



HAL
open science

Computing omega-limit Sets in Linear Dynamical Systems

Emmanuel Hainry

► **To cite this version:**

Emmanuel Hainry. Computing omega-limit Sets in Linear Dynamical Systems. Unconventional Computation, Aug 2008, Vienne, Austria. pp.83–95, 10.1007/978-3-540-85194-3_9 . inria-00250111v3

HAL Id: inria-00250111

<https://inria.hal.science/inria-00250111v3>

Submitted on 21 Aug 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Computing omega-limit Sets in Linear Dynamical Systems

Emmanuel Hainry

LORIA, Université Henri Poincaré

Campus scientifique, BP 239 - 54506 Vandœuvre-lès-Nancy, France

`Emmanuel.Hainry@loria.fr`

Dynamical systems allow to modelize various phenomena or processes by only describing their way of evolution. It is an important matter to study the global and the limit behaviour of such systems. A possible description of this limit behaviour is via the omega-limit set: the set of points that can be limit of subtrajectories. The omega-limit set is in general uncomputable. It can be a set highly difficult to apprehend. Some systems have for example a fractal omega-limit set. However, in some specific cases, this set can be computed. This problem is important to verify properties of dynamical systems, in particular to predict its collapse or its infinite expansion. We prove in this paper that for linear continuous time dynamical systems, it is in fact computable. More, we also prove that the ω -limit set is a semi-algebraic set. The algorithm to compute this set can easily be derived from this proof.

Keywords: Dynamical Systems, omega-limit set, hybrid systems, reachable set, verification, safety properties.

1. Introduction

The physical equations that govern interactions between celestial bodies give a local description of the trajectory of those bodies: given the positions and speeds of all the stars and planets, we know the evolution of those variables. Other systems, motivated by meteorological phenomena, chemical interactions, biological examples, mathematics equations or computing systems can be described in a similar local manner: given any set of instantaneous conditions, the local behaviour of the system is defined. Those examples can be described as dynamical systems. A dynamical system behaves either in discrete time, either in continuous time. In both cases, it will be defined by an initial position and a dynamics map. In the discrete case, the dynamics can, from the conditions at time n , predict the positions at time $n + 1$. In the continuous case, the direction in which the system moves from a given state point x is a function of x .

The evolution of a dynamical system is hence described in a very simple way but it can be hard to grasp where a point that undergoes the dynamics will go. Hence, one of the fundamental questions with such systems is their asymptotic behaviour. Knowing whether they collapse to one single point, diverge or become periodic is important to grasp the evolution of a dynamical system. The case of celestial bodies is a fine example of the complexity of this problem: we can predict the whole trajectory of a system with two bodies, but as soon as there are three or more bodies, it becomes undecidable to know whether the bodies will not eventually collide.

Dynamical systems are much studied or used to describe various phenomena that can belong to mathematics [12], physics, biology [18]... The famous Lorenz' attractor [15] is an example of a dynamical system describing a meteorological phenomenon. However, as standard as those systems are, and as simple as the description of their dynamics may be, many important problems such as limit and reachability are undecidable. The challenge is of interest in computer science as computational models can be modeled by dynamical systems. Hybrid systems in particular rely on dynamical systems plus some discrete behaviour and as such, if a problem is difficult in dynamical systems, it is bound to be more difficult in hybrid systems.

The difficulty of the prediction of the trajectory of dynamical systems is testified by many undecidability results for natural problems on such systems. Some problems are decidable but undecidability comes fast into the picture. Even considering polynomial systems yields many undecidable problems: [10] for example shows that it is possible to simulate a Turing machine using a polynomial dynamical system. It is hence undecidable whether or not a trajectory will reach the region corresponding to the halting state of the machine. This particular problem can be seen as a continuous version of the Skolem-Pisot problem [17, 4, 11] which studies whether a component of a discrete linear system will reach 0. This problem is not different from deciding if this system reaches a hyperplan of the space, described by $y_k = 0$ where k is the number of the component considered.

The (point to point) reachability problem, which is undecidable in the general case, has been shown undecidable for various restricted classes of dynamical systems, such as Piecewise Constant Derivative systems [7] 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, 5].

The problem of the limit set of dynamical systems is also undecidable in the general case. It is of interest for ensuring safety properties such as the absence of infinite expansion or the ultimate presence in a given region. In this paper, we will study this problem, more precisely the ω -limit set, in a simple class of continuous-time dynamical systems: linear dynamical systems. As Turing machines can be encoded in dynamical systems, the description of the ω -limit set would give an answer to the halting problem hence it is not decidable in polynomial dynamical systems. However, this article proves that the ω -limit set is computable in linear dynamical systems and gives a way to compute a semi-algebraic representation of this set.

The section 2 presents the problems we are going to solve and mathematical notions that will be useful in the following. The next sections are the core of this paper: they prove the main result of this paper, Theorem 1, that asserts that the ω -limit set of a given linear system is semi-algebraic and thus computable. Part 3 recalls that putting the matrix into Jordan form is doable. Then part 4 shows how to solve the problem in the specific case where the matrix is in Jordan form.

2. Prerequisites

In this section we will first present the problems that motivate this document and some basic definitions and results on polynomials and matrices.

2.1. 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 [13].

Définition 1 (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:*

$$\begin{cases} X' &= AX \\ X(0) &= X_0. \end{cases} \quad (1)$$

X is called a trajectory of the system.

Définition 2 (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 as the solution of 1.*

Définition 3 (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 $Y = \lim_{n \rightarrow +\infty} X(t_n)$.*

Définition 4 (ω -limit sets) *The ω -limit set of a dynamical system is the set of its ω -limit points: $\omega(X) = \overline{\bigcap_n \bigcup_{t > n} X(t)}$, where \overline{A} is the closure of the set A .*

Définition 5 (semi-algebraic set) *A subset S of \mathbb{R}^n is called semi-algebraic if it can be defined by a finite sequence of polynomial equations or a finite union of sets so described. Formally, it can be written*

$$S = \bigcup_{i=1}^m \left\{ x \in \mathbb{R}^n; \bigwedge_j p_{i,j}(x) = 0 \wedge \bigwedge_l p_{i,l}(x) > 0 \right\}$$

Let us now define the problem that we will be interested in solving: the ω -limit set problem.

Problem 1 (ω -limit set) *Given a dynamical system, compute a representation of its ω -limit set.*

The theorem 1 gives an answer to this problem for linear dynamical systems and proves that the ω -limit set is semi-algebraic in this case.

2.2. 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} . We will usually use \mathbb{Q} as this field.

Définition 6 (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 $a_i \in \mathbb{K}$. The integer n is the degree of P .

Définition 7 (Roots of a polynomial) The set $Z(P)$ of roots of a polynomial P is defined as $Z(p) = \{x \in \mathbb{C}; P(x) = 0\}$

Définition 8 (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 [16] shows a bound on the distance between roots of a polynomial from its coefficient.

Définition 9 (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 , $a + ib$ is an approximation of α such that $|\alpha - (a + ib)| < \rho$ and α is the only root of P in the open ball $\mathcal{B}(a + ib, \rho)$.

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: indeed the approximation and bound are easy to obtain, and the minimal polynomial can be obtained using classical properties of the resultant that gives the polynomial whose roots are the $H(\alpha_i, \beta_j)$ with H a polynomial. See [6, 8] for details.

We will also use the term commensurable in a specific way: two numbers are commensurable if one is a multiple of the others by a rational factor.

Définition 10 (Commensurable numbers) Two numbers a and b are commensurable if there exists a rational number p/q such that $a = \frac{p}{q}b$.

Let us note that it is easy to check if two algebraic numbers are commensurable: given the representations of those two numbers, we know how to compute the representation of the fraction of those numbers. Then it suffices to check if this fraction is rational which is equivalent to the minimal polynomial being of degree 1.

Proposition 1 (\mathbb{Q} -linear independent algebraic numbers) Given the representations of n algebraic numbers p_1, \dots, p_n , it is decidable whether they are \mathbb{Q} -linear dependent which means there exists $(\alpha_1, \dots, \alpha_n) \in \mathbb{Q}^n - (0)$ such that $\sum \alpha_i p_i = 0$. If so, then this n -uple is computable.

2.3. Matrices

Définition 11 (Characteristic polynomial) Given a matrix $A \in \mathbb{K}^{n \times n}$, its characteristic polynomial is $\chi_A(X) = \det(XI_n - A)$

Définition 12 (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.

Given a square matrix A , we want to solve the Cauchy problem (1). To do that, we will first put the matrix into a useful form: Jordan's form and then compute the exponential of the matrix as it is known that the solution of the linear differential equation will be closely related to the exponential of the matrix A . 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 [13] or [14].

Définition 13 (Jordan block) A Jordan block is a square matrix of one of the two following forms

$$\begin{pmatrix} \lambda & & & \\ 1 & \lambda & & \\ & \ddots & \ddots & \\ & & 1 & \lambda \end{pmatrix}$$

or

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

Définition 14 (Jordan form) A matrix that contains Jordan blocks on its diagonal is said to be in Jordan form.

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

Proposition 2 ([14]) 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. Computing the Jordan form of a matrix

We are given a matrix A and an initial vector X_0 containing rational elements. We want compute a formal solution of the Cauchy problem (1). To do that, we will compute the Jordan form of this matrix and the similarity matrices. The process of putting the matrix A into Jordan form is a classical one which consists in four parts which are detailed in appendix:

- 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).

Let us note that the matrix we obtain is composed of algebraic numbers, hence we know how to compute on those matrices.

4. Computing the ω -limit set of a dynamical system

Let us now suppose that the matrix A is in Jordan form and that A and X_0 are composed of algebraic numbers. Our goal is to compute the ω -limit set of the dynamical system defined by the differential equation (1).

4.1. Computing the solution of the Cauchy problem

Let us first remark that the solution of this differential equation is simple to express. We have

$$A = \begin{pmatrix} D_1 & 0 & \cdots & 0 \\ 0 & D_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & D_k \end{pmatrix}$$

with the D_i being Jordan blocks of the form

$$D_i = \begin{pmatrix} \lambda & & & \\ 1 & \lambda & & \\ & \ddots & \ddots & \\ & & 1 & \lambda \end{pmatrix} \quad (2)$$

or

$$D_i = \begin{pmatrix} B & & & \\ I_2 & B & & \\ & \ddots & \ddots & \\ & & I_2 & B \end{pmatrix} \text{ with } B = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \text{ and } I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (3)$$

The solution of the Cauchy problem is then $X(t) = \exp(tA)X_0$, which we can write as

$$X(t) = \begin{pmatrix} \exp(tD_1) & & & \\ & \exp(tD_2) & & \\ & & \ddots & \\ & & & \exp(tD_k) \end{pmatrix} X_0$$

And computing the $\exp(tD_i)$ is simple: in case D_i is of the form (2),

$$\exp(tD_i) = e^{t\lambda} \begin{pmatrix} 1 & & & & \\ t & 1 & & & \\ \frac{t^2}{2} & t & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \\ \frac{t^m}{m!} & \cdots & \frac{t^2}{2} & t & 1 \end{pmatrix};$$

if on the other hand, D_i is of the form (3), then

$$\exp(tD_i) = e^{t\alpha} \begin{pmatrix} B_2 & & & & \\ tB_2 & B_2 & & & \\ \frac{t^2}{2}B_2^2 & tB_2 & B_2 & & \\ \vdots & \ddots & \ddots & \ddots & \\ \frac{t^m}{m!}B_2^m & \cdots & \frac{t^2}{2}B_2^2 & tB_2 & B_2 \end{pmatrix} \text{ with } B_2 = \begin{bmatrix} \cos(tb) & -\sin(tb) \\ \sin(tb) & \cos(tb) \end{bmatrix}.$$

4.2. Simplifying the matrix

We can without losing information delete the Jordan blocks corresponding to zeros of the initial vector. Indeed those blocks have no impact on the behaviour of the system as the corresponding components will forever stay 0. We can in the same optic remove certain lines and columns from the Jordan blocks if they will forever stay 0.

Let us write X_{0_i} for the components of X_0 corresponding to each D_i . Formally, $X_0 = \begin{pmatrix} X_{0_1} \\ X_{0_2} \\ \vdots \\ X_{0_k} \end{pmatrix}$ and the size of X_{0_i} being equal to the size of D_i . This way, we can write $X(t) = \begin{pmatrix} \exp(tD_1)X_{0_1} \\ \exp(tD_2)X_{0_2} \\ \vdots \\ \exp(tD_k)X_{0_k} \end{pmatrix}$.

If $X_{0_i} = 0$, then $\forall t, X_i(t) = 0$. Hence we need not consider the i -th block to compute the system.

If the l first components of X_{0_j} are 0, then we can erase the l first lines and columns from the j -th Jordan block if it is of form (2) and only the $2 \lfloor \frac{l}{2} \rfloor$ first lines and columns if the block is in form (3).

We then obtain a representation of the solution where all Jordan blocks are useful and all dimensions of the Jordan blocks have a repercussion on the result. From now on, when we

will talk of the multiplicity of a Jordan block, it will refer to its size in this new matrix, more, for the case (3) as both $a + ib$ and its conjugate have same influence, the multiplicity of the corresponding Jordan block will be half the size of the matrix.

Définition 15 (Multiplicity of a Jordan block) *Let $D \in \mathbb{R}^{m \times m}$ be a Jordan block. If D is of the form (2), its multiplicity is defined as being m . If D is of the form (3), its multiplicity is defined as being $\frac{m}{2}$.*

Notice that for the case (3), the size of the matrix is even. Notice also that since an eigenvalue can be responsible for more than one Jordan block, the multiplicity of a Jordan block is not the same as the multiplicity of the eigenvalue.

4.3. Computing the ω -limit set

There are now a few cases to consider. We will first consider the most simple cases and finish with the more complicated ones. The cases we will consider are

- there is an eigenvalue with positive real part;
- there is an eigenvalue with null real part corresponding to a Jordan block of multiplicity > 1 ;
- there are only eigenvalues with negative real part;
- other cases (the only eigenvalues with non negative real part have null real part and multiplicity one).

If an eigenvalue has a positive real part, then a term $e^{\lambda t}$ appears with $\lambda > 0$. It means that the corresponding component will grow unboundedly. Hence the ω -limit set is empty.

If an eigenvalue has a null real part, the exponential part disappears, and the first component will have a bounded trajectory. However, since we suppose that the multiplicity is greater than one, a t factor will have to be taken into account. This factor makes the second component grow unboundedly and hence the ω -limit set is empty.

In the case where all the eigenvalues have a negative real part, all components will have a decreasing exponential in their expression and since for all integer m , $t^m \exp(-t)$ converges towards 0, all components will converge towards 0.

Otherwise, there will appear a trajectory that stay in a given region of the space and can either be periodic (circles, or multi-dimensional Lissajous curves), either be dense in a specific semi-algebraic set.

Théorème 1 *Given a linear dynamical system, its ω -limit set is computable and is a semi-algebraic set.*

Proof: Let us 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 0 hence no real point will be a limit of a subtrajectory.

- 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

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

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 $ib_1, -ib_1, \dots, ib_n, -ib_n$, with the b_i being real algebraic numbers. There are two cases to consider: either the family (b_1, b_2, \dots, b_n) is \mathbb{Q} linearly independent, either it is not.

- Let us assume the (b_1, \dots, 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 $x_{2k+1}^2 + x_{2k+2}^2 = x_{2k+1_0}^2 + x_{2k+2_0}^2$. Indeed, it is trivial if $n = 1$. Let us consider it true for $n = k$. It means that for any given point $(\alpha_{1_1}, \alpha_{1_2}, \dots, \alpha_{k_1}, \alpha_{k_2}, \alpha_{k+1_1}, \alpha_{k+1_2})$ of that set, there exists a sequence of times $(t_i)_{i \in \mathbb{N}}$ such that

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

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, \dots, 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 $t_\phi^*(i_0) \in V_{i'} \cap V_{j'}$, then we have

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

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

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

– Let us assume there exists $\alpha_1, \dots, \alpha_n \in \mathbb{Q}^n$ 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

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

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 $X_i = (x_{2i-1} + ix_{2i})$, we have

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

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

References

- [1] Eugene Asarin, Oded Maler, and Amir Pnueli. Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science*, 138:35–65, 1995.
- [2] Eugene Asarin and Gerardo Schneider. Widening the boundary between decidable and undecidable hybrid systems. In Lubos Brim, Petr Jancar, Mojmir Kretínský, and Antonín Kucera, editors, *CONCUR 2002 - Concurrency Theory, 13th International Conference*, volume 2421 of *Lecture Notes in Computer Science*, pages 193–208. Springer, 2002.
- [3] Eugene Asarin, Gerardo Schneider, and Sergio Yovine. On the decidability of the reachability problem for planar differential inclusions. In Maria Domenica Di Benedetto and Alberto L. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control, 4th International Workshop, HSCC 2001*, volume 2034 of *Lecture Notes in Computer Science*, pages 89–104. Springer, 2001.
- [4] Jean Berstel and Maurice Mignotte. Deux propriétés décidables des suites récurrentes linéaires. *Bulletin de la Société Mathématique de France*, 104:175–184, 1976.

- [5] Vincent Blondel and John N. Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36(9):1249–1274, 2000.
- [6] Alin Bostan. *Algorithmique efficace pour des opérations de base en calcul formel*. PhD thesis, École polytechnique, décembre 2003.
- [7] Olivier Bournez. *Complexité algorithmique des systèmes dynamiques continus et hybrides*. PhD thesis, École Normale Supérieure de Lyon, janvier 1999.
- [8] Joel V. Brawley and Leonard Carlitz. Irreducibles and the composed product for polynomials over a finite field. *Discrete Mathematics*, 65(2):115–139, 1987.
- [9] Henri Cohen. *A Course in Computational Algebraic Number Theory*. Springer, 1993.
- [10] Daniel S. Graça, Manuel L. Campagnolo, and Jorge 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*, volume 3526 of *Lecture Notes in Computer Science*, pages 169–179. Springer, 2005.
- [11] Vesa Halava, Tero Harju, Mika Hirvensalo, and Juhani Karhumäki. Skolem’s problem - on the border between decidability and undecidability. Technical Report 683, Turku Center for Computer Science, 2005.
- [12] Morris W. Hirsch, Stephen Smale, and Robert Devaney. *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Elsevier Academic Press, 2003.
- [13] Morris W. Hirsch and Steve Smale. *Differential Equations, Dynamical Systems, and Linear Algebra*. Academic Press, 1974.
- [14] Jacqueline Lelong-Ferrand and Jean-Marie Arnaudiès. *Cours de mathématiques, tome 1 : algèbre*. Dunod, 1971.
- [15] Edward N. Lorenz. Deterministic non-periodic flow. *Journal of the Atmospheric Sciences*, 20:130–141, 1963.
- [16] Maurice Mignotte. An inequality about factors of polynomials. *Mathematics of Computation*, 28(128):1153–1157, 1974.
- [17] Maurice Mignotte. Suites récurrentes linéaires. In *Séminaire Delange-Pisot-Poitou. Théorie des nombres*, volume 15, pages G14–1–G14–9, 1974.
- [18] James Dickson Murray. *Mathematical Biology*, volume 19 of *Biomathematics*. Springer Verlag, Berlin, Germany, second edition, 1993.
- [19] Joachim von zur Gathen and Jürgen Gerhard. *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 [19, 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 3 *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 [9, p. 139].

Proposition 4 *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 [9, pp. 153-154]. We can then find the real roots with, for example, Newton's iteration algorithm [19, 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 different Jordan blocks composing the matrix are either $\begin{pmatrix} a_i & & & \\ 1 & a_i & & \\ & \ddots & \ddots & \\ & & 1 & a_i \end{pmatrix}$ either $\begin{pmatrix} B & & & \\ I_2 & B & & \\ & \ddots & \ddots & \\ & & I_2 & B \end{pmatrix}$ with $B = \begin{bmatrix} p & -q \\ q & p \end{bmatrix}$ for $\alpha_i = p + iq$. 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).