^n$ with $\alpha_n \neq 0$ such that

$$\sum \alpha_i b_i = 0.$$

Let Ω_1 be the ω-limit set while considering the $n-1$ first components. Let us first recall that $\begin{bmatrix} \cos(bt) & -\sin(bt) \\ \sin(bt) & \cos(bt) \end{bmatrix}$ is similar to $\begin{bmatrix} e^{ibt} & 0 \\ 0 & e^{-ibt} \end{bmatrix}$. Hence, if we do the variable change, we obtain $X_i(t) = X_{0_i}$, and we have $\prod e^{ib_i t}\alpha_i = 1$ and $e^{-ib_n t}\alpha_n = \prod_{i=1}^{n-1} e^{ib_i t}\alpha_i$ and

$$\left(\prod_{i<n} X_{0_i}^{\alpha_i}\right) X_{2n}(t)^{\alpha_i} = X_{0_2 n}^{\alpha_n} \prod_{i<n} X_i(t)^{\alpha_i}.$$

This polynomial equation is verified by all points of the trajectory and hence constitutes a constraint on the ω-limit set. By an argument similar to the one in the previous item, we can show that the set of points verifying this constraint as well as all the projection constraints is effectively contained in the ω-limit set. Hence, with $X_i = (x_{2i-1} + \mathrm{i}x_{2i})$, we have

$$\Omega = \Omega_1 \cap \{(x_1, ..., x_n); x_{2n-1}^2 + x_{2n}^2 = x_{2n-1_0}^2 + x_{2n_0}^2 \quad \wedge \\ \left(\prod_{i<n} X_{0_i}^{\alpha_i}\right) X_{2n}(t)^{\alpha_i} = X_{0_{2n}}^{\alpha_n} \prod_{i<n} X_i(t)^{\alpha_i}\}$$

In each case, we have been able to give a formal representation of the ω-limit set, either as the empty set, a single point or a combination of polynomial equations. All those descriptions are semi-algebraic which proves the semi-algebraicity of the ω-limit set. $\square$

Theorem 36 *The reachability problem for continuous time linear dynamical systems with rational coefficients is decidable.*

To decide whether a point is reachable we will try to obtain an expression of the trajectory X that is usable and with this expression search for the different t that could be solution. Note that the knowledge of the ω-limit set can help to discriminate certain cases.

Without loss of generality, we will once more suppose that the matrix A is in Jordan form with algebraic coefficients and that the X_0 and Y vectors are also composed of algebraic elements. This means The soution of the system is $X(t) = \exp(tA)X_0$. with

$$\exp(tA) = \begin{bmatrix} \exp(tD_1) & & & \\ & \exp(tD_2) & & \\ & & \ddots & \\ & & & \exp(tD_k) \end{bmatrix}$$

Finding a $t \in \mathbb{R}$ such that $X(t) = Y$ is equivalent to finding such a t for each component i and ensuring this is always the same t. We are going to solve the equation Jordan block by Jordan block. It means we choose an i such that the corresponding part of X_0 is not null (in the other case it is easy to decide if either all $t \in \mathbb{R}$ will be solutions or no t will be solution) and search for a t such that $\exp(tD_i)\begin{bmatrix} x_1 \\ \vdots \\ x_{n_i} \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_{n_i} \end{bmatrix}$ where the x_j and y_j are the elements of X_0 and Y corresponding to the block i. To simplify the notations, we will forget i and just consider the problem as being $\exp(tD)X_0 = Y$ and k being the size of this block.

There are two cases to consider: the two different forms of Jordan blocks. For each of those cases, a few sub cases are to be considered which revolve around the nullity of the real part of the eigenvalue. Let us note that as we deal with algebraic numbers, it is possible to verify if the real part or the imaginary part is null.

4.2.1 First form: a real eigenvalue

The first form of Jordan blocks corresponds to a real eigenvalue λ. Two cases need to be dealt with: $\lambda = 0$ and $\lambda \neq 0$

If $\lambda \neq 0$. The exponential is $\exp(tD) = \mathrm{e}^{t\lambda} \begin{bmatrix} 1 & & & & \\ t & 1 & & & \\ \frac{t^2}{2} & t & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \\ \frac{t^k}{k!} & \cdots & \frac{t^2}{2} & t & 1 \end{bmatrix}$. If X_{0_1} is not 0, then there is at most one $t \in \mathbb{R}$ solution. Indeed, let us consider x_i, the first non null element of $\{x_1, x_k\}$. The only possible t is then $\frac{1}{\lambda} \ln\left(\frac{y_i}{x_i}\right)$.

We want to verify that this t is coherent with the rest of the block. Let us remark that $\mathrm{e}^{t\lambda} = \frac{y_i}{x_i}$ is an algebraic number. If the block has size more than 1, then t verifies some algebraic equations hence the proposition 17 says $\lambda t = 0$, it is easy to verify if $t = 0$ is the solution of the block.

If $\lambda = 0$. The case with $\lambda = 0$ means we are searching for a t such that

$$\begin{bmatrix} 1 & & & & \\ t & 1 & & & \\ t^2/2 & t & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \\ \frac{t^k}{k!} & \cdots & t^2/2 & t & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix}$$

For such a t to exist, we need to have $x_1 = y_1$, $x_2 + tx_1 = y_2$, ... Let us say that x_i is the first non-null element of X. Then the only candidate for t is $\frac{y_{i+1} - x_{i+1}}{x_i}$. Since this candidate is algebraic, it is easy to check whether this t is a solution for the block.

4.2.2 Second form

The second form corresponds to complex eigenvalues. The Jordan block is

$$D = \begin{bmatrix} B & & & \\ I_2 & B & & \\ & \ddots & \ddots & \\ & & I_2 & B \end{bmatrix} \text{ with } B = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \text{ and } I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The exponential is

$$\exp(D) = \mathrm{e}^{ta} \begin{bmatrix} B_2 & & & & \\ tB_2 & B_2 & & & \\ \frac{t^2}{2} B_2^2 & tB_2 & B_2 & & \\ \vdots & \ddots & \ddots & \ddots & \\ \frac{t^k}{k!} B_2^k & \cdots & \frac{t^2}{2} B_2^2 & tB_2 & B_2 \end{bmatrix} \text{ with } B_2 = \begin{bmatrix} \cos(tb) & -\sin(tb) \\ \sin(tb) & \cos(tb) \end{bmatrix}.$$

There are two cases to consider, whether a is null or not.

If $a = 0$. In the case where the eigenvalue has a null real part, the $\exp(ta)$ term disappears. Let us suppose c is the smallest odd number such that $x_j \neq 0$ or $x_{j+1} \neq 0$. We first want to solve $\begin{bmatrix} y_j \\ y_{j+1} \end{bmatrix} = B_2 \begin{bmatrix} x_j \\ x_{j+1} \end{bmatrix}$. Let us remark that, since B_2 is a rotation, if $\sqrt{x_j^2 + x_{j+1}^2} \neq \sqrt{y_j^2 + y_{j+1}^2}$, there is no solution and in the other case, there is an infinity of solutions. We can express the solution of this system $t \in \alpha + \frac{2\pi}{b}\mathbb{Z}$ where α is not explicitly algebraic as its expression uses $\tan^{-1}$. Let us remark that for all those candidate t, the matrix B_2 is the same, namely $B_2 = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$. Those $\cos(\alpha)$ and $\sin(\alpha)$ are algebraic numbers that can be computed: we can write an expression in x_j, x_{j+1}, y_j and y_{j+1} for each combination of signs for those numbers.[2]

We then have to verify whether the following components of X and Y are compatible with those t. We have $\begin{bmatrix} y_{j+2} \\ y_{j+3} \end{bmatrix} = t \begin{bmatrix} y_j \\ y_{j+1} \end{bmatrix} + \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x_{j+2} \\ x_{j+3} \end{bmatrix}$. Since y_j or y_{j+1} is non null (as $\sqrt{y_j^2 + y_{j+1}^2} = \sqrt{x_j^2 + x_{j+1}^2} \neq 0$), there is then at most one solution and we can express it as an algebraic number.

Conclusion for $a = 0$. We are able to discriminate 3 possible cases: either there is no solution, either there is exactly one candidate t (defined with a fraction and a few subtractions of elements of X and Y) either there is an infinity of candidate t (defined as $\pm\alpha + \frac{2\pi}{b}\mathbb{Z}$ with the α being fractions of elements of X and Y). This last case will need to be compared with the results for the other Jordan blocks to decide whether there will be solutions or not for the whole system.

If $a \neq 0$. In the case where $a \neq 0$, the term $\exp(ta)$ makes the solution not simply turn around the origin but describe a spiral. If $a > 0$, this spiral is diverging, if $a < 0$ it is converging to the origin. We just have to study the norm of Y.

[2]For example, if $x_j > 0$, $x_{j+1} > 0$, $y_j > 0$ and $y_{j+1} > 0$, we have $\sin(\alpha) = \sqrt{\frac{y_j^2}{y_j^2 + y_{j+1}^2} \frac{x_j^2 + x_{j+1}^2}{(x_j + x_{j+1})^2}}$ and $\cos(\alpha)$ satisfies a similar expression.

We want to solve the system $e^{ta}\begin{bmatrix}\cos(tb) & -\sin(tb)\\ \sin(tb) & \cos(tb)\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} = \begin{bmatrix}y_1\\ y_2\end{bmatrix}$ with x_1 or x_2 not null (if they are, we will choose another x_j). Let us consider the norms of the two sides of this equation: $e^{ta}\sqrt{x_1^2+x_2^2} = \sqrt{y_1^2+y_2^2}$. As we have chosen x_1 or x_2 to be non null, we can write $e^{ta} = \sqrt{\frac{y_1^2+y_2^2}{x_1^2+x_2^2}}$. We hence have exactly one t candidate to be the solution. This t is the logarithm of an algebraic number and we can check whether tb is the correct angle (this is the combination of a non algebraic solution with an infinity of solutions).

Putting together the solutions. As we have seen, for one block, we may have no solution, one solution or an infinity of solutions. We must then bring the blocks together. In the case where one block has no solution, the problem is solved. In the case where there is exactly one solution, it can be algebraic (if $\lambda = 0$, or $\lambda > 0$ and there is more than one component to check), in which case it is easy to compute formally $\exp(tA)X_0$ and compare it with Y.

If we only have non explicitly algebraic solutions, we know that the solution must verify $\forall i, \exp(a_i t) = z_i$ with a_i and z_i algebraic numbers. We must then have $e^{\frac{a_1}{a_2}\ln(z_1)} = z_2$. From theorem 18, it implies that $a_1/a_2 \in \mathbb{Q}$ or $z_1 \in \{0,1\}$. $z_1 = 0$ is not compatible, $z_1 = 1$ means that t is rational and does not belong to this case. $a_1/a_2 \in \mathbb{Q}$ can be checked easily (it means the degree of the minimal polynomial is at most 1). Then we must check that $z_1^{a_1/a_2} = z_2$ which is possible for a rational exponent. This verification must be done for all pairs of a_i.

If we have several infinities of candidates, we have to decide whether those infinities have a common point. To decide whether the $\alpha_i + \frac{2\pi}{b_i}\mathbb{Z}$ intersect, we need to know whether the b_i have an integer common multiple. If they don't, then there will exist an infinity of t belonging to all those sets; if they do, only a finite number of t need to be tested.

The last case is if we have on one hand a non algebraic solution and on the other hand an infinity of solutions. We can summarize this case as the simultaneous resolution of two constraints:

$$\begin{cases} e^{at} = z \\ \begin{bmatrix}\cos(bt) & -\sin(bt)\\ \sin(bt) & \cos(bt)\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} = \begin{bmatrix}y_1\\ y_2\end{bmatrix} \end{cases}.$$

We will rephrase the second part as $\begin{bmatrix}e^{ibt} & 0\\ 0 & e^{-ibt}\end{bmatrix}\begin{bmatrix}1 & -i\\ 1 & i\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} = \begin{bmatrix}1 & -i\\ 1 & i\end{bmatrix}\begin{bmatrix}y_1\\ y_2\end{bmatrix}$.

And we can write the whole system as the following: $\begin{cases} e^{at} = z \\ e^{ibt} = z_2 \\ e^{-ibt} = z_3 \end{cases}$, where a, b, z, z_2, and z_3 are algebraic numbers (some are complex). We have already been confronted with such a system (but it had only two components) and we know that from theorem 18 it means that $i\frac{b}{a}$ belongs to $\mathbb{Q}$ or $z \in \{0,1\}$. $i\frac{b}{a} \in \mathbb{Q}$ can be verified easily as it is an algebraic number; $z = 0$ is impossible, 1 means $e^{at} = 1$ hence $a = 0$ (which belongs to another case) or $t = 0$ hence $z_2 = z_3 = 1$ in which case, $t = 0$ is a solution to the problem.

5 Open problems

5.1 (Un)decidability of Skolem–Pisot's problem

It is still open whether the continuous Skolem–Pisot problem is decidable. Some specific cases are known to be decidable, but the general problem presents complex cases which have not been proved decidable or undecidable. The state-of-the-art on this problem is described in [5].

5.2 (Un)decidability in polynomial systems with few components

As we have shown, the reachability problem is undecidable for degree 2 polynomial systems, but the number of components used is huge. Graça, Campagnolo and Buescu present in [12] a polynomial dynamical system whose reachability is undecidable. This system is of degree 56 and has 16 components[3]. It is easy to remark that for systems with only one component, reachability is decidable for polynomial systems of any degree. We can sum up the links between decidability and the degree d and number of components n:

The reachability problem for polynomial dynamical systems is

- decidable for $d = 1$;
- decidable for $n = 1$;
- undecidable for $d = 2$ and $n \simeq 17^{(2^{55})}$;
- undecidable for $d = 56$ and $n \simeq 6$.

Obviously, if the problem is undecidable for degree d and n components, it will be undecidable for any degree larger than d and number of components larger than n.

The value $17^{(2^{55})}$ is a very large over-approximation. More, as the reduction from degree 56 to degree 2 is completely generic, it is possible to get smaller values that yield undecidability. It is however an open problem whether polynomial dynamical systems with for example 5 components can simulate Turing machines. The frontier between decidability and undecidability still lies in a huge space.

6 Conclusion

The decidability and undecidability results for continuous time polynomial dynamical systems can be summed up as follows:

[3] The exact number of components is in fact not trivial to infer from the proof as many work components are used to get from the simple non smooth system to a polynomial system.

	ω-limit set	Reachability	hyperplane reach.
DS	non computable	undecidable	undecidable
polynomial DS	non computable	undecidable	undecidable
deg.2 poly DS	non computable	undecidable	undecidable
linear DS	computable	decidable	?

The decidability of the continuous Skolem-Pisot problem is open and looks like a complex problem. It is however an interesting problem that arises naturally in diverse applications of dynamical systems: it is a safety problem to check if a system should or should not reach a given region.

The frontier between decidability and undecidability may appear slight as we know that linear system have reachability decidable but for degree 2 systems it is already undecidable. It is also known that linear system plus a min operator also make this problem undecidable. However, working with few components can make a difference, as such, it would be intersting to grasp more precisely for which number of components/degree, the problem becomes undecidable.

References

[1] A , E., M , O., P , A. Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science 138* (1995), 35–65.

[2] A , E., S , G. Widening the boundary between decidable and undecidable hybrid systems. In *CONCUR 2002 - Concurrency Theory, 13th International Conference* (2002), L. Brim, P. Jancar, M. Kretínský, and A. Kucera, Eds., vol. 2421 of *Lecture Notes in Computer Science*, pp. 193–208.

[3] A , E., S , G., Y , S. On the decidability of the reachability problem for planar differential inclusions. In *Hybrid Systems: Computation and Control, HSCC 2001* (2001), M. D. D. Benedetto and A. L. Sangiovanni-Vincentelli, Eds., vol. 2034 of *Lecture Notes in Computer Science*, pp. 89–104.

[4] B , A. *Transcendental Number Theory*. Cambridge University Press, 1990.

[5] B , P., D , J.-C., J , R., B , V. D. The Continuous Skolem-Pisot Problem: On the Complexity of Reachability for Linear Ordinary Differential Equations. Tech. rep., 2008.

[6] B , V., P , N. The presence of a zero in an integer linear recurrent sequence is NP-hard to decide. *Linear algebra and its Applications 351-352* (2002), 91–98.

[7] B , V., T , J. N. A survey of computational complexity results in systems and control. *Automatica 36*, 9 (2000), 1249–1274.

[8] B , A. *Algorithmique efficace pour des opérations de base en calcul formel*. PhD thesis, École Polytechnique, december 2003.

[9] B , J. V., C , L. Irreducibles and the composed product for polynomials over a finite field. *Discrete Mathematics 65*, 2 (1987), 115–139.

[10] C , H. *A Course in Computational Algebraic Number Theory*. Springer, 1993.

[11] G , A. O. *Transcendental and Algebraic Numbers*. Dover Publications, 2003.

[12] G ¸ , D. S., B , J., C , M. L. Computational bounds on polynomial differential equations. *Applied Mathematics and Computation* (2008). In print.

[13] G ¸ , D. S., C , M. L., B , J. Robust simulations of Turing machines with analytic maps and flows. In *CiE 2005: New Computational Paradigms* (2005), S. B. Cooper, B. Löwe, and L. Torenvliet, Eds., vol. 3526 of *Lecture Notes in Computer Science*, pp. 169–179.

[14] G ¸ , D. S., C , J. F. Analog computers and recursive functions over the reals. *Journal of Complexity 19*, 5 (2003), 644–664.

[15] H , S., R , L., S , M. On the computational power of timed differentiable Petri nets. In *International Conference on Formal Modelling and Analysis of Timed Systems, FORMATS'06* (september 2006), E. Asarin and P. Bouyer, Eds., vol. 4202 of *Lecture Notes in Computer Science*, pp. 230–244.

[16] H , E. Computing omega-limit sets in linear dynamical systems. In *Unconventional Computing, UC 2008* (2008), C. S. Calude, J. F. Costa, R. Freund, M. Oswald, and G. Rozenberg, Eds., vol. 5204 of *Lecture Notes in Computer Science*, pp. 83–95.

[17] H , E. Reachability in linear dynamical systems. In *CiE 2008: Logic and Theory of Algorithms* (2008), A. Beckmann, C. Dimitracopoulos, and B. Löwe, Eds., vol. 5028 of *Lecture Notes in Computer Science*, pp. 241–250.

[18] H , V., H , T., H , M., K ¨ , J. Skolem's problem - on the border between decidability and undecidability. Tech. Rep. 683, Turku Center for Computer Science, 2005.

[19] H , M. W., S , S. *Differential Equations, Dynamical Systems, and Linear Algebra*. Academic Press, 1974.

[20] H , M. W., S , S., D , R. *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Elsevier, 2003.

[21] K , R., L , R. J. Polynomial-time algorithm for the orbit problem. *Journal of the ACM 33*, 4 (october 1986), 808–821.

[22] K , P., M , C. Closed-form analytic maps in one and two dimensions can simulate universal Turing machines. *Theoretical Computer Science 210*, 1 (1999), 217–223.

[23] L -F , J., A ` , J.-M. *Cours de mathématiques, tome 1 : algèbre*. Dunod, 1971.

[24] L , E. N. Deterministic non-periodic flow. *Journal of the Atmospheric Sciences 20* (1963), 130–141.

[25] M , M. An inequality about factors of polynomials. *Mathematics of Computation 28*, 128 (1974), 1153–1157.

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

[27] M , J. D. *Mathematical Biology*, second ed., vol. 19 of *Biomathematics*. Springer, Berlin, Germany, 1993.

[28] S , C. E. Mathematical theory of the differential analyzer. *J. Math. Phys. MIT 20* (1941), 337–354.

[29] G , J., G , J. *Modern Computer Algebra*. Cambridge University Press, 2003.

A Appendix

A.1 Factorizing a polynomial in $\mathbb{Q}[X]$

The characteristic polynomial $\chi_A(X)$ of the matrix $A \in \mathbb{Q}^{n\times n}$ belongs to $\mathbb{Q}[X]$. We will first factorize $\chi_A(X)$ in $\mathbb{Q}[X]$ to obtain some square-free polynomials. This is a classical problem. One solution is to use Yun's algorithm [29, p. 371] that writes our polynomial χ_A into the form

$$\chi_A = \prod_i R_i^i$$

where the R_i are square-free and do not share roots. The polynomial $\prod R_i$ is then a square-free polynomial that has the same roots as P.

Proposition 37 *Suppose given a polynomial P that we can write as*

$$P = \prod (X - \alpha_j)^{\beta_j}$$

with the α_j distinct. Let $Q = P/\gcd(P, P')$, then Q is square-free and

$$Q = \prod (X - \alpha_j).$$

We then want to factorize this polynomial Q in irreducible factors in $\mathbb{Q}[X]$. This problem is again a classical problem. An algorithm that achieves this goal is for example presented in [10, p. 139].

Proposition 38 *Given a square-free polynomial $P \in \mathbb{Q}[X]$, we can compute its factorization in $\mathbb{Q}[X]$.*

So we have obtained $Q = \prod Q_i$ with the Q_i being polynomial that are irreducible in $\mathbb{Q}[X]$

A.2 Computing the roots

To obtain χ_A's roots, we are going to compute the roots of Q. Those are algebraic numbers. We only then need to compute a representation of each of those roots. It means finding the minimal polynomial and giving a rational approximation of the root and an error bound to discriminate other roots of the minimal polynomial. Let us consider a Q_i.

There can be both real roots and complex roots that are not real. Sturm's theorem allows us to know the number of each of them [10, pp. 153-154]. We can then find the real roots with, for example, Newton's iteration algorithm [29, sec. 9.4]. The complex roots will for example be computed with Schönhage's method.

From this, we obtain approximations of the roots of the polynomial Q_i. Let α_j be one of those roots. The minimal polynomial of α_j divides Q_i and belongs to $\mathbb{Q}[X]$. As Q_i

is irreducible in $\mathbb{Q}[X]$, the minimal polynomial can only be Q_i (1 has no root and hence cannot be a minimal polynomial).

We then obtain a factorization of Q as $\prod(X - \alpha_j)$ with the α_j explicitly defined as algebraic numbers.

A.3 Jordanizing the matrix

The final step to be able to use the method described earlier is to do the factorization of χ_A in $\mathbb{C}[X]$. In fact, it is sufficient to do it in $\mathbb{Q}(\{\alpha_j\})[X]$ to obtain a factorization into monomials. So from now on, we will work in $\mathbb{Q}(\alpha_j)$ which is the field generated from $\mathbb{Q}$ and the algebraic numbers $\{\alpha_j\}$.

To find the multiplicity of each root, we just need to know how many times the minimal polynomial divides χ_A. We then obtain a decomposition

$$\chi_A(X) = \prod(X - a_i)^{b_i} \prod((X - \alpha_i)(X - \bar{\alpha}_i))^{\beta_i}$$

with the α_i being the complex not real roots and the a_i the real roots.

The Jordan blocks composing the matrix can be either $\begin{pmatrix} a_i & & & \\ 1 & a_i & & \\ & \ddots & \ddots & \\ & & 1 & a_i \end{pmatrix}$ or $\begin{pmatrix} B & & & \\ I_2 & B & & \\ & \ddots & \ddots & \\ & & I_2 & B \end{pmatrix}$ with $B = \begin{bmatrix} p & -q \\ q & p \end{bmatrix}$ for $\alpha_i = p + \mathrm{i}q$. Note that an eigenvalue can be responsible for more than one block. The number of different blocks an eigenvalue λ creates is $\dim(\ker(A - \lambda))$. Similarly, let $\delta_i = \dim(\ker(A - \lambda)^i)$, $\delta_{i+1} - \delta_i$ is the number of blocks of size at least $i + 1$. We can hence know the number of blocks of each size and write a Jordan matrix J consisting of blocks in decreasing size order (any order would be fine). This Jordan matrix is similar to the original matrix A.

We finally need to compute the similarity matrix P which will be such that $A = P^{-1}JP$. This matrix is obtained by computing the eigenvectors of the matrix A (or J).