An Interval Approach to Compute Invariant Sets

Thomas Le Mézo, Luc Jaulin, Benoit Zerr

An interval approach to compute invariant sets

Thomas Le Mézo, Luc Jaulin and Benoît Zerr

***Index Terms*—Guaranteed integration, interval analysis, invariant set, limit cycle, constraint programming, state equation.**

***Abstract*—This paper proposes an original interval-based method to compute an outer approximation of all invariant sets (such as limit cycles) of a continuous-time non-linear dynamic system which are included inside a prior set of the state space. Contrary to all other existing approaches, our method has the following properties: (i) it is guaranteed (a solution cannot be lost), (ii) it is applicable to a large class of systems without any specific assumption such as the knowledge of a Lyapunov function or any partial linearity, and (iii) there is no need to integrate the system.**

I. Introduction

The computation of invariant sets of dynamical systems has been the subject of extensive research over the past several decades [1]. It has many applications such as verifying the safety of unmanned flight [2]. Most methods that have been proposed for computing invariant sets assume that the system is linear, with the possibility of state jumps [3]–[5] or bounded disturbances [6], [7] but this is not always realistic.

In this paper, we consider the problem of characterizing invariant sets in the case where the system is nonlinear and the time is continuous (see, *e.g.*, [8] when the time is discrete). It is only assumed that the system is described by a state equation of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ where $\mathbf{x}$ is the state vector and $\mathbf{f}$ is continuous. Since we want the method to be general, we will not consider symbolic methods such as in [9], [10], but numerical methods instead. Moreover, since we want the characterization to be guaranteed, we will consider a method based on interval analysis. This choice is motivated, in the context of the analysis of dynamical system stability, by the fact that interval analysis has already been successfully used by the mathematician W. Tucker to solve the 14th problem of Smale which aims to prove that the Lorenz attractor is strange [11]. The approach used by Tucker has also been extended to other specific dynamical systems (see *e.g*, [12]), nevertheless a general method has not been proposed to deal with the general case. Most of the interval methods, which deal with the characterization of invariant sets [13], use the concept of *guaranteed integration* [14]. From an initial box $[\mathbf{x}](0)$, guaranteed integration [15] provides a set of techniques, based on interval arithmetic, to compute a box-valued function $[\mathbf{x}](t)$ (or *tube*) which contains all the true values verifying the initial value problem. These techniques are based on an interval counterparts of Euler [16], Runge-Kutta [17] or Taylor [18] integration and they validate their result using the Picard Theorem.

In the context of the analysis of nonlinear systems, guaranteed integration is mainly used for characterizing reachable sets [19]–[21] or for computing the viability kernel [22]. However, these methods cannot be considered as efficient to characterize invariant sets for the two following reasons: *(i)* the interval integration methods are too conservative and *(ii)* a fusion of a collection of tubes (obtained after an interval integration), in order to extract an invariant set, is difficult to obtain.

Still with regard to stability analysis, there exists another class of methods that do not require any integration. These methods are based on Lyapunov theory and approximate the dynamic of the system by a function [23], [24] that is supposed to decrease with time. From this function, they derive a set of inequalities defining a set which is invariant [25], [26] with respect to the dynamic of the system [27]. The resolution can then be performed using interval analysis [8], [28] or by other symbolic methods. The main drawback of the Lyapunov-based methods is that they require the knowledge of a Lyapunov function which is the case only for a small class of systems.

In this paper we propose an original and generic interval method to compute the largest invariant set $\text{Inv}(\mathbb{X})$ of a subset $\mathbb{X}$ of the state space. Our method does not require any interval integration and does not assume the knowledge of a Lyapunov function. The method performs a discretization of the state space by a graph. Contrary to all other existing approach, it represents the invariant set $\text{Inv}(\mathbb{X})$ as a set of trajectories that are inside $\mathbb{X}$ for all t. A projection of the infinite dimensional set of trajectories onto the state space is then performed to get $\text{Inv}(\mathbb{X})$. Our method is based on *constraint programming* [29] and thus follows the following principle: instead of focusing on the solution, we eliminate all parts of the research space that are inconsistent.

The paper is organized as follows. Section II proposes a formalization of the problem. Section III shows that our problem can be expressed as a constraint network and recalls the principle of the constraint propagation that will be used for the resolution. Section IV shows how trajectories can be enclosed by a set of path associated with both a paving and a collection of subpaths. The new concept of *road,* which corresponds to a set of subpath of $\mathbb{R}^n$, is introduced. The corresponding contractors able to eliminate unfeasible subpaths are introduced in Section V. Section VI presents the main algorithm and Section VII illustrates the principle of the method on a classical test case. Section VIII concludes the paper.

In this paper, we use the notations given by Table I.

Manuscript received February 18, 2016; accepted March 7, 2017. This work has been supported by the French Government Defense procurement and technology agency (DGA).

The authors are with the Perception, Robotics, Autonomous SYStems (PRASYS) Group, ENSTA-Bretagne, Lab-STICC, 2 rue François Verny, 29806 Brest, France (email: thomas.le_mezo@ensta-bretagne.org; luc.jaulin@ensta-bretagne.fr; benoit.zerr@ensta-bretagne.fr)

Subsets of $\mathbb{R}^n$:	$\mathbb{A}, \mathbb{B}$
Intervals of $\mathbb{R}$:	$[a]$
Boxes of $\mathbb{R}^n$:	$[\mathbf{a}]$
Trajectory in $\mathbb{R}^n$	$\mathbf{x}(\cdot)$
Set of boxes of $\mathbb{R}^n$:	$\mathbb{IR}^n$
Set of subpaths:	$\mathbb{SP}$
Path box:	$\langle [\mathbf{a}], [\mathbf{b}], [\mathbf{x}] \rangle$

TABLE I: Notations

II. Formalization of the problem

This section defines the notion of path and invariant set for a continuous-time system and formalizes the problem that we want to solve.

A *trajectory* is a smooth function $\mathbf{x}(\cdot)$ from $\mathbb{R}$ to $\mathbb{R}^n$. The *path* associated with a trajectory $\mathbf{x}(\cdot)$ is the set of: all $\mathbf{x}(t) \in \mathbb{R}^n$ and an orientation with respect to t [30] .

Example. Consider the trajectory

$$\mathbf{x}(\cdot) : \begin{cases} \mathbb{R} & \rightarrow & \mathbb{R}^2 \\ t & \mapsto & \begin{pmatrix} cos(t) \\ sin(t) \end{pmatrix} \end{cases} \tag{1}$$

The corresponding non-oriented path is a circle. The oriented path associated with $\mathbf{x}(\cdot)$ is a circle with the direct trigonometric orientation (see Figure 1).

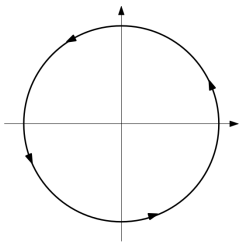

Fig. 1: A directed path corresponding to a circle

Consider the state equation of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^n$. An oriented path is *feasible* if it is associated with a trajectory $\mathbf{x}(t)$ which is a solution of (2). Note that such a path can be cyclic or can even be a single point. A path cannot make any loop (*i.e.*, it cannot cross itself) or even cross another path [9]. The problem that we consider in this paper is the following.

Problem. Given a compact set $\mathbb{X}$ of $\mathbb{R}^n$, find the set Inv($\mathbb{X}$) which encloses all points $\mathbf{x}_0$ which belong to a feasible path of (2) strictly included in $\mathbb{X}$.

Link with invariant sets. A set $\mathbb{A}$ is *positive invariant* [1] if for any trajectory $\mathbf{x}(\cdot)$ of (2), we have

$$\mathbf{x}(0) \in \mathbb{A}, t \geq 0 \implies \mathbf{x}(t) \in \mathbb{A}, \tag{3}$$

and it is *negative invariant* if for any trajectory $\mathbf{x}(\cdot)$ of (2), we have

$$\mathbf{x}(0) \in \mathbb{A}, t \leq 0 \implies \mathbf{x}(t) \in \mathbb{A}. \tag{4}$$

The set $\mathbb{A}$ is *invariant* if it is both positive and negative invariant. The set of invariant sets is a sub-lattice of the set of all subsets of $\mathbb{R}^n$, denoted by $\mathcal{P}(\mathbb{R}^n)$. This means that if $\mathbb{A}$ and $\mathbb{B}$ are invariant, then $\mathbb{A} \cap \mathbb{B}$ and $\mathbb{A} \cup \mathbb{B}$ are also invariant. As a consequence, for a given set $\mathbb{X} \in \mathcal{P}(\mathbb{R}^n)$, there exists a (unique) largest invariant set, which corresponds to Inv($\mathbb{X}$), included in $\mathbb{X}$. The set Inv($\mathbb{X}$) corresponds to the union of all invariant subsets of $\mathbb{X}$.

III. Constraint propagation

We say that a path is *valid* if it is included inside $\mathbb{X}$. Otherwise it is *dead*. A state $\mathbf{x}$ which belongs to a valid (resp. dead) path is inside (resp. outside) Inv($\mathbb{X}$). Figure 2 shows a valid path (red) which is here a limit cycle. Two dead paths (blue) have at least one point outside $\mathbb{X}$. All paths are feasible, since they are consistent with (2). The path (ii) is also a cycle. The path (iii) is closed to an unstable limit cycle.

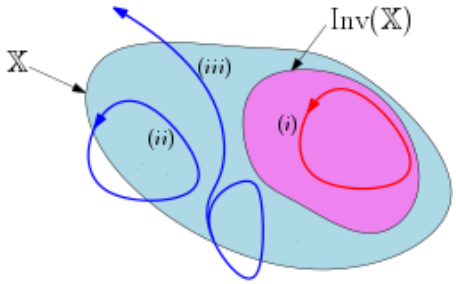

Fig. 2: The path (i) is valid since it is included inside $\mathbb{X}$. Both path (ii) and path (iii) are dead.

Computing Inv($\mathbb{X}$) amounts to characterizing the set of all valid paths. The approach, we propose to follow, is to formulate the problem of characterizing Inv($\mathbb{X}$) as a constraint network and then to apply a contractor technique for the resolution. We now recall briefly the notion of a constraint network and contractors.

Constraint network. A *constraint network* [31] $\mathcal{H}$ is composed of a set of variables $\mathcal{V} = \{x_1, \ldots, x_n\}$, a set of constraints $\mathcal{C} = \{c_1, \ldots, c_m\}$ and a set of abstract domains [32] $\{\mathbb{X}_1, \ldots, \mathbb{X}_n\}$ containing the x_i's. The values for variables x_i can be symbols, real numbers [33], vectors of $\mathbb{R}^n$, and sometimes trajectories (see [34]). The constraints can be equations between the variables (such as $x_3 = x_1 + e^{x_2}$) or differential equations (such as $\dot{x}_3 \cdot \ddot{x}_1 + e^{x_2} = 0$) and the domains can be intervals, boxes [35], zonotopes [36] or tubes [37].

Constraint propagation. The goal of propagation techniques is to find the solution of a constraint network. The principle is to contract as much as possible the domains for the variables without loosing any solutions. The corresponding operator is called a *contractor*. For the resolution of a constraint network, we take all constraints and call the corresponding contractors until no more contraction can be observed. Therefore, constraint propagation is only able to find an outer approximation of the problem.

Invariant sets as a constraint network. For our problem, to apply a constraint approach, we need to define the constraint network $\mathcal{H}$. The variables are the paths of $\mathbb{R}^n$ which are consistent with $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$. The unique constraint is the following: "the path should be included inside $\mathbb{X}$". We have

now defined the variables and the constraints. It remains to define the domains for the paths which is the objective of the next section.

IV. Subpath and roads

The approach we will follow in this paper to compute Inv($\mathbb{X}$) is to characterize all paths that are totally inside $\mathbb{X}$, from $t = -\infty$ to $t = \infty$. We now define the notion of subpath that will be used for the decomposition of the path we want to approximate. A path of (2) can be covered by a union of non overlapping subpaths. More formally, a subpath of (2) can be defined as follows.

Subpath. Consider a trajectory $\mathbf{x}(\cdot)$ of (2) and an interval $[t_a, t_b]$. The corresponding part of the path is called a *subpath* $\overrightarrow{P}$. If $\overrightarrow{P}$ is non-cyclic, we can define the left endpoint $left(\overrightarrow{P})$ and the right endpoint $right(\overrightarrow{P})$ as the two points $\mathbf{x}(t_a)$ and $\mathbf{x}(t_b)$. The set of non-cyclic subpaths is denoted by $\mathbb{SP}$.

Paving. A paving $\mathcal{Q}$ is a union of non overlapping (*i.e.*, their interiors are disjoint) boxes covering $\mathbb{X}$, which is assumed to be a compact set of $\mathbb{R}^n$.

Decomposition. Given a paving $\mathcal{Q}$. A path can be decomposed into subpaths $\overrightarrow{P}(k) \in \mathbb{SP}$ (see Figure 3) such that

$$\exists\, [\mathbf{x}] \in \mathcal{Q} \mid \begin{cases} \overrightarrow{P}(k) \subset [\mathbf{x}] \\ left(\overrightarrow{P}(k)) \in \partial\, [\mathbf{x}] \\ right(\overrightarrow{P}(k)) \in \partial\, [\mathbf{x}] \end{cases} \tag{5}$$

where $\partial\, [\mathbf{x}]$ corresponds to the boundary of the box $[\mathbf{x}]$. Note that a box of $\mathcal{Q}$ may contain several subpaths. Figure 3 shows a paving made with 5 boxes and a cyclic path which can be composed into 7 subpaths.

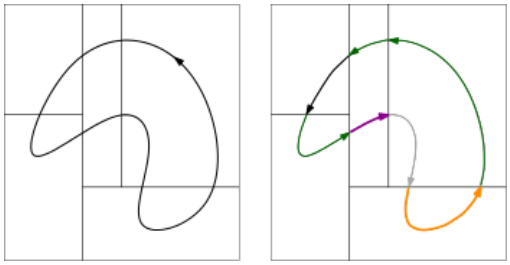

Fig. 3: Left: one path and one paving $\mathcal{Q}$, Right: decomposition of the path into 7 subpaths

To approximate the solution paths using contractor-based techniques [33], [38], we need a paving which provides a decomposition into subpaths and a representation for the domains of the subpaths (*i.e.*, a way to represent a set of subpath). A *domain* for a subpath is a set of subpaths that aims to be contracted by operators (the *contractors*). These operators are allowed to remove subpaths that are inconsistent with the state equation (2). The domains that will be chosen for the subpaths are called *roads* and are presented for the first time in this section. Before introducing the concept of *road*, we first need to define *pathboxes*, which will later be interpreted as a specific road.

Pathbox. A *pathbox* $\langle [\mathbf{a}], [\mathbf{b}], [\mathbf{x}] \rangle$, where $[\mathbf{a}] \subset [\mathbf{x}]$ and $[\mathbf{b}] \subset [\mathbf{x}]$ is a set of non-cyclic subpaths defined as:

$$\langle [\mathbf{a}], [\mathbf{b}], [\mathbf{x}] \rangle \;=\; \{ \quad \begin{array}{l} \overrightarrow{P} \in \mathbb{SP}, left(\overrightarrow{P}) \in [\mathbf{a}], \\ right(\overrightarrow{P}) \in [\mathbf{b}], \overrightarrow{P} \subset [\mathbf{x}] \end{array} \quad \}. \tag{6}$$

As illustrated by Figure 4, a pathbox can be interpreted as an *abstract domain* [39], classically used in static analysis [32] to validate computer programs. Such a domain can be contracted [40] with respect to existing constraints. Here, the constraint is that the subpath should be consistent with the state equation (2). Next section theorems will provide the conditions that will allow us to build a contractor for the pathbox.

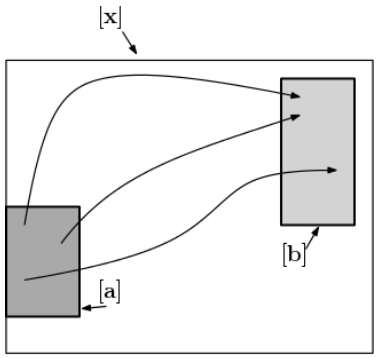

Fig. 4: The three subpaths belong to the pathbox $\langle [\mathbf{a}], [\mathbf{b}], [\mathbf{x}] \rangle$

Road. In the definition of the pathbox $\langle [\mathbf{a}], [\mathbf{b}], [\mathbf{x}] \rangle$, the domains that have been taken for the left and right endpoints of the subpath are the boxes $[\mathbf{a}]$ and $[\mathbf{b}]$. Now, we could have taken other types of domains such as a tiling, *i.e.*, a union of boxes. A *road* is a finite union of pathboxes in the same box $[\mathbf{x}]$:

$$\langle \{[\mathbf{a}](i)\}_{i\geq 1}, \{[\mathbf{b}](j)\}_{j\geq 1}, [\mathbf{x}] \rangle = \bigcup_{i,j} \langle [\mathbf{a}](i), [\mathbf{b}](j), [\mathbf{x}] \rangle\,.$$

An illustration of a road is given on Figure 5.

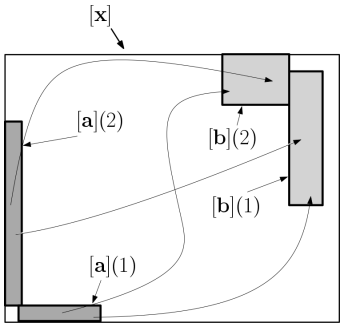

Fig. 5: A road with 4 feasible subpaths

V. Contractors

This section introduces two contractors for the roads that will be called during the resolution. The *dynamic contractor* removes the subpath that are not consistent with the state equation and the *continuity contractor* removes subpaths that cannot be enclosed inside the set $\mathbb{X}$. We now give two new theorems that will be used by the dynamic contractor.

Notation. We denote by $[\mathbf{f}]([\mathbf{x}])$ an inclusion function for $\mathbf{f}([\mathbf{x}])$ [16], *i.e.*, given a box $[\mathbf{x}]$, $[\mathbf{f}]([\mathbf{x}])$ is a box which encloses the range of $\mathbf{f}$ over $[\mathbf{x}]$. More precisely, we have

$$\mathbf{f}([\mathbf{x}]) = \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in [\mathbf{x}]\} \subset [\mathbf{f}]([\mathbf{x}]). \tag{7}$$

Accurate inclusion functions can be easily computed using interval arithmetic for a huge class of functions $\mathbf{f}$.

Theorem 1. *Consider a subpath $\overrightarrow{P}$ of $\langle[\mathbf{a}],[\mathbf{b}],[\mathbf{x}]\rangle$ consistent with (2). Define $\mathbf{a}^* = left(\overrightarrow{P})$ and $\mathbf{b}^* = right(\overrightarrow{P})$. Then, $\exists \beta > 0$ such that*

$$\mathbf{b}^* - \mathbf{a}^* \in \beta \cdot [\mathbf{f}]([\mathbf{x}]). \tag{8}$$

Proof: Since $\overrightarrow{P} \in \langle[\mathbf{a}],[\mathbf{b}],[\mathbf{x}]\rangle$ is consistent with (2), there exists one trajectory $\mathbf{z}(t)$ corresponding to $\overrightarrow{P}$ such that $\mathbf{z}(0) = \mathbf{a}^*$ and such that $\mathbf{z}(\beta) = \mathbf{b}^*$ where $\beta > 0$. From the generalized mean value theorem[1] [41] applied on $\mathbf{z}(t)$ for $[0,\beta]$, we have

$$\mathbf{b}^* \in \mathbf{z}(0) + \left[\frac{d\mathbf{z}}{dt}([0,\beta])\right] \cdot \beta. \tag{9}$$

where $\left[\frac{d\mathbf{z}}{dt}([0,\beta])\right]$ is a box which encloses the set $\{\frac{d\mathbf{z}}{dt}(t), t \in [0,\beta]\}$. Since $\mathbf{z}(0) = \mathbf{a}$ and $\left[\frac{d\mathbf{z}}{dt}([0,\beta])\right] \subset [\mathbf{f}]([\mathbf{x}])$ we have

$$\mathbf{b}^* - \mathbf{a}^* \in [\mathbf{f}]([\mathbf{x}]) \cdot \beta. \tag{10}$$

■

We can use this constraint to contract $[\mathbf{a}]$ and $[\mathbf{b}]$ without removing any consistent path. This contraction is illustrated by Figure 6. The cone on the top-left of the subfigures represents the feasible directions for $\mathbf{f}(\mathbf{x})$. If we inflate $[\mathbf{x}]$, this cone will also be inflated. The top part of $[\mathbf{b}]$ is contracted because we cannot reach this part from $[\mathbf{a}]$ following the directions imposed by $[\mathbf{f}]([\mathbf{x}])$. In the same manner, the right part of $[\mathbf{a}]$ is contracted because from this right part, we cannot reach $[\mathbf{b}]$ following the allowed directions.

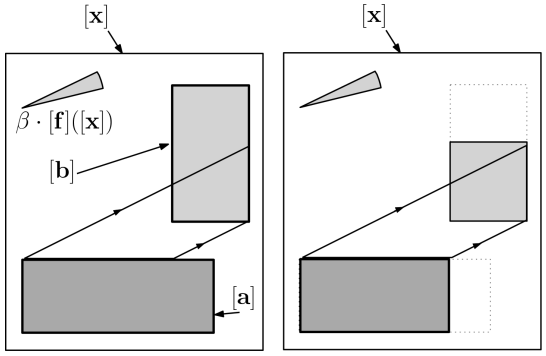

Fig. 6: Contraction of a pathbox. Left: before contraction; Right: after contraction

The following theorem, will allow us to compute a sub-polygon of $[\mathbf{x}]$ which encloses all subpaths of $\langle[\mathbf{a}],[\mathbf{b}],[\mathbf{x}]\rangle$ consistent with (2).

[1]Generalized mean value theorem. If $\mathbf{h} : [t_a, t_b] \subset \mathbb{R} \to \mathbb{R}^n$ is smooth. If $\forall t \in [t_a, t_b]$, $\frac{d\mathbf{h}}{dt}(t) \in [\mathbf{v}] \in \mathbb{IR}^n$ then $\exists \mathbf{v} \in [\mathbf{v}]$ such that $\mathbf{h}(t_b) \in \mathbf{h}(t_a) + \mathbf{v} \cdot (t_b - t_a)$.

Theorem 2. *Given a subpath $\overrightarrow{P}$ of $\langle[\mathbf{a}],[\mathbf{b}],[\mathbf{x}]\rangle$ consistent with (2). For all $\mathbf{s} \in \overrightarrow{P}$, we have*

$$\begin{cases} \exists \beta_a > 0 \mid & \mathbf{s} \in [\mathbf{a}] + \beta_a \cdot [\mathbf{f}]([\mathbf{x}]) \\ \exists \beta_b > 0 \mid & \mathbf{s} \in [\mathbf{b}] - \beta_b \cdot [\mathbf{f}]([\mathbf{x}]) \end{cases} \tag{11}$$

From this theorem, we immediately derive the following corollary.

Corollary 3. *The polygon*

$$\begin{array}{rcl} poly(\langle[\mathbf{a}],[\mathbf{b}],[\mathbf{x}]\rangle) & = & [\mathbf{x}] \\ & \cap & [\mathbf{a}] + [0,\infty] \cdot [\mathbf{f}]([\mathbf{x}]) \\ & \cap & [\mathbf{b}] - [0,\infty] \cdot [\mathbf{f}]([\mathbf{x}]) \end{array} \tag{12}$$

encloses all subpaths $\overrightarrow{P}$ of $\langle[\mathbf{a}],[\mathbf{b}],[\mathbf{x}]\rangle$ consistent with (2).

Remark. This corollary will be used later in the test-case to have a representable approximation of Inv($\mathbb{X}$). Moreover, this polygonal approximation is needed if we have to search for a positive invariant set.

Proof: (of Theorem 2). From the subpath $\overrightarrow{P}$ of $\langle[\mathbf{a}],[\mathbf{b}],[\mathbf{x}]\rangle$, we define the two points $\mathbf{a}^* = left(\overrightarrow{P})$ and $\mathbf{b}^* = right(\overrightarrow{P})$. There exists a trajectory $\mathbf{z}(t)$ corresponding to $\overrightarrow{P}$ such that $\mathbf{z}(0) = \mathbf{a}^*$ and such that $\mathbf{z}(t_b) = \mathbf{b}^*$ where $t_b > 0$. Since $\mathbf{s} \in \overrightarrow{P}$, there exists $\beta_a \in [0, t_b]$ such that $\mathbf{s} = \mathbf{z}(\beta_a)$. From the generalized mean value theorem applied on $\mathbf{z}(t)$ on $[0,\beta_a]$, we have

$$\mathbf{s} = \mathbf{z}(\beta_a) \in \mathbf{z}(0) + \left[\frac{d\mathbf{z}}{dt}([0,\beta_a])\right] \cdot \beta_a. \tag{13}$$

Since $\mathbf{z}(0) = \mathbf{a}$ and $\left[\frac{d\mathbf{z}}{dt}([0,\beta_a])\right] \subset [\mathbf{f}]([\mathbf{x}])$ we have $\mathbf{s} \in [\mathbf{a}] + \beta_a \cdot [\mathbf{f}]([\mathbf{x}])$. The same reasoning starting from $\mathbf{b}$ instead of $\mathbf{a}$ leads to $\mathbf{s} \in [\mathbf{b}] - \beta_b \cdot [\mathbf{f}]([\mathbf{x}])$. ■

Dynamic contractor. To contract a road, it suffices to contract all pathboxes $\langle[\mathbf{a}](i),[\mathbf{b}](j),[\mathbf{x}]\rangle$ of the union. For our implementation (illustrated later in the test-case at Section VII), we have taken $2 \cdot n$ boxes, each of them corresponding to a part of the $2 \cdot n$ faces of the box $[\mathbf{x}]$.

Continuity contractor. Consider one trajectory inside $\mathbb{X}$ from $t = -\infty$ to $t = +\infty$, and a box $[\mathbf{x}]$ of $\mathcal{Q}$ where $\mathcal{Q}$ is a paving covering $\mathbb{X}$. If $\mathbf{0} \notin [\mathbf{f}]([\mathbf{x}])$, then, the trajectory enters inside $[\mathbf{x}]$ at some point $\mathbf{a}$ and leaves $[\mathbf{x}]$ at some point $\mathbf{b}$ within a finite time. This means that no cycle nor equilibrium point can exist inside $[\mathbf{x}]$. Since $\mathcal{Q}$ covers the whole path, when the trajectory enters inside a box $[\mathbf{x}]$, it also leaves another box of $\mathcal{Q}$. In the same manner, when the trajectory leaves $[\mathbf{x}]$, it enters inside another box of $\mathcal{Q}$. Then, we can contract the left endpoint boxes $\{[\mathbf{a}](i)\}_{i\geq 1}$ with respect to all neighbors and all right endpoint boxes $\{[\mathbf{b}](j)\}_{j\geq 1}$. This contraction is illustrated by Figure 7 in the case where the road is a pathbox. The entry subbox $[\mathbf{a}]$ (gray) of $[\mathbf{x}]$ is contracted with respect to the two other neighbor exit boxes (gray and hatched). The exit subbox $[\mathbf{b}]$ (black and hatched) of $[\mathbf{x}]$ is contracted to the flat box with respect to the gray entry box of the neighbor of $[\mathbf{x}]$ below it.

VI. METHOD

In this section, we propose a method able to enclose Inv($\mathbb{X}$). Recall that Inv($\mathbb{X}$) contains all paths corresponding to the

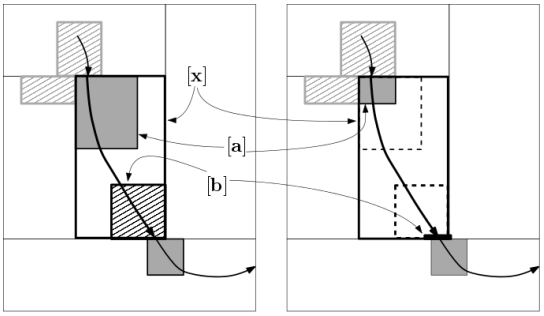

Fig. 7: Contraction of the middle box road; Left: before contraction, Right: after contraction

trajectories that are totally included inside a compact set $\mathbb{X}$ of $\mathbb{R}^n$. The method we propose here follows the idea of Saint-Pierre algorithm [42], *i.e.*, it removes parts of the initial set $\mathbb{X}$ that cannot be inside Inv($\mathbb{X}$) with the objective to converge from outside toward Inv($\mathbb{X}$). The main difference is that the tools we use here (interval analysis and contractors) will allow us to be guaranteed with respect to floating point operations,do not require any knowledge concerning Lipschitz constants and has no parameter to tune. As a consequence, our algorithm is directly implementable, *i.e.*, we are able to provide a friendly solver, named CYCLE, which is able to compute Inv($\mathbb{X}$) for two-dimensional problems. The only inputs for the user are the expression of the evolution function $\mathbf{f}$ and the initial set $\mathbb{X}$. To our knowledge, no other equivalent solver exists.

The principle is to build a paving covering $\mathbb{X}$, made with non-overlapping boxes. Then, we contract each road of the paving using the continuity principle that is now explained.

Algorithm. We propose the following algorithm to compute an enclosure of Inv($\mathbb{X}$):

1) *Set* $k = 0$. *Cover* $\mathbb{X}$ *with a paving* $\mathcal{Q}$. This paving can be for instance a single box.
2) *For each* $[\mathbf{x}]$ of $\mathcal{Q}$, *initialize the road with* $\langle [\mathbf{x}], [\mathbf{x}], [\mathbf{x}] \rangle$. This initialization means that all subpaths in $[\mathbf{x}]$ are considered as feasible, a priori.
3) *For each box* $[\mathbf{x}]$ *of* $\mathcal{Q}$ *such that* $\mathbf{0} \notin [\mathbf{f}]([\mathbf{x}])$, *contract the road with respect to its neighbors*, using the continuity contractor. This procedure is a consequence of the continuity of a path. It takes into account the fact that for a given valid trajectory inside $\mathcal{Q}$, if the path enters inside a box $[\mathbf{x}]$ of $\mathcal{Q}$, it also leaves another box of $\mathcal{Q}$ and if the path leaves $[\mathbf{x}]$, it also enters inside another box of $\mathcal{Q}$.
4) *For each box* $[\mathbf{x}]$ *of* $\mathcal{Q}$, *contract the road* $\langle \{[\mathbf{a}](i)\}_{i\geq 1}, \{[\mathbf{b}](j)\}_{j\geq 1}, [\mathbf{x}] \rangle$. More precisely, the dynamic contractor is used to eliminate subpaths that are not consistent with the state equation.
5) *Bisect all boxes of* $\mathcal{Q}$ *associated with a non-empty road.*
6) *Increase* k *by one and go to Step* 3.

This algorithm only removes unfeasible subpaths. Thus, at each iteration, we can guarantee that the polygons (see Theorem 2) associated with each road computed by the algorithm encloses all corresponding feasible subpaths. The union of these polygons is thus an outer approximation of Inv($\mathbb{X}$) [43]. This can be formalized by the following theorem.

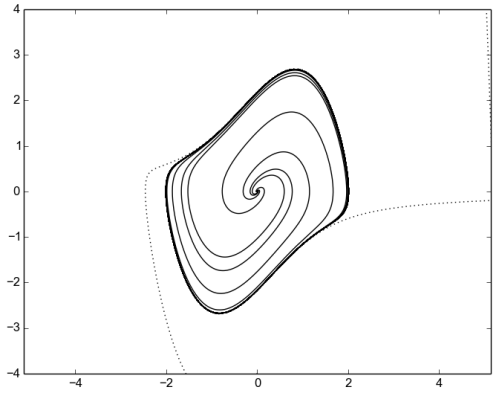

Fig. 8: The dotted trajectories leave the box $\mathbb{X}$ when $t \to -\infty$. The continuous-stroke trajectories are totally inside $\mathbb{X}$

Theorem 4. *At iteration* k, *the set*

$$\mathbb{Q}(k) = \bigcup_{[\mathbf{x}]\in Q(k)} \bigcup_{i\geq 1, j\geq 1} poly(\langle [\mathbf{a}](i), [\mathbf{b}](j), [\mathbf{x}] \rangle)$$

encloses the invariant set Inv($\mathbb{X}$). Moreover, the algorithm always converges when $k \to \infty$.

Proof: We will first prove that $\mathbb{Q}(k)$ encloses Inv($\mathbb{X}$). Consider one point $\mathbf{z} \in$Inv($\mathbb{X}$). There exists one trajectory inside $\mathbb{X}$ going through $\mathbf{z}$ or equivalently, $\mathbf{z}$ belongs to a feasible path $\overrightarrow{P}$ totally included in $\mathbb{X}$. Assume that at iteration k, $\overrightarrow{P} \subset \mathbb{Q}(k)$. From Theorem 2, the dynamic contractor at Step 4 does not remove a single point of $\overrightarrow{P}$. Since $\overrightarrow{P}$ is continuous, the continuity contractor at Step 3 does not remove any point of $\overrightarrow{P}$. Moreover, the bisection procedure does not remove any point of the state space. As a consequence, we have $\overrightarrow{P} \subset \mathbb{Q}(k+1)$. We conclude that for all k, $\mathbf{z} \in \mathbb{Q}(k)$ and thus that Inv($\mathbb{X}$)$\subset \mathbb{Q}(k)$.
The sequence of sets $\mathbb{Q}(k)$ is decreasing with respect to the inclusion, *i.e.*, $\mathbb{Q}(k+1) \subset \mathbb{Q}(k)$. From the Tarski fixed-point theorem [44], the sequence converges to the largest fixed point of our contraction procedure, *i.e.*, the sequence $\mathbb{Q}(k)$ converges to a set made with roads, which cannot be contracted by any of our two contractors. Note that this fixed point may not correspond to Inv($\mathbb{X}$), but it always corresponds to a set which contains it. ■

VII. TEST-CASE

Consider the system described by the Van der Pol equation:

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \left(1 - x_1^2\right) \cdot x_2 - x_1 \end{cases} \tag{14}$$

and the box $\mathbb{X} = [-4, 4] \times [-4, 4]$. Some trajectories that are totally inside $\mathbb{X}$ for all $t \in \mathbb{R}$ are represented by the continuous paths of Figure 8. The dotted trajectories cannot be considered as solutions, since they leave $\mathbb{X}$ when t moves toward $-\infty$.

If we apply our method, and we stop the algorithm after $k = 12$ (less than 4 sec on a single processor i5-2520M@2.50GHz), we get the outer approximation of Inv($\mathbb{X}$) as represented in Figure 9.

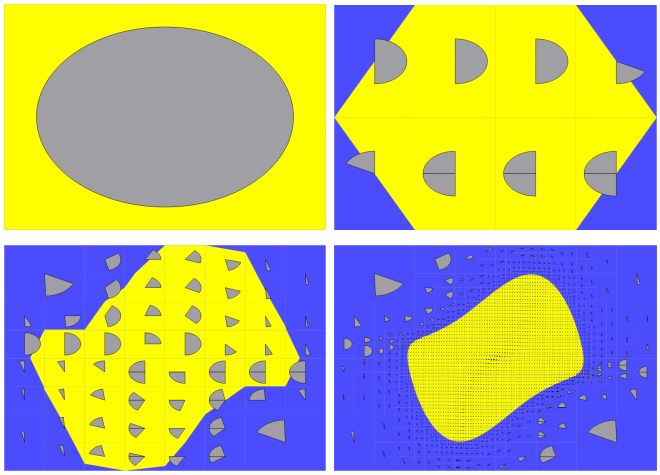

Fig. 9: All paths totally included in the box $\mathbb{X}$ are inside the yellow area (from left to right, top to bottom : $k = \{0, 2, 6, 12\}$. The grey cones corresponds to $\beta \cdot [\mathbf{f}]([\mathbf{x}])$.

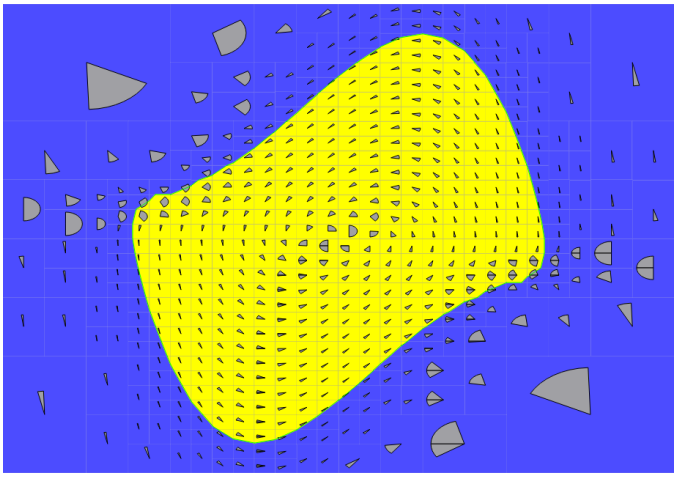

Fig. 10: The first positive invariant set found by the algorithm

The first positive invariant set was found at step 8 by the algorithm and is represented in Figure 10. This was done by checking that on the boundary $\partial\mathbb{Q}(k)$ of the set $\mathbb{Q}(k)$ (the convex hull of the union of all polygons given by Corollary 3, see the yellow area), the evolution cone (painted in gray) is pointing toward the inside of $\mathbb{Q}(k)$.

If we now take $\mathbb{X} = [-4, 4]^{\times 2} \backslash [-0.5, 0.5]^{\times 2}$, our algorithm provides the enclosure of Inv($\mathbb{X}$) depicted on Figure 11. It also corresponds to a guaranteed enclosure of the limit cycle of our system.

The solver CYCLE that generated the figures of this section and some illustrative videos can be found at www.ensta-bretagne.fr/jaulin/cycle.html.

VIII. CONCLUSION

In this paper, we have proposed a new approach to compute invariant sets of continuous-time dynamical systems. More precisely, our method provides a guaranteed outer approximation of the largest invariant set Inv($\mathbb{X}$) included in a given compact set $\mathbb{X}$. The principle of our method is similar to the Saint-Pierre algorithm [42], since it removes part of $\mathbb{X}$ that are outside the solution set Inv($\mathbb{X}$). The main difference is that the tools we use here (interval analysis and contractors): allow us to be guaranteed with respect to floating point operations, does not require any knowledge of any Lipschitz constants and has no parameter to tune. As a consequence, we are able to provide an easy-to-use solver, named CYCLE, that is able to compute Inv($\mathbb{X}$) for two-dimensional problems. The only inputs for the user are the expression of the evolution function $\mathbf{f}$ and the initial set $\mathbb{X}$. To our knowledge, no other equivalent solver exists. The principle of our method has been illustrated on the characterization of the invariant sets and the limit cycle of the Van der Pol system. When the invariant is stable, we have also shown that we were able to compute an approximation which is positive invariant.

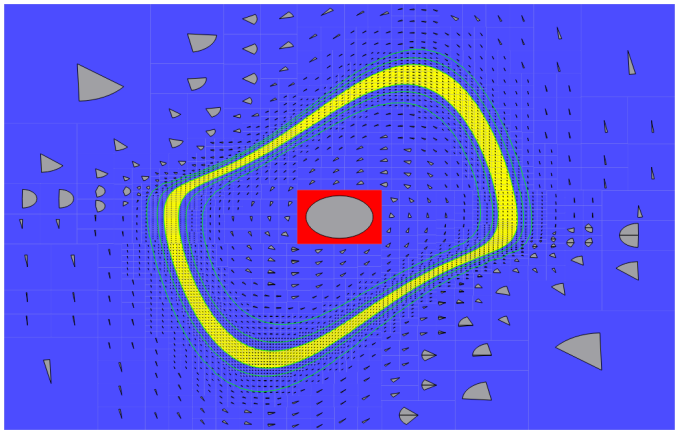

Fig. 11: Guaranteed enclosure of the limit cycle of the Van der Pol system obtained when the red box is removed. The green curves correspond to the positive invariant sets found at different steps of the algorithm.

To solve the problem, we followed a constraint propagation approach. Therefore, instead of focusing on the solution as made by most existing approaches, our method focuses on inconsistent paths. Due to this, we do not need to use the Picard operator nor any interval counterpart of integration method [45] and we do not need to have the knowledge of a Lyapunov function of our system. For the implementation, we had to introduce the notions of pathbox and road as a new type of domains to enclose the valid paths, *i.e.*, the paths that are entirely inside $\mathbb{X}$ and that are consistent with the state equation.

REFERENCES

[1] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Springer Science & Business Media, Oct. 2007.

[2] D. Althoff, M. Althoff, and S. Scherer, "Online safety verification of trajectories for unmanned flight with offline computed robust invariant sets," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[3] S. Olaru, J. D. Dona, M. Seron, and F. Stoican, "Positive invariant sets for fault tolerant multisensor control schemes," *International Journal of Control*, vol. 83, no. 12, pp. 2622–2640, 2010.

[4] F. F. Tahir and M. Jaimoukha, "Low-complexity polytopic invariant sets for linear systems subject to norm-bounded uncertainty," *IEEE Trans. Autom. Control*, vol. 60, pp. 1416–1421, 2015.

[5] S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne, "Invariant approximations of the minimal robust positively invariant set," *IEEE Trans. Autom. Control*, vol. 50, no. 3, pp. 406–410, 2005.

[6] E. Kofman, M. M. Seron, and H. Haimovich, "Robust control design with guaranteed state ultimate bound," in *Proceedings of the 3rd International Conference on Integrated Modeling and Analysis in Applied Control and Automation*, 2007.

[7] S. Rakovic, E. Kerrigan, K. Kouramas, and D. Mayne, "Invariant approximations of robustly positively invariant sets for constrained linear discrete-time systems subject to bounded disturbances," *Dept of Eng., University of Cambridge, Tech. Rep*, 2004.

[8] J. Wan, J. Vehi, and N. Luo, "A numerical approach to design control invariant sets for constrained nonlinear discrete-time systems with guaranteed optimality," *"Journal of Global Optimization"*, vol. 44, pp. 395–407, 2009.

[9] H. Khalil, *Nonlinear Systems, Third Edition*. Prentice Hall, 2002.

[10] J.-J. E. Slotine and W. Li, *Applied nonlinear control*. Englewood Cliffs (N.J.): Prentice Hall, 1991. [Online]. Available: http://opac.inria.fr/record=b1132812

[11] W. Tucker, "The Lorenz Attractor Exists," *Comptes Rendus de l'Académie des Sciences*, vol. 328, no. 12, pp. 1197–1202, 1999.

[12] A. Goldsztejn, W. Hayes, and P. Collins, "Tinkerbell Is Chaotic," *SIAM Journal on Applied Dynamical Systems*, vol. 10, no. 4, pp. 1480–1501, 2011.

[13] W. Kühn, "Rigorously computed orbits of dynamical systems without the wrapping effect," *Computing*, vol. 61, pp. 47–67, 1998.

[14] M. Berz, C. Bischof, G. Corliss, and G. A., Eds., *Computational Differentiation: Techniques, Applications and Tools*. Philadelphia, Penn.: SIAM, 1996.

[15] D. Wilczak and P. Zgliczynski, "Cr-lohner algorithm," *Schedae Informaticae*, vol. 20, pp. 9–46, 2011.

[16] R. E. Moore, *Methods and Applications of Interval Analysis*. Philadelphia, PA: SIAM, 1979.

[17] J. A. dit Sandretto and A. Chapoutot, "Validated explicit and implicit runge-kutta methods," *Reliable Computing*, vol. 22, p. 79, 2016.

[18] N. Revol, K. Makino, and M. Berz, "Taylor models and floating-point arithmetic: proof that arithmetic operations are validated in COSY," *Journal of Logic and Algebraic Programming*, vol. 64, pp. 135–154, 2005.

[19] P. Collins and A. Goldsztejn, "The Reach-and-Evolve Algorithm for Reachability Analysis of Nonlinear Dynamical Systems," *Electronic Notes in Theoretical Computer Science*, no. 223, pp. 87–102, 2008.

[20] N. Ramdani and N. Nedialkov, "Computing Reachable Sets for Uncertain Nonlinear Hybrid Systems using Interval Constraint Propagation Techniques," *Nonlinear Analysis: Hybrid Systems*, vol. 5, no. 2, pp. 149–162, 2011.

[21] E. Goubault, O. Mullier, S. Putot, and M. Kieffer, "Inner approximated reachability analysis," in *Proceedings of the 17th international conference on Hybrid systems: computation and control, HSCC '14*, Berlin, Germany, 2014, pp. 163–172.

[22] M. Lhommeau, L. Jaulin, and L. Hardouin, "Capture Basin Approximation using Interval Analysis," *International Journal of Adaptative Control and Signal Processing*, vol. 25, no. 3, pp. 264–272, 2011.

[23] S. Ratschan and Z. She, "Providing a Basin of Attraction to a Target Region of Polynomial Systems by Computation of Lyapunov-like Functions ," *SIAM J. Control and Optimization*, vol. 48, no. 7, pp. 4377–4394, 2010.

[24] L. Jaulin and F. L. Bars, "An Interval Approach for Stability Analysis; Application to Sailboat Robotics," *IEEE Transaction on Robotics*, vol. 27, no. 5, 2012.

[25] J. Yorke, "Invariance for ordinary differential equations," *Mathematical System Theory*, vol. 1, no. 4, pp. 353–372, 1967.

[26] L. Lapierre, R. Zapata, and P. Lépinay, "Combined Path-following and Obstacle Avoidance Control of a Wheeled Robot," *International Journal of Robotics Research*, vol. 26, no. 4, pp. 361–375, 2007.

[27] S. L. Menec, "Linear Differential Game with Two Pursuers and One Evader," in *Advances in Dynamic Games*, M. Breton and K. Szajowski, Eds., vol. 11, 2011, pp. 209–226.

[28] O. Bouissou, A. Chapoutot, A. Djaballah, and M. Kieffer, "Computation of parametric barrier functions for dynamical systems using interval analysis," in *2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*, Dec. 2014, pp. 753–758.

[29] Y. Deville, M. Janssen, and P. V. Hentenryck, "Consistency techniques in ordinary differential equations," in *Proceedings of the Fourth International Conference on Principles and Practice of Constraint Programming*, ser. Lecture Notes in Computer Science. Springer Verlag, 1998.

[30] M. Spivak, *Calculus On Manifolds: A Modern Approach To Classical Theorems Of Advanced Calculus*. Westview Press, 1965.

[31] A. K. Mackworth, "Consistency in networks of relations," *Artificial Intelligence*, vol. 8, no. 1, pp. 99–118, Feb. 1977.

[32] E. Goubault and S. Putot, "Static analysis of numerical algorithms," in *In Proceedings of SAS 06, LNCS 4134*. Springer-Verlag, 2006, pp. 18–34.

[33] I. Araya, G. Trombettoni, and B. Neveu, "A Contractor Based on Convex Interval Taylor," in *Proc. of CPAIOR*, LNCS 7298, Springer, 2012, pp. 1–16.

[34] F. L. Bars, J. Sliwka, O. Reynet, and L. Jaulin, "State estimation with fleeting data," *Automatica*, vol. 48, no. 2, pp. 381–387, 2012.

[35] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. London: Springer-Verlag, 2001.

[36] C. Combastel, "A state bounding observer for uncertain non-linear continuous-time systems based on zonotopes," in *CDC-ECC '05*, 2005.

[37] V. Drevelle and P. Bonnifait, "Localization confidence domains via set inversion on short-term trajectory," *IEEE Transactions on Robotics*, 2013.

[38] G. Chabert and L. Jaulin, "QUIMPER, A Language for Quick Interval Modelling and Programming in a Bounded-Error Context," *Artificial Intelligence*, vol. 173, pp. 1079–1100, 2009.

[39] P. Cousot and R. Cousot, "Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints," in *Conference Record of the Fourth ACM Symposium on Principles of Programming Languages*, Los Angeles, California, 1977, pp. 238–252.

[40] G. Trombettoni and G. Chabert, "Constructive Interval Disjunction," in *Proc. CP, Constraint Programming*, LNCS 4741, 2007, pp. 635–650.

[41] S. Lagrange, N. Delanoue, and L. Jaulin, "On sufficient conditions of injectivity, development of a numerical test via interval analysis," *Reliable computing*, vol. 13, no. 5, pp. 409–421, 2007.

[42] P. Saint-Pierre, "Approximation of the Viability Kernel," *Applied Mathematics and Optimization*, vol. 29, no. 2, 1994.

[43] L. Jaulin, "Outer approximation of attractors using an interval quantization," *Reliable Computing*, vol. 19, pp. 261–273, 2014.

[44] A. Tarski, "A lattice-theoretical fixpoint theorem and its applications," *Pacific Journal of Mathematics*, vol. 5, no. 2, pp. 285–309, 1955.

[45] N. Nedialkov, K. Jackson, and G. Corliss, "Validated Solutions of Initial Value Problems for Ordinary Differential Equations," *Applied Mathematics and Computation*, vol. 105, no. 1, pp. 21–68, 1999.