

# Computability of limit sets for two-dimensional flows

Daniel S. Graça

Universidade do Algarve, C. Gambelas, 8005-139 Faro, Portugal  
& Instituto de Telecomunicações, Portugal

N. Zhong

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

July 5, 2021

## Abstract

A classical theorem of Peixoto qualitatively characterizes, on the two-dimensional unit ball, the limit sets of structurally stable flows defined by ordinary differential equations. Peixoto's density theorem further shows that such flows are typical in the sense that structurally stable systems form an open dense set in the space of all continuously differentiable flows.

In this note, we discuss the problem of explicitly finding the limit sets of structurally stable planar flows.

## 1 Introduction

In this note, we discuss limit sets of planar  $C^1$  dynamical systems from the viewpoint of computability.

The dynamical systems to be considered are of the type

$$\dot{x} = f(x) \tag{1}$$

where  $f : E \rightarrow \mathbb{R}^2$  is continuously differentiable ( $C^1$ ) on  $E$  - either an open or a compact subset of  $\mathbb{R}^2$  (if  $E$  is compact, then  $f$  is assumed to be  $C^1$  in some open set containing  $E$ ),  $t \in \mathbb{R}$  is the independent variable, and  $\dot{x} = dx/dt$ . The solution to the equation with the initial condition  $x(0) = x_0$  is a function  $\phi(f, x_0)$  of time  $t$  that describes the time dependence of  $x_0$  in the phase space, either  $\mathbb{R}^2$  or a subset of  $\mathbb{R}^2$ . The function  $\phi(f, \cdot)(\cdot)$  is called the flow defined by the vector field  $f$  and  $\phi(f, x_0)(\cdot)$  is called a trajectory (or orbit) passing through  $x_0$ .

Since the trajectories can be defined for arbitrarily long terms and explicit solution formulas do not exist for most dynamical systems, it becomes necessary and essential to study asymptotic (long term) behaviors of the trajectories. The topic is extensively studied in mathematics and physics. The asymptotic behavior of a dynamical system is captured by its limit sets, which are the states the trajectories approach to or land on as  $t \rightarrow \pm\infty$ . The limit sets are well understood *qualitatively* for  $C^1$  planar dynamical systems: a closed and bounded limit set other than an equilibrium point or a periodic orbit consists of equilibria and solutions connecting them according to the Poincaré-Bendixson Theorem. From the *quantitative* perspective, limit sets of planar flows remain elusive; there are a number of open problems in the field, including Hilbert's 16th problem. The second part of Hilbert's 16th problem asks for the maximum number and relative positions of periodic orbits of planar polynomial (real) vector fields of a given degree. The problem is open even for the simplest nonlinear flows - the quadratic flows.

It is well known that the operator  $(f, x_0) \rightarrow \phi(f, x_0)(\cdot)$  (as a function of  $t$ ) is computable (see [GZB12] and references therein). Intuitively, this means that there is an algorithm that plots a polygon curve  $p(t)$  on a computer screen satisfying  $\max_{-T \leq t \leq T} \|\phi_t(f, x_0) - p(t)\| \leq 2^{-n}$  for every natural number  $n$ , every rational number  $T$ , and “good enough” information on  $f$  and  $x_0$ . (It is a convention to write  $\phi_t(f, x_0)$  for  $\phi(f, x_0)(t)$ .) However, the algorithm is local in the sense that it provides little information on asymptotic behaviors of the trajectories. On the other hand, the limit sets are asymptotic and global in nature - global properties are generally more difficult to deal with in classical mathematics and to compute in numerical as well as in computable analysis. It turns out that there are  $C^1$  computable planar flows whose periodic orbits are badly non-computable. This is our first theorem.

**Theorem 1** *For any  $k \geq 1$ , there is a  $C^k$  computable function  $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  such that none of the periodic orbits of the  $C^k$  planar system  $\dot{x} = f(x)$ ,  $x \in \mathbb{R}^2$ , is r.e. or co-r.e. as a closed subset of  $\mathbb{R}^2$ .*

Intuitively, the theorem says that it is impossible to plot any periodic orbit of the flow on a computer screen, not even a good adumbration of it. Then, under what conditions can the qualitatively well-understood limit sets of a planar flow be computable - quantitatively plotted with arbitrarily high magnification? Our second theorem provides an answer.

**Theorem 2** *There is an algorithm that locates the positions of equilibrium points and periodic orbits with arbitrarily high precision for any structurally stable  $C^1$  planar vector field defined on the closed unit disk. Moreover, the computation is uniform on the set of all structurally stable planar vector fields.*

Recall that the density theorem of Peixoto [Pei62, Theorem 2] shows that, on two-dimensional compact manifolds, structurally stable systems are “typical”

in the sense that such systems form a dense open subset in the set of all  $C^1$  planar systems. Hence, Theorem 2 says that the limit set of a *typical* differential equation (1) defined on the unit disk of  $\mathbb{R}^2$ , where  $f$  is of class  $C^1$ , is computable.

## 2 Preliminaries

In this section, we recall necessary definitions. We begin with definitions related to computable analysis with assumption that the reader is familiar with the classical computable functions from  $\mathbb{Z}_1$  to  $\mathbb{Z}_2$ , where  $\mathbb{Z}_i$  is the set of (or the set of tuples of) natural numbers ( $\mathbb{N}$ ), integers ( $\mathbb{Z}$ ), or rational numbers ( $\mathbb{Q}$ ). In computable analysis, informally speaking, an operator  $\Phi : X \rightarrow Y$  is computable if

- (1) elements in  $X$  and  $Y$  can be encoded by sequences of exact “functions” of finite size (such as rational numbers, polynomials with rational coefficients, polygonal curves with rational corners, etc) which converge to those elements at a known rate of convergence; such sequences are called names of the corresponding elements; and
- (2) there is a computer (a Turing machine, an algorithm or a computer program) that outputs an approximation to  $\Phi(x)$  within accuracy  $2^{-n}$  on input of  $n$  (accuracy) and (a name of)  $x$ .

To execute evaluations in practice, an infinite input datum - a name of  $x$  - can be conveniently treated as an interface to a program computing  $\Phi$ : for every  $n \in \mathbb{N}$ , the name supplies a good enough (finite size) approximation  $p$  of  $x$  to the program, the program then performs computations based on inputs  $n$  and  $p$ , and returns a (finite size) approximation  $q$  of  $\Phi(x)$  with an error bounded by  $2^{-n}$ . This is often termed as  $\Phi(x)$  is computable from (a name of)  $x$ . For more details the reader is referred to e.g. [BHW08].

The following is the precise definition.

**Definition 3** 1. A name of a real number  $x$  is a function  $a : \mathbb{N} \rightarrow \mathbb{Q}$  such that  $|x - a(n)| \leq 2^{-n}$ . If the function  $a$  is (classically) computable, then  $x$  is said to be computable.

2. Let  $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  be a  $C^k$  function, and let  $B = \{x \in \mathbb{R}^2 : \|x\| \leq r\}$ , where  $r$  is a rational number. A  $C^k$ -name of  $f$  on  $B$  is a sequence  $\{P_l\}$  of polynomials with rational coefficients such that  $d^k(f, P_l) \leq 2^{-l}$ , where

$$d^k(f, P_l) = \max_{0 \leq j \leq k} \max_{x \in B} \|D^j f(x) - D^j P_l(x)\|.$$

In particular:

- A.  $f$  is said to be ( $C^k$ -) computable on  $B$  if there is a Turing machine (or a computer) that outputs a  $C^k$ -name  $\{P_l\}$  of  $f$  in the following sense: on input  $l$  (accuracy), it outputs the rational coefficients of the polynomial  $P_l$ .

*B. Or, equivalently,  $f$  is said to be  $(C^k-)$  computable on  $B$  if there is an oracle Turing machine such that for any input  $l \in \mathbb{N}$  (accuracy) and any name of  $x \in B$  given as an oracle, the machine will output the rational vectors  $q_0, q_1, \dots, q_k$  in  $\mathbb{R}^2$  such that  $\|q_j - D^j f(x)\| \leq 2^{-l}$  for all  $0 \leq j \leq k$  (see e.g. [Ko91], [BY06]).*

As already mentioned above, an oracle can be conveniently treated as an interface to a program computing  $f$  in practice.

We turn now to define computable open and closed subsets of  $\mathbb{R}^2$ . In  $\mathbb{R}^2$ , computability can be intuitively visualized by plotting pixels: a subset of  $\mathbb{R}^2$  is computable if it can be plotted on the screen of a computer with arbitrarily high magnification. The following definition shares this spirit.

**Definition 4** *Let  $U$  be an open subset of  $\mathbb{R}^2$ , and let  $C$  be a closed subset contained in  $B$ , where  $B$  is a closed disk of  $\mathbb{R}^2$  centered at the origin with a rational radius.*

1.  *$U$  is said to be r.e. open if there are computable functions  $a : \mathbb{N} \rightarrow \mathbb{Q}^2$  and  $r : \mathbb{N} \rightarrow \mathbb{Q}$  such that  $U = \cup_{n=1}^{\infty} B(a(n), r(n))$ , where  $B(a(n), r(n))$  is the open disk centered at  $a(n)$  with radius  $r(n)$ . In other words  $U$  can be filled up by the fattened pixels at  $a_n$  - the open disk  $B(a_n, r_n)$ .*
2.  *$C$  is called co-r.e. closed if  $B \setminus C$  is r.e. open in  $B$ .  $C$  is called r.e. closed if  $C$  has a computable dense sequence.  $C$  is called computable if it is co-r.e. and r.e. closed. Or, equivalently, if there is a Turing machine that, on input  $n \in \mathbb{N}$  (accuracy), outputs finite sequences  $r_j \in \mathbb{Q}$  and  $a_j \in \mathbb{Q}^2$ ,  $1 \leq j \leq j(n)$ , such that  $d_H(C, B \setminus \cup_{j=1}^{j(n)} B(a_j, r_j)) \leq 2^{-n}$ , where  $d_H(\cdot, \cdot)$  denotes the Hausdorff distance between two compact subsets of  $\mathbb{R}^2$ .*

We observe that the r.e. openness is a local property - one pixel at a time - and the plotting does not give any adumbration of the whole picture of  $U$ . On the other hand, co-r.e. closeness is a global property: Assume that  $C = K \setminus \cup B(a_n, r_n)$ . If one plots  $B(a_n, r_n)$  as before, then at each step one obtains a set (the portion not covered by the pixels) containing the entire  $C$ , an adumbration of  $C$ .

Now we turn to define structurally stable planar dynamical systems. Let  $K \subseteq \mathbb{R}^2$  be a compact set, and let  $f : K \rightarrow \mathbb{R}^2$  be a  $C^1$  function. The  $C^1$ -norm is used for  $C^1$  functions

$$\|f\|_1 = \max_{x \in K} \|f(x)\| + \max_{x \in K} \|Df(x)\|$$

**Definition 5** *The system (1), where  $f : K \rightarrow \mathbb{R}^2$  is of class  $C^1$ , is structurally stable if there exists some  $\varepsilon > 0$  such that for all  $g \in C^1(K)$  satisfying  $\|f - g\|_1 \leq \varepsilon$ , the trajectories (orbits) of*

$$y' = g(y) \tag{2}$$

are homeomorphic to the trajectories of (1), i.e. there exists some homeomorphism  $h$  such that if  $\gamma$  is a trajectory of (1), then  $h(\gamma)$  is a trajectory of (2). Moreover, the homeomorphism  $h$  is required to preserve the orientation of trajectories by time.

Intuitively, (1) is structurally stable if the shape of its dynamics is (globally) robust to small perturbations. If  $K$  is the unit disk  $\mathbb{D} = \{x \in \mathbb{R}^2 : \|x\| \leq 1\}$ , as in Theorem 2, it is common to assume that the vector field points inwards along the boundary of  $\mathbb{D}$ . Otherwise, the system may be structurally unstable if the flow is tangent to a point on the boundary. We note that not all planar systems are structurally stable. Explicit examples of structurally unstable systems can be found in e.g. [HSD04, Figure 9.4 in p. 193]. However, due to Peixoto's density theorem, structurally stable systems are typical in the sense that structurally stable systems form an open dense set in the space of all continuously differentiable flows.

### 3 Proof of Theorem 1

We now proceed with the proof of Theorem 1. We begin by recalling a theorem by Weihrauch (Theorem 4.2.8. [Wei00]): the countable set  $\mathbb{R}_c$  of all computable real numbers can be covered by the union of a computable sequence of open intervals,  $I_n = (\alpha_n, \beta_n)$ , such that the length of  $\sum I_n$  is at most 1. Let  $A = \mathbb{R} \setminus \bigcup I_n$ . Then  $A \neq \emptyset$  is co-r.e. closed and none of points in  $A$  is computable.

Fix  $k \in \mathbb{N}$ . Now we construct a  $C^k$  computable function  $g : \mathbb{R} \rightarrow \mathbb{R}$  such that  $g(x) = 0$  if and only if  $x \in A$ . Let  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  be the computable  $C^\infty$  standard bump function

$$\phi(x) = \begin{cases} e^{-\frac{x^2}{1-x^2}} & \text{if } |x| < 1 \\ 0 & \text{otherwise} \end{cases}$$

The function  $g : \mathbb{R} \rightarrow \mathbb{R}$  is defined by the following formula: for every  $x \in \mathbb{R}$ ,

$$g(x) = \sum_{n=1}^{\infty} \phi\left(\frac{x - \frac{\alpha_n + \beta_n}{2}}{r_n}\right) 2^{-n}$$

where  $r_n = \frac{\beta_n - \alpha_n}{2}$ . It is readily seen that the function  $g$  is  $C^k$  computable,  $g(x) \geq 0$ , and  $g(x) = 0$  if and only if  $x \in A$ . We are now ready to define the desired function  $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ :  $f(x_1, x_2) = (f_1(x_1, x_2), f_2(x_1, x_2))$ , where

$$f_1(x_1, x_2) = -x_2 + x_1(x_1^2 + x_2^2) \cdot g(x_1^2 + x_2^2)$$

and

$$f_2(x_1, x_2) = x_1 + x_2(x_1^2 + x_2^2) \cdot g(x_1^2 + x_2^2).$$

It is clear that  $f$  is a  $C^k$  computable function.

For the following system

$$\dot{x}_1 = f_1(x_1, x_2), \quad \dot{x}_2 = f_2(x_1, x_2)$$

it can be rewritten, in polar coordinates, in the form of

$$\dot{r} = r^3 g(r^2), \quad \dot{\theta} = 1$$

for  $r > 0$  with  $\dot{r} = 0$  at  $r = 0$ . It is clear that the system has a unique equilibrium point at the origin of  $\mathbb{R}^2$ . Since  $g(x) \geq 0$  for all  $x \in \mathbb{R}$ , it follows that the only periodic orbits are circles with center at the origin and radius  $r$  satisfying  $r > 0$  and  $g(r^2) = 0$ .

Consider one such circle  $\Gamma_0$  with center at the origin and radius  $r_0$ . Then it follows from the construction of  $g$  that  $r_0^2$  is a non-computable real. For any point  $(x_1, x_2)$  on  $\Gamma_0$ , if the point is computable, then  $x_1$  and  $x_2$  must be computable reals, which implies that  $r_0^2 = x_1^2 + x_2^2$  is a computable number. This is a contradiction. Hence none of the points on  $\Gamma_0$  is computable, which implies that  $\Gamma_0$  cannot be r.e. Next we show that  $\Gamma_0$  is not co-r.e. either. Suppose otherwise  $\Gamma_0$  was co-r.e. in  $\mathbb{R}^2$ . Since the intersection of two co-r.e. closed sets is again co-r.e., the set  $\Gamma_0 \cap \{(x, 0) : x \geq 0\} = \{(r_0, 0)\}$  is co-r.e.. Then it follows from Theorem 6.2.9 [Wei00] that there is a computable function  $\gamma, \gamma : \mathbb{R}^2 \rightarrow \mathbb{R}$ , such that  $\{(r_0, 0)\} = \gamma^{-1}\{0\}$ . Since  $(r_0, 0)$  is the unique zero of  $\gamma$ ,  $(r_0, 0)$  is a computable point (Corollary 6.3.9 [Wei00]). This contradicts the fact that  $r_0^2$  is non-computable. Hence  $\Gamma_0$  is not co-r.e.. This completes the proof of Theorem 1.

We mention in passing that the set  $\mathcal{P}$  of all periodic orbits of the planar system defined above is co-r.e. closed because it is the set  $G^{-1}(0)$ , where  $G : \mathbb{R}^2 \rightarrow \mathbb{R}, G(x_1, x_2) = g(x_1^2 + x_2^2)$ , is a computable function. Hence, it is possible to plot over-adumbrations of  $\mathcal{P}$  with better and better accuracies (although the accuracies are unknown per se since  $\mathcal{P}$  is not computable). On the other hand, the relative positions of the periodic orbits are completely in dark - there is no good over- or under-adumbration of any periodic orbit.

## 4 Main ideas of the proof of Theorem 2

One apparent difficulty to plot periodic orbits of the flow in Theorem 1 is that there are too many of them. Can we plot the periodic orbits of a flow if there are only finitely many of them? While the problem is open for  $C^1$  computable planar flows in general, the answer is yes if the planar flow is structurally stable. The structural stability of a planar flow is characterized in terms of its limit set by Peixoto in 1962 in his seminal paper [Pei62]. Let  $f$  be a  $C^1$  vector field defined on a compact two-dimensional differentiable manifold  $K \subseteq \mathbb{R}^2$ . Peixoto showed that if  $f$  is structurally stable on  $K$ , then, among other characterizations, the number of equilibria (i.e. zeros of  $f$ ) and of periodic orbits

is finite and each is hyperbolic, and there is no trajectories connecting saddle points. Similar results hold if  $K$  is a manifold with boundary; in particular,  $K = \mathbb{D} = \{x \in \mathbb{R}^2 : \|x\| \leq 1\}$  with the assumption that the vector fields always point inwards along the boundary of  $\mathbb{D}$ .

We now outline the proof of Theorem 2, which shows that there is an (uniform) algorithm that locates the positions of equilibrium points and periodic orbits with arbitrarily high precision for any structurally stable  $C^1$  planar vector field defined on  $\mathbb{D}$ .

**Remark.** The algorithm is a “master” program in the sense that it computes all equilibria and periodic orbits simultaneously when given a structurally stable planar vector field  $f$ .

**Main ideas of the proof.** The proof is long and intricate; only a brief outline is presented. The complete proof can be found in the preprints [GZ20], [GZ21].

- (A) Construct a sub-algorithm to locate the equilibrium points: on input  $n \geq 1$  (accuracy) and (a  $C^1$ -name of)  $f$ , the sub-algorithm outputs a union of mutually disjoint squares such that each square contains exactly one equilibrium and the side-length of a square is at most  $1/n$ . The construction relies on the fact that there are only finitely many equilibria and each is hyperbolic. The hyperbolicity ensures that the Jacobean of  $f$  at each equilibrium is non-zero. By computing  $f$  and  $Df$  simultaneously on refined square-grids of  $\mathbb{D}$ , the algorithm will return a desired output after finitely many updates on the square-grids.

More specifically, if  $s$  is some square, its corners will have rational coordinates, which ensures all squares are computable. Hence, both  $f(s)$  and  $Df(s)$  are computable from any given  $C^1$ -name of  $f$ . If  $s$  does not contain any zero of  $f$ , then  $0 \notin f(s)$  (precisely, it should be  $(0,0) \notin f(s)$ ; 0 is used to denote the origin of either  $\mathbb{R}$  or  $\mathbb{R}^2$ ). Consequently, the distance between 0 and  $f(s)$ ,  $d(0, f(s)) = \min_{x \in f(s)} d(0, x)$ , is greater than 0. Since  $f(s)$  is computable (from  $f$ ), an over-approximation  $A_l(s)$  (a polygon with rational corners) of  $f(s)$  can be computed with accuracy bounded by  $2^{-l}$  for some  $l \geq n$ . If  $0 \notin f(s)$ , then  $0 \notin A_l(s)$  for  $l$  sufficiently large. Hence, by testing if  $0 \notin A_l(s)$  for all squares  $s$  and  $l = n, n+1, n+2, \dots$ , the algorithm can eventually identify the squares which do not have zeros after finitely many updates on  $l$ . The problematic squares are those containing zeros, because whether  $d(0, f(s)) = 0$  cannot in general be decided by finitely many approximations - one may not be able to conclude either  $0 \in f(s)$  or  $0 \notin f(s)$  with (any) current choice of  $l$ . To deal with this problem, one makes use of the hypothesis that all zeros are hyperbolic; hence, the Jacobean at each zero is invertible. In other words,  $f$  is locally invertible at each of its zeros. This is why the algorithm computes both

$f(s)$  and  $Df(s)$ . After finitely many updates on  $l$ , the algorithm arrives at a midway stage that either  $d(0, f(s)) > 2^{-l}$  (i.e.  $s$  contains no zero of  $f$ ) or  $\|Df(s)\| > 2^{-m}$  for some  $m \geq l \geq n$ . For the latter case, the algorithm computes - based on an effective version of the inverse function theorem - a polygon (under-)approximation of  $f(s)$  as the domain of  $f^{-1}$ . Hence, if 0 is in this (under-)approximation of  $f(s)$ , then  $s$  contains a zero of  $f$ . A possible problem is that  $s$  might have a zero whose image is outside this (under-)approximation of  $f(s)$ . This problem can be solved by covering  $s$  with several overlapping smaller squares and then applying the procedure to all those overlapping smaller squares. By proceeding in this way and by increasing the accuracy  $l \geq n$  used in the computations, the algorithm will be able to determine whether or not each square has a zero after finitely many updates on  $l$ . If a square contains a zero, the zero is unique because on this square  $f$  is invertible. (See [GZ20] for more details).

- (B) Construct a sub-algorithm to locate the periodic orbits: It suffices to construct an algorithm that takes as input  $(n; f)$  and returns a union  $A_n$  of compact subsets, each has polygonal boundaries, such that the Hausdorff distance between  $A_n$  and  $\mathbf{P}$  - the set of periodic orbits - is less than  $1/n$  for every  $n \in \mathbb{N}$  and every structurally stable vector field  $f$ .

Before constructing this algorithm, it is important to remark that, by Peixoto's theorem (see [Pei62]), the limit set of (1) is formed by hyperbolic equilibrium points and hyperbolic periodic orbits. Hence, each periodic orbit is either attracting or repelling. The hyperbolicity condition ensures (see e.g. [Per01]) that, for each periodic orbit  $\gamma$ , there is an open set  $U_\gamma$  (a so-called *basin of attraction*) containing  $\gamma$  such that, if the periodic orbit is attracting, then any trajectory entering  $U_\gamma$  will converge to  $\gamma$  exponentially fast as  $t \rightarrow \infty$  (a similar condition holds for repelling periodic orbits when  $t \rightarrow -\infty$ ). Furthermore, if the system (1) has no saddle points, then there exists some time  $T_\varepsilon \geq 0$  such that any trajectory starting at a point  $x \in \mathbb{D}$  at least  $\varepsilon$ -distance away from equilibrium point(s) or repelling periodic orbit(s) will be inside the basin of attraction of some attracting periodic orbit  $\gamma$  after time  $T_\varepsilon$ , i.e.  $\phi_t(x) \in U_\gamma$  for all  $t \geq T_\varepsilon$ . Hence, by computing  $\phi_t(\mathbb{D})$  for increasing (decreasing) values of time, one is able to approach the set of attracting (repelling) periodic orbits as  $t \rightarrow \infty$  ( $t \rightarrow -\infty$ ), as long as the neighborhoods of equilibrium point(s) are avoided (those neighborhoods - squares each containing a unique equilibrium point - have already been identified in step (A)). This can be done with the following steps:

- (1) Cover the compact set  $\mathbb{D}$  with a finite number of square "pixels;"
- (2) Use a rigorous numerical method to compute the (flow) images of all

pixels after some time  $T$ , and take the union  $A_{n,T}$  of the images of all pixels as a first-round candidate for an approximation to  $\mathbf{P}$ . Then pick sets of pixels from  $A_{n,T}$  and test whether they are forward or backward time invariant, essentially by testing whether  $\phi_t(A) \subseteq A$  for positive or negative  $t$ . In this manner, a set  $B_{n,T} \subseteq A_{n,T}$  is obtained as a second-round candidate for an (over-) approximation to  $\mathbf{P}$ .

For simplicity and consistency of the algorithm sketched here,  $A_{n,T}$  will be used once again to denote  $B_{n,T}$ . The next step is to see whether  $A_{n,T}$  is a “good enough” approximation of  $\mathbf{P}$ .

- (3) Test whether  $A_{n,T}$  is an over-approximation of  $\mathbf{P}$  within the desired accuracy. If the test is successful, set  $A_n = A_{n,T}$  and output  $A_n$ . If the test fails, increase time  $T$  and use a finer lattice of square pixels when numerically approximating the flow after  $T$ . Similar simulations using time  $-T$  are run in parallel to find repellers.

Recall that a periodic orbit  $\gamma$  will separate  $\mathbb{D}$  into the interior and the exterior according to the Jordan curve theorem. The same can be said to a “good enough” approximation of  $\gamma$ . Hence,  $A_{n,T}$  can be separated into connected components; each of the components will have a “doughnut shape.” If one is able to identify the interior and exterior of each doughnut, then one can determine the maximum width of each doughnut. The maximum widths provide an upper bound on the error occurred when using  $A_{n,T}$  to approximate  $\mathbf{P}$ . To identify the interior and exterior regions delimited by a connected component of  $A_{n,T}$ , a coloring algorithm is constructed, which works along the following lines. All pixels considered below are disjoint from the component of  $A_{n,T}$ : (i) pick some pixel and paint this pixel blue; (ii) paint pixels adjacent to blue pixels with the color blue; (iii) when there are no more unpainted pixels which can be painted blue, paint one of the remaining pixels red; (iv) paint unpainted pixels adjacent to red pixels with the color red; (iv) if there are still unpainted pixels, restart the algorithm with a better accuracy (the connected component under consideration has not yet had a doughnut shape). After the successful termination of this (sub-)algorithm, the interior and exterior regions of the considered connected component will correspond to regions of different colors. It can be shown that the main algorithm will eventually halt and that when it does halt, it provides a correct result.

The intricate components of the algorithm are where the saddle points are dealt with, and the search for a time  $T$  such that the Hausdorff distance between  $A_{n,T}$  and  $\mathbf{P}$  is less than  $1/n$  with  $(n; f)$  being the input to the

algorithm. The problem with a saddle point is that it may take an arbitrarily long time for the flow starting at some point near but not on the stable manifold of the saddle to eventually move away from the saddle. This undesirable behavior is dealt with by transforming the original flow near a saddle to a linear flow using a computable version of Hartman-Grobman’s theorem ([GZD12]). The time needed for the linear flow to go through a small neighborhood can be explicitly calculated. (See [GZ21] for details)

We conclude this note with two open questions, which are suggested to us by one of the referees.

- Does there exist a computable function  $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  such that Theorem 1 remains true, where  $f$  is either an analytic function or a polynomial?
- It is known that the structurally stable systems defined on  $\mathbb{D}$  form an open dense subset of  $C^1(\mathbb{D})$ . Is this open subset computable? In other words, is it decidable whether a planar system defined on  $\mathbb{D}$  is structurally stable?

We remark that the proof of Theorem 1 relies on bump functions. A bump function is usually not analytic at the “foot” of the bump. For a polynomial planar system, it is well-known that such a system can only have finitely many limit cycles [Ily91]. Thus, for polynomial planar systems, the question is: Can the finiteness ensure the computability? Concerning the second question, we note that the open subset of all structurally stable systems defined on  $\mathbb{D}$  can be shown to be r.e. open in  $C^1(\mathbb{D})$ .

## Acknowledgments

D. Graça was partially funded by FCT/MCTES through national funds and co-funded EU funds under the project UIDB/50008/2020.



This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 731143.

## References

- [BHW08] V. Brattka, P. Hertling, and K. Weihrauch. A tutorial on computable analysis. In S. B. Cooper, , B. Löwe, and A. Sorbi, editors, *New Computational Paradigms: Changing Conceptions of What is Computable*, pages 425–491. Springer, 2008.
- [BY06] M. Braverman and M. Yampolsky. Non-computable Julia sets. *Journal of the American Mathematical Society*, 19(3):551–578, 2006.

- [GZ20] Daniel S. Graça and Ning Zhong. The set of hyperbolic equilibria and of invertible zeros on the unit ball is computable, 2020. Submitted for publication, available at <https://arxiv.org/abs/2002.08199>.
- [GZ21] Daniel S. Graça and Ning Zhong. Computing the exact number of periodic orbits for planar flows, 2021. Submitted for publication, available at <http://arxiv.org/abs/2101.07701>.
- [GZB12] D. S. Graça, N. Zhong, and J. Buescu. Computability, noncomputability, and hyperbolic systems. *Applied Mathematics and Computation*, 219(6):3039–3054, 2012.
- [GZD12] D. S. Graça, N. Zhong, and H. S. Dumas. The connection between computability of a nonlinear problem and its linearization: the Hartman-Grobman theorem revisited. *Theoretical Computer Science*, 457(26):101–110, 2012.
- [HSD04] M. W. Hirsch, S. Smale, and R. Devaney. *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Academic Press, 2004.
- [Ily91] Y. Ilyashenko. *Finiteness Theorems for Limit Cycles*, volume 84 of *Translations of Mathematical Monographs*. American Mathematical Society, 1991.
- [Ko91] K.-I Ko. *Complexity Theory of Real Functions*. Birkhäuser, 1991.
- [Pei62] M. Peixoto. Structural stability on two-dimensional manifolds. *Topology*, 1:101–121, 1962.
- [Per01] L. Perko. *Differential Equations and Dynamical Systems*. Springer, 3rd edition, 2001.