Kleene Algebra to Compute Invariant Sets of Dynamical Systems

Thomas Le Mézo, Luc Jaulin, Damien Massé, Benoit Zerr

algorithms

MDPI

Article

Kleene Algebra to Compute Invariant Sets of Dynamical Systems

Thomas Le Mézo [1,*], Luc Jaulin [1], Damien Massé [2] and Benoit Zerr [1]

[1] ENSTA Bretagne, Robex, LabSTICC, 2 Rue François Verny, 29806 Brest, France; luc.jaulin@ensta-bretagne.fr (L.J.); benoit.zerr@ensta-bretagne.fr (B.Z.)
[2] Lab-STICC, UMR 6285, University of Brest, 29238 Brest, France; damien.masse@univ-brest.fr
* Correspondence: thomas.le_mezo@ensta-bretagne.org

Abstract: In this paper, we show that a basic fixed point method used to enclose the greatest fixed point in a Kleene algebra will allow us to compute inner and outer approximations of invariant-based sets for continuous-time nonlinear dynamical systems. Our contribution is to provide the definitions and theorems that will allow us to make the link between the theory of invariant sets and the Kleene algebra. This link has never be done before and will allow us to compute rigorously sets that can be defined as a combination of positive invariant sets. Some illustrating examples show the nice properties of the approach.

Keywords: invariant sets; Kleene algebra; nonlinear dynamical systems; path planning

Citation: Le Mézo, T.; Jaulin, L.; Massé, D.; Zerr, B. Kleene Algebra to Compute Invariant Sets of Dynamical Systems. *Algorithms* **2022**, *15*, 90. https://doi.org/10.3390/a15030090

Academic Editor: Pierre Leone

Received: 28 January 2022
Accepted: 4 March 2022
Published: 8 March 2022

1. Introduction

In this paper, we deal with a dynamical system $\mathcal{S}$ defined by the following state equation:

$$\mathcal{S} : \dot{\mathbf{x}}(t) = \boldsymbol{\gamma}(\mathbf{x}(t)) \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector and $\boldsymbol{\gamma} : \mathbb{R}^n \mapsto \mathbb{R}^n$ is the evolution function of $\mathcal{S}$ [1,2]. Denote by $\boldsymbol{\varphi}_\gamma$ the flow map of (1); i.e., with the initial vector $\mathbf{x}_0 = \mathbf{x}(0)$, the system $\mathcal{S}$ reaches the state $\boldsymbol{\varphi}_\gamma(t, \mathbf{x}_0)$ at time t. Our goal is to compute invariant sets [3] associated to the system with an algebraic approach, which is new in this context. Moreover, we propose to compute sets that can be expressed as a combination (union, intersection, image by a function, etc.) of invariant sets. In the application section of this paper, we will show why computing such combinations may be important in practice.

In particular, we will show the link between problems that can be expressed in terms of invariant sets and the Kleene algebra, the elements of which are automorphisms of a lattice [4]. We will take advantage of this algebraic structure to derive new efficient algorithms that are able to solve problems involving invariant sets that were not possible to compute with existing methods.

Our approach does not only provide a method to compute invariant sets. However, it allows us to compute sets that can be defined as a combination of invariant sets, which is the main contribution. We decided to explore an algebraic approach in order to get an elegant formalization of the problem resolution, as shown in several examples at the end of this paper.

A specific type of invariant set will be considered: the *positive invariant set*, which is an important concept in control [5], fault detection [6], safety [7], verification [8,9], or reachability [10].

A set $\mathbb{A}$ is *positive invariant* of (1) if we have

$$\mathbf{a} \in \mathbb{A}, t \geq 0 \implies \boldsymbol{\varphi}_\gamma(t, \mathbf{a}) \in \mathbb{A}. \tag{2}$$

The set of all positive invariant sets is a complete lattice; i.e., the union and the intersection of two positive invariant sets is positive invariant. A consequence is that given a set $\mathbb{A}$, the notion of the *greatest positive invariant set* contained in $\mathbb{X}$ and *smallest positive invariant set* enclosing $\mathbb{A}$ can be defined. For instance, the greatest positive invariant set for (1) included in $\mathbb{A}$ is given by

$$\text{Inv}^{+}(\gamma, \mathbb{A}) = \Big\{\mathbf{a} | \forall t \geq 0, \boldsymbol{\varphi}_{\gamma}(t, \mathbf{a}) \in \mathbb{A}\Big\}. \tag{3}$$

Methods exist to characterize positive invariant sets for specific cases such as for instance when γ is linear [11–13] or discrete time [14]. Outer approximation can be computed by providing a model for approximation [15] or support functions [16]. Moreover, most approximations correspond to convex sets such as ellipsoids, zonotopes [17], or polytopes. These linear-based approaches can be extended to hybrid linear systems [18–20].

For continuous-time nonlinear systems, computing invariant sets is much more difficult, and different types of approaches can be extracted from the literature.

- The first approach is based on sampling. It has been used for instance by Saint Pierre [21] to rigorously compute viability kernel, which is a specific type of controlled invariant set. Bobiti and Lazar [22] also used a sampling-based method for stability verification of both continuous and discrete-time nonlinear systems. To get guaranteed results, an interval integration is often needed (see e.g., [23–25]).
- The second approach is based on Lyapunov theory [3,26] and is convenient for proving asymptotic stability of problems with an infinite time horizon. The principle is to find a parameter vector $\mathbf{p}$ or a Lyapunov-like function $V(\mathbf{p}, \mathbf{x})$ such that the set

  $$\{\mathbf{x} | V(\mathbf{p}, \mathbf{x}) = 0 \text{ and } \dot{V}(\mathbf{p}, \mathbf{x}) \geq 0\}$$

  is empty. In such a case, the set $\mathbb{S}(\mathbf{p}) = \{\mathbf{x} | V(\mathbf{p}, \mathbf{x}) \leq 0\}$ defines a positive invariant set. The function that is generally chosen for $V(\mathbf{p}, \mathbf{x})$ is polynomial in the x_i's and linear in the p_j's. When γ is polynomial, this choice for $V(\mathbf{p}, \mathbf{x})$ makes it possible to use LMI/SOS/interval resolution techniques, which can be efficient when a low-dimensional approximation exists. This approach relies on assumptions such as the knowledge of an equilibrium point [27] or the polynomial property of the dynamics of the system [28], which is not always realistic.
- The third approach is based is based on *occupation measures*, which are adapted to a finite time horizon [27]. Now, it can also consider problems with infinite time horizon at the price of technical difficulties [29].
- The fourth approach is based a polygonal decomposition of the state space [30,31] and corresponds to the approach we will consider in the paper. For instance, in [30], a triangulated region yields an index filtration for a Morse decomposition of the flow on the system [32], which approximates the flow arbitrarily closely.

Different solvers such as SpaceEx [33,34] are available to compute numerically such invariant sets.

The paper proposes to compute inner and outer approximations of invariant sets in the general case where the system is continuous-time and nonlinear. More than that, it introduces for the first time an approach based on the *Kleene algebra* to compute sets that can be defined as operations on invariant sets. A possible application is the *path planning and avoid* problem [35], where we search for the set of all initial conditions for trajectories starting from of set $\mathbb{A}$ reaching the set $\mathbb{B}$ while avoiding the set $\mathbb{C}$. It is not possible to solve this type of problem rigorously using existing approaches.

This paper is organized as follows. Section 2 recalls some notions on lattices that will be needed to understand the Kleene algebra in Section 3. Then, the specific case where the elements of the Kleene algebra are made with automorphism is considered in Section 4. The link with invariant sets in dynamical systems is introduced in Section 5. Some illustrative test cases are given in Section 6. Section 7 concludes the paper.

2. Lattices

Invariant sets have a lattice structure [36], and this will allow us to formalize our problem in an algebraic form. This section recalls some classical definitions on lattices and provides some illustrations to understand the basic principles that are needed to understand our methodology.

2.1. Definitions

A *lattice* $(\mathcal{L}, \leq, \wedge, \vee)$ is a partially ordered set, closed under least upper and greatest lower bounds [37]. The least upper bound of x and y is called the *join* and is denoted by $x \vee y$. The greatest lower bound is called the *meet* and is written as $x \wedge y$.

A lattice $\mathcal{E}$ is *complete* if for all (finite or infinite) subsets $\mathcal{A}$ of $\mathcal{E}$, the least upper bound $\vee\mathcal{A}$ and the greatest lower bound $\wedge\mathcal{A}$ belong to $\mathcal{E}$. We define the top and the bottom of $\mathcal{E}$ as $\top = \vee\mathcal{E}$ and $\bot = \wedge\mathcal{E}$. A sublattice of a lattice $(\mathcal{L}, \leq)$ is a nonempty subset of $\mathcal{L}$ that is a lattice with the same meet and join operations $\vee$ and $\wedge$ as $\mathcal{L}$.

2.2. Machine Lattice

Consider a complete lattice $(\mathcal{L}, \leq, \wedge, \vee)$. A *machine lattice* $(\mathcal{L}_M, \leq, \wedge, \vee)$ of $\mathcal{L}$ is a complete sublattice of $(\mathcal{L}, \leq, \wedge, \vee)$ which is finite (thus, we can store it in the memory of a computer [38]). Moreover, both $\mathcal{L}$ and $\mathcal{L}_M$ have the same top $\top$ and bottom $\bot$. This is illustrated by Figure 1, which can be interpreted as follows.

- The gray square represents $\mathcal{L}$.
- The element k is greater than f, since it is at its top right.
- The grid made with blue dots corresponds to $\mathcal{L}_M$.
- The variables a, b, c, d all belong to $\mathcal{L}$, and we have $c = a \vee b$ and $d = a \wedge b$.
- The red polygon $\mathcal{P}$ is a sub-lattice of $\mathcal{L}$. Its bottom is $\bot_{\mathcal{P}} = \bot$ and its top is $\top_{\mathcal{P}} = e$.
- The element $i \in \mathcal{L}$ is inside $[\bot_{\mathcal{P}}, \top_{\mathcal{P}}]$. Thus, there exists an element n in $\mathcal{P}$ which corresponds to the smallest element in $\mathcal{P}$, which is larger than i. There exists also an element m in $\mathcal{P}$ that corresponds to the greatest element in $\mathcal{P}$, which is smaller than i.
- The smallest machine interval containing e is $[g, f]$.

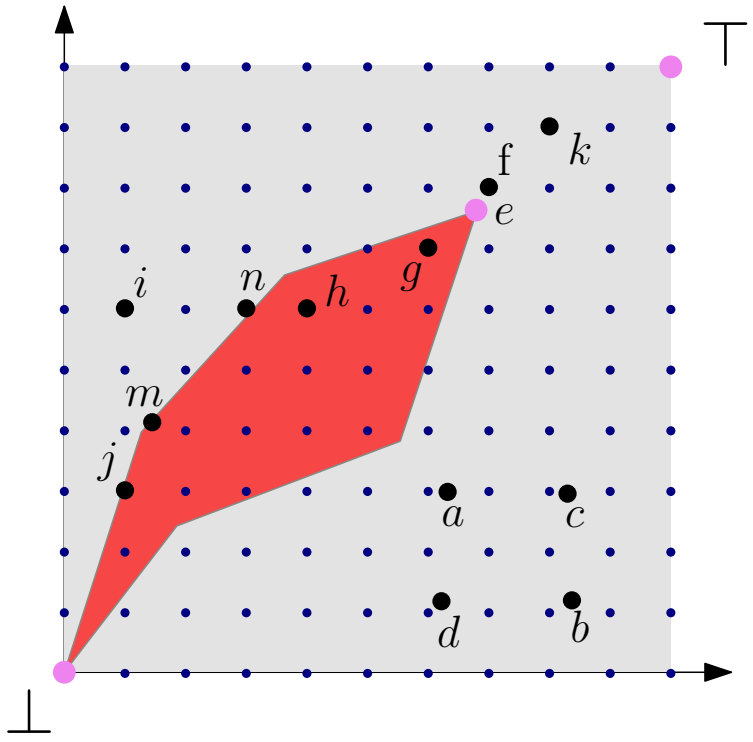

Figure 1. The grid corresponds to the machine lattice $\mathcal{L}_M$ associated to the lattice $\mathcal{L}$.

Remark 1. *In our context, the lattice $\mathcal{L}$ will correspond to the set of all subsets of $\mathbb{R}^n$ and $\mathcal{L}_M$ to the set of all machine sets (or mazes [39,40]). In Figure 1, the red polygon could represent the set of all positive invariant sets included in the set represented by e. The figure could also illustrate that given a set $\mathbb{A}$, there exists a smallest invariant set that contains $\mathbb{A}$, and there exists a largest element in $\mathcal{P}$, which is included in $\mathbb{A}$.*

3. Kleene Algebra

The invariant sets of a dynamical system can be defined as fixed points of monotonic operators, which can be formalized elegantly using a Kleene algebra. The basic notions related to Kleene algebras are recalled in this section.

3.1. Definition

A Kleene algebra $(\mathcal{K}, +, \cdot, *)$ is a set $\mathcal{K}$ together with two binary operations $+ : \mathcal{K} \times \mathcal{K} \to \mathcal{K}$ and $\cdot : \mathcal{K} \times \mathcal{K} \to \mathcal{K}$ and one function $* : \mathcal{K} \to \mathcal{K}$, so that the axioms listed in the Table 1 are satisfied.

The literature contains several inequivalent definitions of Kleene algebras and related algebraic structures [41]. Our definition differs when we introduce the Kleene operator: we assume *star-continuity*, which is needed as soon as infinite cardinal sets are considered.

Probably the mainstream definition is that of Kozen in [42], where the Kleene operator is defined formally by the following properties

$$\begin{gathered} 1 + aa^* \leq a^* \\ 1 + a^*a \leq a^* \\ ax \leq x \Rightarrow a^*x \leq x \\ xa \leq x \Rightarrow xa^* \leq x \end{gathered}$$

and the star-continuity is defined by

$$xa^*y = \textstyle\sum_{i \geq 0} xa^iy.$$

Now, if we take $x = y = 1$, we get

$$a^* = 1 \cdot a^* \cdot 1 = \sum_{i \geq 0} 1 \cdot a^i \cdot 1 = \sum_{i \geq 0} a^i.$$

which corresponds to our definition.

The following proposition illustrates the fact that applying a Kleene star operator to a amounts to computing a fixed point. This will be used later in the algorithms.

Proposition 1. *Given a star continuous Kleene algebra $(\mathcal{K}, +, \cdot, *)$, we have*

$$(1 + a)^\infty = \lim_{i \to \infty} (1 + a)^i = a^* \tag{4}$$

where

$$\begin{array}{rcl} a^0 & = & 1 \\ a^{i+1} & = & a \cdot a^i \\ a^\infty & = & \lim_{i \to \infty} a^i \quad \textit{(if it exists)} \end{array}$$

Proof. We first show by induction that

$$(1 + a)^n = \sum_{i \in \{0, n\}} a^i. \tag{5}$$

Assume that the proposition is true for n. We have

$$\begin{array}{rcl}(1+a)^{n+1} & = & (1+a)(1+a)^n \\ & \overset{(5)}{=} & (1+a)\cdot\sum_{i\in\{0,n\}} a^i \\ & = & \left(\sum_{i\in\{0,n\}} a^i + a\cdot\sum_{i\in\{0,n\}} a^i\right) \\ & = & \left(\sum_{i\in\{0,n\}} a^i + \sum_{i\in\{0,n\}} a^{i+1}\right) \\ & = & \left(\sum_{i\in\{0,n\}} a^i + \sum_{i\in\{1,n+1\}} a^i\right) \\ & = & \sum_{i\in\{0,n+1\}} a^i.\end{array}$$

Since the two sequences involved in (5) are equal, they have the same limit. Therefore,

$$\lim_{n\to\infty}(1+a)^n = \lim_{n\to\infty}\left(\sum_{i\in\{0,n\}} a^i\right) = \sum_{i\in\{0,\ldots,\infty\}} a^i = a^*.$$

□

Table 1. Axioms/definitions of a star-continuous Kleene algebra.

Kleene algebra	$(\mathcal{K},+,\cdot,*)$
Addition	$a+b$
Product	$a\cdot b$
Associativity	$a+(b+c)=(a+b)+c$ $a\cdot(b\cdot c)=(a\cdot b)\cdot c$
Commutativity	$a+b=b+a$
Distributivity	$a\cdot(b+c)=(a\cdot b)+(a\cdot c)$ $(b+c)\cdot a=(b\cdot a)+(c\cdot a)$
Zero	$a+0=a$
One	$a\cdot 1=1\cdot a=a$
Annihilation	$a\cdot 0=0\cdot a=0$
Idempotence	$a+a=a$
Partial order	$a\leq b\Leftrightarrow a+b=b$
Kleene star	$a^*=1+a+a\cdot a+a\cdot a\cdot a+\ldots$

3.2. Intervals

Consider a star-continuous Kleene algebra $(\mathcal{K},+,\cdot,*)$. It is equipped with the order relation $\leq$ defined by $a\leq b\Leftrightarrow a+b=b$ (see Table 1). We consider $(\mathcal{K}_M,\leq,\wedge,\vee)$ a machine lattice of $\mathcal{K}$ with respect to $\leq$, i.e.,

$$\begin{array}{rcl}(i)\quad a\wedge b & = & \max\{c\in\mathcal{K}_M | c\leq a, c\leq b\} \\ & = & \sum\{c\in\mathcal{K}_M | c\leq a, c\leq b\} \\ (ii)\quad a\vee b & = & \min\{c\in\mathcal{K}_M | a\leq c, b\leq c\} \\ & = & a+b.\end{array}$$

The max and min are used since $\mathcal{K}_M$ is finite. To prove (i) we consider c,d in $\mathcal{K}_M$. We have

$$\begin{array}{rcl}\left\{\begin{array}{l}c\leq a\wedge b\\ d\leq a\wedge b\end{array}\right. & \Rightarrow & \left\{\begin{array}{l}c\leq a, c\leq b\\ d\leq a, d\leq b\end{array}\right. \\ & \Rightarrow & \left\{\begin{array}{l}c+d\leq a+a\\ c+d\leq b+b\end{array}\right. \\ & \Rightarrow & \left\{\begin{array}{l}c+d\leq a\\ c+d\leq b\end{array}\right. \\ & \Rightarrow & c+d\leq a\wedge b\end{array}$$

such that $c \leq a, c \leq b$ and $d \leq a, d \leq b$. To check that $a \vee b = a + b$ (in (ii)), it suffices to observe that $a \leq a + b$ and that $b \leq a + b$.

It can be shown that $(\mathcal{K}_M, +, \cdot, *)$ is also a Kleene algebra. Note that since $\mathcal{K}_M$ is finite, it is obviously star-continuous.

Definition 1. *An interval of $(\mathcal{K}, +, \cdot, *)$ is a subset $[a]$ of $\mathcal{K}$ that can be written as*

$$[a] = \left[a^-, a^+\right] = \left\{a \in \mathcal{K} | a^- \leq a \leq a^+\right\}$$

where a^-, a^+ belong to $\mathcal{K}_M$.

Note that both $\emptyset$ and $\mathcal{K}$ are intervals of $\mathcal{K}$. Thus, an interval arithmetic similar to that proposed by Moore [43] for real numbers can be derived. This will be used later to compute with quantities defined as expressions of fixed point operators. As a consequence, if $a \in [a] = [a^-, a^+]$, $b \in [b] = [b^-, b^+]$, we have

$$\begin{array}{rcl} [a^-, a^+]^* & = & \left[(a^-)^*, (a^+)^*\right] \\ [a^-, a^+] + [b^-, b^+] & = & [a^- + b^-, a^+ + b^+] \\ [a^-, a^+] \cdot [b^-, b^+] & = & [a^- \cdot b^-, a^+ \cdot b^+]. \end{array} \tag{6}$$

due to the monotonicity of the operators $+, \cdot, *$.

4. Kleene Algebra of Automorphisms

The notions presented in Sections 2 and 3 can be seen as direct extensions of existing algebraic approaches related to lattices and Kleene algebra. In this section, we introduce automorphism-based Kleene algebras [4], which will allow us to make a first bridge between algebraic tools and invariant sets of nonlinear dynamical systems.

4.1. Automorphisms

Given a complete lattice $(\mathcal{L}, \leq, \wedge, \vee, \bot, \top)$, an automorphism of $\mathcal{L}$ is a function $f{:}\mathcal{L} \to \mathcal{L}$ such that

$$\begin{array}{lrcl} \text{(i)} & f(\top) & = & \top \\ \text{(ii)} & f(a \wedge b) & = & f(a) \wedge f(b) \end{array}. \tag{7}$$

As we will see later (see, e.g., Table 2), no property is assumed concerning the operator $\vee$.

We denote by $\mathcal{A}(\mathcal{L})$ the set of automorphisms of $\mathcal{L}$.

Proposition 2. *If f, g are in $\mathcal{A}(\mathcal{L})$, then:*

$$\begin{array}{ll} \textit{(i)} & a \leq b \Rightarrow f(a) \leq f(b) \\ \textit{(ii)} & f \wedge g \in \mathcal{A}(\mathcal{L}) \\ \textit{(iii)} & f \circ g \in \mathcal{A}(\mathcal{L}) \\ \textit{(iv)} & Id \wedge f \wedge f^2 \wedge f^3 \wedge \cdots \in \mathcal{A}(\mathcal{L}) \end{array} \tag{8}$$

where the function $f \wedge g$ is defined by $(f \wedge g)(a) = f(a) \wedge f(a)$.

Proof. (i) We have

$$\begin{array}{rcll} a \leq b & \Leftrightarrow & a = b \wedge a & \\ & \Rightarrow & f(a) = f(b \wedge a) & \\ & \Leftrightarrow & f(a) = f(b) \wedge f(a) & \text{(see Eq 7,ii)} \\ & \Rightarrow & f(a) \leq f(b). & \end{array} \tag{9}$$

(ii) We have $(f \wedge g)(\top) = \top \wedge \top = \top$. Moreover,

$$\begin{array}{rcl}(f \wedge g)(a \wedge b) & = & (f(a \wedge b)) \wedge (g(a \wedge b)) \\ & = & f(a) \wedge f(b) \wedge g(a) \wedge g(b) \\ & = & (f \wedge g)(a) \wedge (f \wedge g)(b)\end{array} \tag{10}$$

(iii) We have $f \circ g(\top) = f(\top) = \top$. Moreover,

$$\begin{array}{rcl}(f \circ g)(a \wedge b) & = & f(g(a) \wedge g(b)) \\ & = & (f \circ g)(a) \wedge (f \circ g)(b)\end{array} \tag{11}$$

(iv) It corresponds to the Kleene operator and is a direct consequence of (ii), (iii). □

Proposition 3. *Assume that for $f \in \mathcal{A}(\mathcal{L})$, $f^* = Id \wedge f \wedge f^2 \wedge f^3 \wedge \ldots$ exists (star-continuity assumption), then the set $(\mathcal{A}(\mathcal{L}), \wedge, \circ, *)$ is a Kleene algebra.*

Proof. We need to check all properties of Table 2. In the two tables, the symbols of the operators changed: $+$ corresponds to $\wedge$, $*$ corresponds to $\circ$, and $\geq$ corresponds to $\leq$. To prove the distributivity $f \circ (g \wedge h) = (f \circ g) \wedge (f \circ h)$, we proceed as follows:

$$\begin{array}{rcl}f \circ (g \wedge h)(a) & = & f \circ (g(a) \wedge h(a)) \\ & \overset{(7,ii)}{=} & (f \circ g)(a) \wedge (f \circ h)(a).\end{array} \tag{12}$$

All other properties can be proved similarly. □

Table 2. The set of automorphisms forms a star-continuous Kleene algebra.

Kleene Algebra	$(\mathcal{A}(\mathcal{L}), \wedge, \circ, *)$
Addition	$f \wedge g$
Product	$f \circ g$
Associativity	$f \wedge (g \wedge h) = (f \wedge g) \wedge h$ $f \circ (g \circ h) = (f \circ g) \circ h$
Commutativity	$f \wedge g = g \wedge f$
Distributivity	$f \circ (g \wedge h) = (f \circ g) \wedge (f \circ h)$ $(g \wedge h) \circ f = (g \circ f) \wedge (h \circ f)$
Zero	$f \wedge \top = f$
One	$f \circ \mathrm{Id} = \mathrm{Id} \circ f = f$
Annihilation	$f \circ \top = \top$
Idempotency	$f \wedge f = f$
Partial order	$f \geq g \Leftrightarrow f \wedge g = g$
Kleene star	$f^* = \mathrm{Id} \wedge f \wedge f^2 \wedge f^3 \wedge \ldots$

Remark 2. *The relation order for automorphism has been chosen backward (i.e., $\geq$ instead of $\leq$). This choice is motivated by the fact that the inclusion order will be used later, and we will have the following correspondences: $\leq \leftrightarrow \subset$, $\wedge \leftrightarrow \cap$.*

Proposition 4. *The fixed points of $Id \wedge f$ correspond to that of f^**, i.e.,

$$\mathrm{Fix}(f^*) = \{a | f^*(a) = a\} = \mathrm{Fix}(\mathrm{Id} \wedge f) \tag{13}$$

Proof. The proof is decomposed in two parts. In the first part, we prove that $\mathrm{Fix}(\mathrm{Id} \wedge f) \subset \mathrm{Fix}(f^*)$ and in the second part, we prove the inverse.

(1) We have

$$\begin{array}{rcc}(\mathrm{Id}\wedge f)(x)=x & \Rightarrow & \forall i\geq 1,(\mathrm{Id}\wedge f)^i(x)=x\\ & \Rightarrow & (\mathrm{Id}\wedge f)^\infty(x)=x\\ & \Leftrightarrow & f^*(x)=x\end{array}$$

since, from (4), we have $(\mathrm{Id}\wedge f)^\infty = f^*$. Thus, $\mathrm{Fix}(\mathrm{Id}\wedge f)\subset \mathrm{Fix}(f^*)$.

(2) We now want to prove the inverse inclusion: $\mathrm{Fix}(f^*)\subset \mathrm{Fix}(\mathrm{Id}\wedge f)$.

We first prove by induction that

$$\forall i\geq 0,(\mathrm{Id}\wedge f)^i\geq(\mathrm{Id}\wedge f)^{i+1}. \tag{14}$$

From the partial order property of Table 2, we have

$$\mathrm{Id}\geq(\mathrm{Id}\wedge f)\Leftrightarrow \mathrm{Id}\wedge(\mathrm{Id}\wedge f)=\mathrm{Id}\wedge f \tag{15}$$

since $\mathrm{Id}\wedge(\mathrm{Id}\wedge f)=(\mathrm{Id}\wedge\mathrm{Id})\wedge f=\mathrm{Id}\wedge f$, we get that the right-hand side of (15) is true and thus $\mathrm{Id}\geq \mathrm{Id}\wedge f$ is true. This means that for $i=0$, (14) is true.

Assume now that (14) is true for i, let us check that it is true for $i+1$. From Proposition 2 (i), we get that

$$f\geq g\Rightarrow h\circ f\geq h\circ g.$$

With the substitution $f\rightarrow(\mathrm{Id}\wedge \mathrm{f})^i, g\rightarrow(\mathrm{Id}\wedge f)^{i+1}, h\rightarrow \mathrm{Id}\wedge f$, we get

$$(\mathrm{Id}\wedge f)^i\geq(\mathrm{Id}\wedge f)^{i+1}\Rightarrow(\mathrm{Id}\wedge f)\circ(\mathrm{Id}\wedge f)^i\geq(\mathrm{Id}\wedge f)\circ(\mathrm{Id}\wedge f)^{i+1}.$$

Since (14) is true for i, we get

$$(\mathrm{Id}\wedge f)^{i+1}\geq(\mathrm{Id}\wedge f)^{i+2},$$

thus, we have proved (14).

From (14), we get $(\mathrm{Id}\wedge f)^i$ is a decreasing sequence and thus

$$\mathrm{Id}\geq(\mathrm{Id}\wedge f)\geq(\mathrm{Id}\wedge f)^i\geq(\mathrm{Id}\wedge f)^\infty.$$

Take a fixed point x of $(\mathrm{Id}\wedge f)^\infty$, we get

$$x\geq(\mathrm{Id}\wedge f)(x)\geq(\mathrm{Id}\wedge f)^\infty(x)=x,$$

thus, $(\mathrm{Id}\wedge f)(x)=x$, which implies that x is also a fixed point of $\mathrm{Id}\wedge f$. □

Since f is a monotonic, so is $\mathrm{Id}\wedge f$ and f^*. From the Knaster–Tarski theorem, we deduce that $\mathrm{Fix}(f^*)$ is a complete sublattice of $\mathcal{L}$.

4.2. Factorization

In this paper, we want to compute expressions involving the Kleene star operator. For efficiency reasons, we want to avoid reaching the fixed point each time the star operator is used. Ideally, we would like to have a unique fixed point to be reached. For this, we need to transform an expression containing several stars into an expression with only one star. Equivalently, we want to factorize the fixed point operator as much as possible. For this, we can use the factorization rules such as [42]:

$$\begin{array}{c}f^*\wedge f^*=f^*\\ f^*\circ f^*=f^*\\ (f^*)^*=f^*\\ (f^*\wedge g^*)^*=(f\wedge g)^*\end{array} \tag{16}$$

but we can do more. Assume for instance that we have to compute

$$f^*(a) \wedge g^*(b), \tag{17}$$

we understand that it is not necessary to compute $f^*(a)$ and $g^*(b)$ independently to finally observe that we get $\bot$. For instance, we may have spent a lot of time to compute accurately $f^*(a)$ and a few milliseconds to get that $g^*(b) = \bot$ to finally reach the conclusion that $f^*(a) \wedge g^*(b) = \bot$. As a result, we need to develop some specific algorithms taking into account that calling closures has a cost. The factorization allows us to reduce several fixed point iterations into a single one. This can be used to increase the speed of convergence of fixed point algorithms.

4.3. Intervals

Given a lattice $(\mathcal{L}, \leq, \wedge, \vee)$ and an automorphism $f \in \mathcal{A}(\mathcal{L})$, we want to compute $f^*(a)$ where $a \in \mathcal{L}$. We consider also a machine lattice $\mathcal{L}_M$ of $\mathcal{L}$. An automorphism of $\mathcal{L}_M$ is called a *machine automorphism*. As seen in Subsection 3.2, since $\mathcal{A}(\mathcal{L})$ is a Kleene algebra, we can define intervals in $\mathcal{A}(\mathcal{L})$.

Definition 2. *An interval of $\mathcal{A}(\mathcal{L})$ is a subset $[f]$ of $\mathcal{A}(\mathcal{L})$, which can be written as*

$$[f] = \left[f^-, f^+\right] = \left\{f \in \mathcal{A}(\mathcal{L}) | f^- \leq f \leq f^+\right\} \tag{18}$$

where f^-, f^+ belong to $\mathcal{A}(\mathcal{L}_M)$.

Proposition 5. *We have*

$$Fix\left((f^-)^*\right) \subset \mathcal{L}_M \cap Fix(f^*) \subset Fix\left((f^+)^*\right). \tag{19}$$

Proof. Let us first prove the first inclusion. First, since $(f^-)^* \in \mathcal{A}(\mathcal{L}_M)$, all its fixed points are inside $\mathcal{L}_M$. We have

$$\begin{array}{lcll} a \in \mathrm{Fix}\left((f^-)^*\right) & \Rightarrow & (f^-)^*(a) = a, & \\ & \Leftrightarrow & a \wedge f^-(a) = a & \text{(see (13))} \\ & \Rightarrow & a \wedge f(a) \geq a & \text{(since } f^- \leq f\text{)} \\ & \Leftrightarrow & a \wedge f(a) = a & \text{(since } a \wedge x \leq a\text{)} \\ & \Leftrightarrow & f^*(a) = a & \text{(see (13)).} \end{array}$$

Let us now prove the second inclusion. We have

$$\begin{array}{lcl} & a \in \mathcal{L}_M \cap \mathrm{Fix}(f^*) & \\ \Rightarrow & f^*(a) = a, a \in \mathcal{L}_M & \\ \Leftrightarrow & a \wedge f(a) = a, a \in \mathcal{L}_M & \text{(see (13))} \\ \Rightarrow & a \wedge f^+(a) \geq a & \text{(since } f \leq f^+\text{)} \\ \Leftrightarrow & a \wedge f^+(a) = a & \text{(since } a \wedge x \leq a\text{)} \\ \Leftrightarrow & (f^+)^*(a) = a & \text{(see (13))} \end{array}$$

□

Theorem 1. *If $a \in [a^-, a^+]$, where a^-, a^+ both belong to $\mathcal{L}_M$, then*

$$\begin{array}{ll} \textit{(i)} & f^*(a) \in \left[(f^-)^*(a^-), (f^+)^*(a^+)\right] \\ \textit{(ii)} & f^* \circ (f^-)^*(a^-) = (f^-)^*(a^-) \\ \textit{(iii)} & f^*(a) \leq (Id \wedge f^+)^i(a^+), \forall i \geq 0 \end{array} \tag{20}$$

Remark 3. *The membership relation (i) means that we are able to compute in a finite time an enclosure of the fixed point $f^*(a)$. Relation (ii) states that the fixed point obtained by $(f^-)^*$ is a*

fixed point of f^*. However, this is not true for $(f^+)^*$. Relation (iii) tells us that at each iteration i, we have an upper bound for the solution $f^*(a)$, but we need to reach the fixed point to get a lower approximation $(f^-)^*(a^-)$ of $f^*(a)$.

Proof. (i) is a consequence of the fact that $f^i(a) \in \left[(f^-)^i(a^-), (f^+)^i(a^+)\right]$.

(ii) Since $(f^-)^* \circ (f^-)^*(a^-) = (f^-)^*(a^-)$, the quantity $(f^-)^*(a^-)$ is a fixed point of $(f^-)^*$. Now, from (19), we have $\mathrm{Fix}\left((f^-)^*\right) \subset \mathrm{Fix}(f^*)$. Therefore, $(f^-)^*(a^-)$ is also a fixed point of f^*, i.e., $f^* \circ (f^-)^*(a^-) = (f^-)^*(a^-)$.

(iii) We have

$$\begin{array}{ccccc} f^*(a) & = & (\mathrm{Id} \wedge f)^\infty(a) & \leq & (\mathrm{Id} \wedge f)^i(a) \\ & \leq & (\mathrm{Id} \wedge f^+)^i(a) & \leq & (\mathrm{Id} \wedge f^+)^i(a^+) \end{array}$$

□

The inclusion (iii) is illustrated by Figure 2 where the grid corresponds to $\mathcal{L}_M$, the magenta points correspond to $\mathrm{Fix}\left((f^-)^*\right)$, the blue points correspond to $\mathrm{Fix}\left((f^+)^*\right)$, and the light red polygon corresponds to $\mathrm{Fix}(f^*)$.

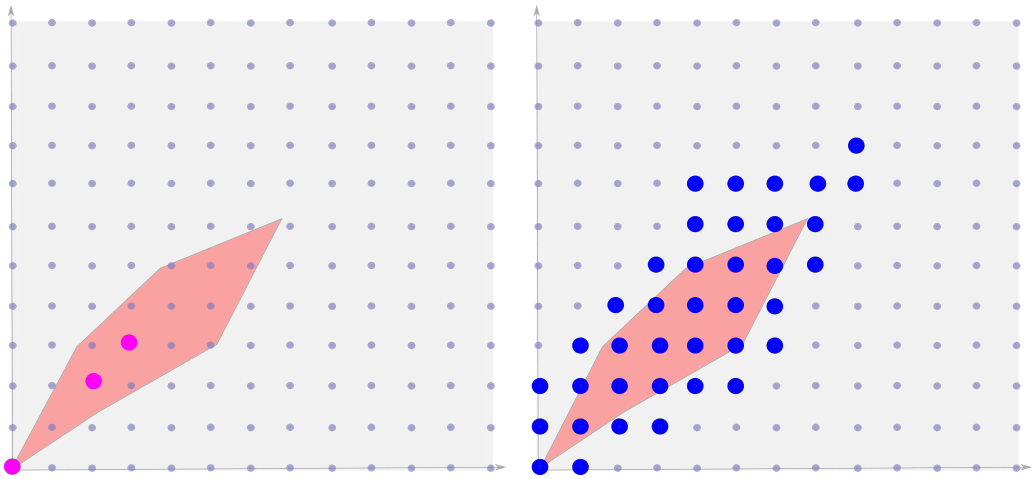

Figure 2. Fixed points $\mathrm{Fix}\left((f^-)^*\right)$ in magenta, $\mathrm{Fix}\left((f^+)^*\right)$ in blue, and $\mathrm{Fix}(f^*)$ is the light red polygon.

4.4. Algorithm to Find the Greatest Fixed Point

We propose here an interval algorithm [44] to compute $f^*(a)$, where $a \in [a] = [a^-, a^+]$. We assume that we have an interval $[f] = [f^-, f^+]$ containing f, where f^-, f^+ belong to $\mathcal{A}(\mathcal{L}_M)$. From Equation (20), we know that

$$f^*(a) \in \left[(f^-)^*(a^-), (\mathrm{Id} \wedge f^+)^i(a^+)\right]. \tag{21}$$

To compute $f^*(a)$, we apply the sequence of interval operations defined by

$$\begin{array}{ccc} [x](i+1) & = & (\mathrm{Id} \wedge [f])([x](i)) \\ [x](0) & = & [a] \end{array} \tag{22}$$

up to the fixed point. From Theorem 1, we get a guaranteed approximation of $f^*(a)$. Since $\mathcal{L}_M$ is finite, the algorithm always terminates. The principle of the algorithm is illustrated by Figure 3. First, a is approximated by an interval $[a^-, a^+]$ of $\mathcal{A}(\mathcal{L}_M)$. Then, we compute $(\mathrm{Id} \wedge [f])([a])$, which corresponds to $[b] = [b^-, b^+]$. Then, we compute $(\mathrm{Id} \wedge [f])([b])$, which corresponds to $[c] = [c^-, c^+]$. The last subfigure corresponds to the fixed point interval $[d] = ([f^*])([a])$ which contains the solution $f^*(a)$.

The sequence (4) provides a guaranteed enclosure for the solution, and the accuracy is related to the precision we used to define the machine lattice $\mathcal{L}_M$. Once the algorithm

terminates, if we are not satisfied by the quality of the approximation, we should restart from the beginning by redefining $\mathcal{L}_M$ with a finer level of granularity.

Remark 4. *In our implementation, a multi-scale approach is used for more efficiency and more accuracy: once the fixed point interval $[d^-, d^+]$ is reached, we build a new grid inside the interval $[d^-, d^+]$. We also combine with an inflation process, which increases d^- without overtaking the solution d. More precisely, in Figure 3, we may increase d^- top-right still staying inside the red polygon in order to get a more accurate approximation for the solution d. We called this process inflation, since in the context of this paper, the points of the figure correspond to subsets of $\mathbb{R}^n$ equipped with the order relation is $\subset$. When we increase d, we may understand that we inflate the set corresponding to d^- still being included in the solution set corresponding to d.*

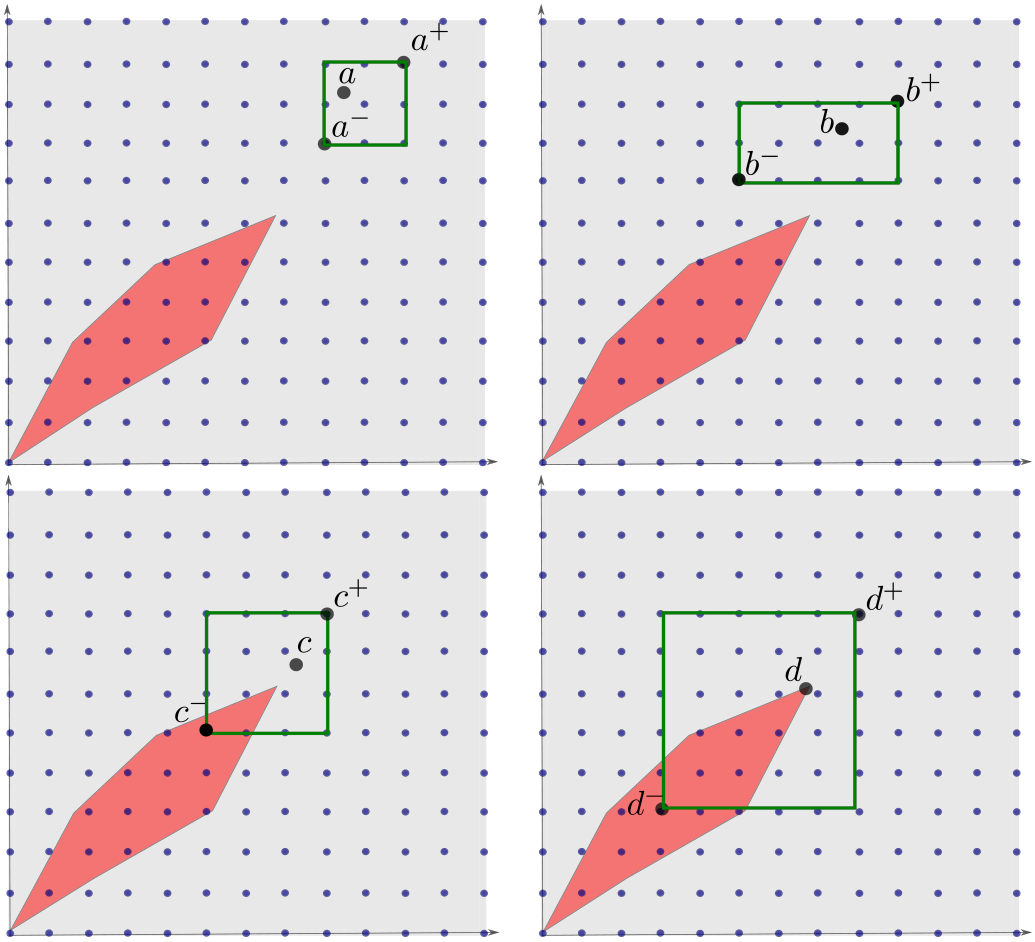

Figure 3. Algorithm that computes an approximation of $f^*(a), a \in [a]$.

5. Application to Dynamical Systems

In this section, we show that the previous algorithm can directly be used to compute invariant sets of continuous-time dynamical systems. Furthermore, we will show that we are able to compute sets that can be defined as a combination of invariant sets.

5.1. Greatest Positive Invariant

Consider the system $\mathcal{S}$ defined by Equation (1). The power set $\mathcal{P}(\mathbb{R}^n)$ of the state space $\mathbb{R}^n$ and equipped with $\cap, \cup$ is a lattice. We denote by $(\mathcal{A}(\mathcal{P}(\mathbb{R}^n)), \cap, \circ, *)$ the associated set of automorphisms. We want to find an automorphism $\overrightarrow{f}$ in $\mathcal{A}(\mathcal{P}(\mathbb{R}^n))$ that can be enclosed between two machine automorphisms and such that $\overrightarrow{f}^*(\mathbb{A})$ corresponds to the greatest positive invariant set included in $\mathbb{A}$, i.e.,

$$\overrightarrow{f}^*(\mathbb{A}) \quad = \quad \left\{\mathbf{a} \in \mathbb{A} | \forall t \geq 0, \boldsymbol{\varphi}_\gamma(t, \mathbf{a}) \in \mathbb{A}\right\} \tag{23}$$

We may find some tools for that such as CAPD [24] or a tube approach [45] devoted to this type of problem. Now, these types of approaches only consider a finite time integration and are unable to compute the fixed point (23) in a reasonable time as shown in [46,47]. Therefore, it is important to build an automorphism $\overrightarrow{f}$, which is fast to evaluate and that will converge quickly. This can be done by using an Eulerian positive predictor [10], which analyzes the geometry of the vector field associated to the dynamic $\dot{\mathbf{x}}(t) = \gamma(\mathbf{x}(t))$ of the system without performing any time integration. We propose to use a discretization of the state space using mazes [31]. Mazes correspond to a polygonal decomposition of the state space coupled with an interval enclosure for $\gamma(\mathbf{x})$, which is valid inside the corresponding polygon. This decomposition by mazes can be interpreted as an interval of dynamics with a lower bound and an upper bound. The polygonal representation associated to the maze is a discrete object that can be represented in the memory of the computer by floating point numbers. It will be used to approximate from inside and from outside the sequence of sets that should converge in a finite number of iterations to the invariant set we want to compute. Equivalently, the polygonal representation corresponds to the machine lattice we use in our implementation to represent subsets of $\mathbb{R}^n$.

5.2. Paths

Definition 3. *Given a point* $\mathbf{a}$ *and the system* $\mathcal{S} : \dot{\mathbf{x}}(t) = \gamma(\mathbf{x}(t))$. *The* path *associated to* $\mathbf{a}$ *is defined as*

$$\boldsymbol{\Psi}_\gamma(\mathbf{a}) = \left\{\mathbf{z} | \forall \varepsilon > 0, \exists t_1 > 0, \|\boldsymbol{\varphi}_\gamma(t_1, \mathbf{a}) - \mathbf{z}\| \leq \varepsilon\right\}. \tag{24}$$

Equivalently, we can write

$$\boldsymbol{\Psi}_\gamma(\mathbf{a}) = \boldsymbol{\varphi}_\gamma([0, \infty], \mathbf{a}). \tag{25}$$

The path $\boldsymbol{\Psi}_\gamma(\mathbf{a})$ is a closed set and contains the equilibrium points or cycles to which the system will converge from $\mathbf{a}$.

Definition 4. (Path inside a region). *Consider the system (1), a region* $\mathbb{Y}$, *and a point* $\mathbf{a} \in \mathbb{Y}$. *We define the* path inside $\mathbb{Y}$ *as* $\boldsymbol{\Psi}_{\gamma|\mathbb{Y}}(\mathbf{a})$ *where the function* $\gamma|\mathbb{Y}$ *is defined as*

$$(\gamma|\mathbb{Y})(\mathbf{x}) = \begin{cases} \gamma(\mathbf{x}) & \text{if } \mathbf{x} \in \mathbb{Y} \\ \mathbf{0} & \text{otherwize} \end{cases} \tag{26}$$

This notion is illustrated by Figure 4 where two trajectories starting from $\mathbf{a}$ and $\mathbf{b}$ are represented. The path $\boldsymbol{\Psi}_{\gamma|\mathbb{Y}}(\mathbf{a})$ contains a limit cycle, whereas $\boldsymbol{\Psi}_{\gamma|\mathbb{Y}}(\mathbf{b})$ stops at the boundary of $\mathbb{Y}$.

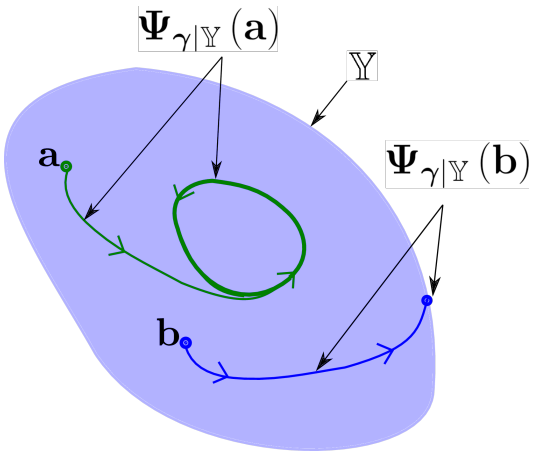

Figure 4. Illustration of the definition of a path inside a region.

5.3. Cover and Automorphism

In this section, we define the automorphism $\overrightarrow{f}$ that we will use to solve our problems involving invariant sets. It will be based on the notion of cover that we now define. A *cover* $\mathcal{P}$ is a collection of open boxes of $\mathbb{R}^n$ whose union is $\mathbb{R}^n$. Denote by $\mathcal{P}(\mathbf{x})$ the union of all boxes of $\mathcal{P}$ containing $\mathbf{x}$.

Proposition 6. *If $\mathbb{A}$ is closed, we have*

$$\mathbf{a} \in Inv^{+}(\mathbb{A}) \Rightarrow \mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{a})}(\mathbf{a}) \subset \mathbb{A}.$$

where $Inv^{+}(\mathbb{A})$ is the largest positive invariant set included in $\mathbb{A}$.

Proof. We have

$$\begin{array}{rcl} \mathbf{a} \in \mathrm{Inv}^{+}(\mathbb{A}) & \Rightarrow & \forall t \geq 0, \boldsymbol{\varphi}_{\gamma}(t, \mathbf{a}) \in \mathbb{A} \\ & \Rightarrow & \forall t \geq 0, \boldsymbol{\varphi}_{\gamma|\mathcal{P}(\mathbf{a})}(t, \mathbf{a}) \in \mathbb{A} \\ & \Rightarrow & \mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{a})}(\mathbf{a}) \subset \mathbb{A} \end{array}$$

□

Theorem 2. *Consider the system $\mathcal{S} : \dot{\mathbf{x}}(t) = \gamma(\mathbf{x}(t))$, a closed set $\mathbb{A}$, and a cover $\mathcal{P}$ of the state space. The set-valued function defined by*

$$\overrightarrow{f}(\mathbb{A}) = \left\{\mathbf{x} | \mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{x})}(\mathbf{x}) \subset \mathbb{A}\right\} \tag{27}$$

is an automorphism. We call $\overrightarrow{f}(\mathbb{A})$ the forward Eulerian predictor, since it predicts where the state will go for one step. Now, the step is not temporal (as for Lagrangian predictors) but spacial and related to the cover $\mathcal{P}$.

Proof. First, note that we have $\overrightarrow{f}(\mathbb{R}^n) = \mathbb{R}^n$. Moreover,

$$\begin{array}{rcl} \overrightarrow{f}(\mathbb{A} \cap \mathbb{B}) & = & \left\{\mathbf{x} | \mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{x})}(\mathbf{x}) \subset \mathbb{A} \cap \mathbb{B}\right\} \\ & = & \left\{\mathbf{x} | \left(\mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{x})}(\mathbf{x}) \subset \mathbb{A}\right)\right. \\ & & \left. \text{and } \left(\mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{x})}(\mathbf{x}) \subset \mathbb{B}\right)\right\} \\ & = & \left\{\mathbf{x} | \mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{x})}(\mathbf{x}) \subset \mathbb{A}\right\} \\ & & \cap \left\{\mathbf{x} | \mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{x})}(\mathbf{x}) \subset \mathbb{B}\right\} \\ & = & \overrightarrow{f}(\mathbb{A}) \cap \overrightarrow{f}(\mathbb{B}). \end{array} \tag{28}$$

□

This is illustrated by Figure 5 in the case where $\gamma(\mathbf{x})$, represented by its blue arrow vector field, is constant and oriented to the right. The polygonal set on the first sub-figure represents $\mathbb{A}$. In this figure, the cover is made with two boxes, which are open and overlap (just a little) on their boundaries. Now, this overlapping is not represented for the sake of clarity.

In the figure $\mathbf{a} \in \mathbb{A}$, $\mathcal{P}(\mathbf{a})$ is made with a single box (the left one). The set $\mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{a})}(\mathbf{a})$ is represented by the blue dotted segment starting from $\mathbf{a}$. Since $\mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{a})}(\mathbf{a})$ is not a subset of $\mathbb{A}$, $\mathbf{a} \notin \overrightarrow{f}(\mathbb{A})$. It means from Proposition 6 that $\mathbf{a} \notin \mathrm{Inv}^{+}(\mathbb{A})$, and this is why it can be removed. We have $\mathbf{b} \in \mathbb{A}$ and $\mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{b})}(\mathbf{b})$ is the blue dotted segment starting from $\mathbf{b}$. Since $\mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{b})}(\mathbf{b}) \subset \mathbb{A}$, we have $\mathbf{b} \in \overrightarrow{f}(\mathbb{A})$. For point $\mathbf{c} \in \mathbb{A}$, the set $\mathcal{P}(\mathbf{c})$ is made with the two boxes, instead of one for $\mathcal{P}(\mathbf{a})$ and $\mathcal{P}(\mathbf{b})$. As a result, $\mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{c})}(\mathbf{c})$, the left dotted green segment is not a subset of $\mathbb{A}$. Thus, $\mathbf{c} \notin \overrightarrow{f}(\mathbb{A})$. Since $\mathbf{c} \in \mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{b})}(\mathbf{b})$ and $\mathbf{c} \notin \overrightarrow{f}(\mathbb{A})$,

we conclude that $\mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{b})}(\mathbf{b})$ is not inside $\mathbb{A} \cap \overrightarrow{f}(\mathbb{A})$. Thus, $\mathbf{b} \notin \mathbb{A} \cap \overrightarrow{f}(\mathbb{A}) \cap \overrightarrow{f}^2(\mathbb{A})$. It is eliminated, since it cannot be an element of $\text{Inv}^+(\mathbb{A})$.

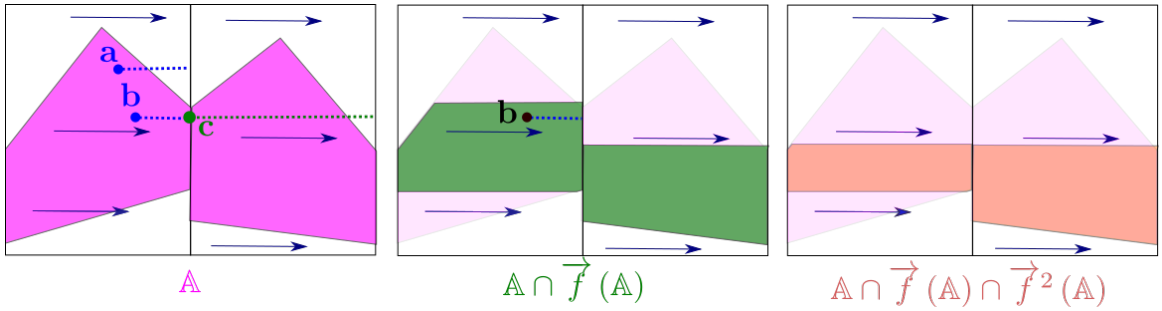

Figure 5. Forward Eulerian predictor $\overrightarrow{f}(\mathbb{A})$ eliminating points that will escape from $\mathbb{A}$.

5.4. Invariant Set

In this section, we now show that the automorphism is linked to invariant sets by a Kleene star operator. More precisely, we will show that $\text{Inv}^+(\mathbb{A}) = \overrightarrow{f}^*(\mathbb{A})$. As a consequence, an interval evaluation of $\overrightarrow{f}^*(\mathbb{A})$ will allow us to have an inner and an outer approximation of an invariant set $\text{Inv}^+(\mathbb{A})$.

Theorem 3. *The set* $\overrightarrow{f}^*(\mathbb{A})$ *contains the greatest positive invariant subset of* $\mathbb{A}$*, i.e.,*

$$Inv^+(\mathbb{A}) \subset \overrightarrow{f}^*(\mathbb{A}). \tag{29}$$

Proof. The proof is by contradiction. Assume that (6) is false. Thus, there exists the largest integer i such that $\text{Inv}^+(\mathbb{A}) \subset \mathbb{B} = (\text{Id} \cap \overrightarrow{f})^i(\mathbb{A})$. Equivalently, there exists $\mathbf{a}$ (see Figure 6) such that

$$\begin{cases} \text{(i)} & \mathbf{a} \in \mathbb{B} \\ \text{(ii)} & \mathbf{a} \notin \left(\text{Id} \cap \overrightarrow{f}\right)(\mathbb{B}) \\ \text{(iii)} & \mathbf{a} \in \text{Inv}^+(\mathbb{B}) \end{cases} \tag{30}$$

now

$$\left(\text{Id} \cap \overrightarrow{f}\right)(\mathbb{B}) = \left\{\mathbf{x} \in \mathbb{B} \,|\, \mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{x})}(\mathbf{x}) \subset \mathbb{B}\right\}, \tag{31}$$

thus

$$\begin{cases} \mathbf{a} \in \text{Inv}^+(\mathbb{B}) & \text{(see (30, iii))} \\ \mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{a})}(\mathbf{a}) \not\subset \mathbb{B} & \text{(since (30, ii))} \end{cases} \tag{32}$$

which is inconsistent with Proposition 6. □

Theorem 4. *We have*

$$\overrightarrow{f}^*(\mathbb{A}) \subset Inv^+(\mathbb{A}). \tag{33}$$

Proof. The proof is by contradiction, i.e., we assume that $\overrightarrow{f}^*(\mathbb{A}) \not\subset \text{Inv}^+(\mathbb{A})$. In such a case, there exists $\mathbf{a} \in \overrightarrow{f}^*(\mathbb{A})$, which is not in $\text{Inv}^+(\mathbb{A})$, as illustrated by Figure 7. For such a point $\mathbf{a}$, the trajectory $\boldsymbol{\varphi}_\gamma(t, \mathbf{a}), t \geq 0$ leaves $\overrightarrow{f}^*(\mathbb{A})$ at the point $\mathbf{b} = \boldsymbol{\varphi}_\gamma(t_b, \mathbf{a})$ for some $t_b > 0$ and then leaves $\mathbb{A}$ at point $\mathbf{d}$. In the figure, the set $\mathbf{\Psi}_{\gamma|\mathcal{P}(\mathbf{b})\cap\mathbb{A}}(\mathbf{b})$ corresponds to the trajectory between $\mathbf{b}$ and $\mathbf{c}$. Therefore, we can write

$$\begin{array}{llll} (i) & \mathbf{b} & \in & \text{boundary}\left(\overrightarrow{f}^*(\mathbb{A})\right) \\ (ii) & \mathbf{b} & \notin & \left(\text{Id} \cap \overrightarrow{f}\right)\left(\overrightarrow{f}^*(\mathbb{A})\right) \end{array}, \tag{34}$$

however, also, there exists a neighborhood $\mathcal{V}_{\mathbf{b}}$ of $\mathbf{b}$ such that

$$\forall \mathbf{b}' \in \mathcal{V}_{\mathbf{b}}, \mathbf{b}' \notin \underbrace{\left(\mathrm{Id} \cap \overrightarrow{f}\right)\left(\overrightarrow{f}^{*}(\mathbb{A})\right)}_{=\overrightarrow{f}^{*}(\mathbb{A})} \tag{35}$$

which is inconsistent with (34, *(i)*). □

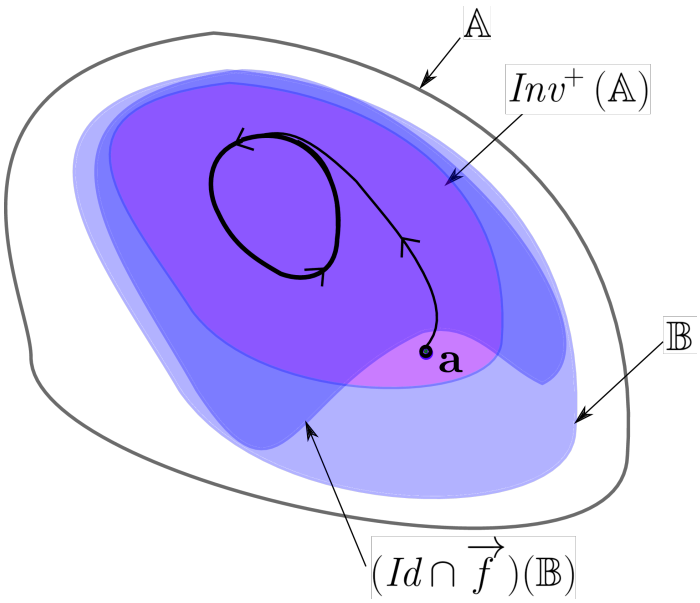

Figure 6. An inconsistent situation that can never occur where $\mathbf{a} \in \mathrm{Inv}^{+}(\mathbb{A})$ but $\mathbf{a} \notin \overrightarrow{f}^{*}(\mathbb{A})$.

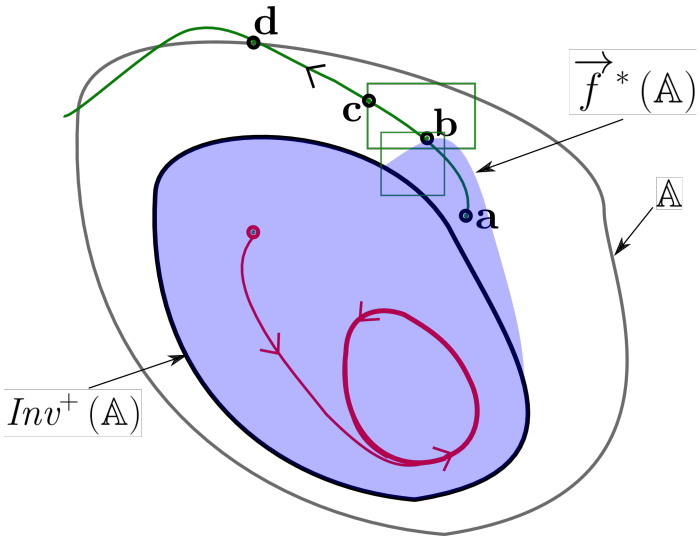

Figure 7. An impossible situation where $\mathbf{a} \notin \mathrm{Inv}^{+}(\mathbb{A})$ but $\mathbf{a} \in \overrightarrow{f}^{*}(\mathbb{A})$ that is used in the proof of Theorem 4.

Corollary 1. *The set-valued function*

$$\overleftarrow{f}(\mathbb{A}) = \left\{\mathbf{x} | \mathbf{\Psi}_{-\gamma|\mathcal{P}(\mathbf{x})}(\mathbf{x}) \subset \mathbb{A}\right\} \tag{36}$$

is an automorphism. Moreover, $\overleftarrow{f}^{*}(\mathbb{A})$ *corresponds to the greatest negative invariant subset of* $\mathbb{A}$*, i.e.,*

$$\overleftarrow{f}^{*}(\mathbb{A}) = Inv^{+}(-\gamma, \mathbb{A}) \overset{(3)}{=} \left\{\mathbf{a} | \forall t \geq 0, \boldsymbol{\varphi}_{\gamma}(-t, \mathbf{a}) \in \mathbb{A}\right\}. \tag{37}$$

6. Test Cases

We consider here several test cases in order to illustrate the principle and the efficiency of our approach. We can note that our method is limited to small-dimension systems because of the exponential complexity of the algorithm w.r.t. the dimension. This is indeed the case for all safe methods dealing with non-convex solution sets.

6.1. Negative Invariant

We consider here a problem treated in [30] involving the Van der Pol system:

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \left(1 - x_1^2\right) \cdot x_2 - x_1. \end{cases} \tag{38}$$

Let us take for initial box $\mathbb{A} = [-3, 3] \times [-3, 3]$. To compute the greatest negative invariant subset $\mathbb{X}$ of $\mathbb{A}$, we compute $\left[\overleftarrow{f}^*\right](\mathbb{A})$, where $\overleftarrow{f}$ is the automorphism defined by 36 and $\left[\overleftarrow{f}^*\right]$ is a machine interval enclosure for $\overleftarrow{f}^*$ (see Equation (21)). The resulting approximation is illustrated by Figure 8, which is obtained in less than 5 s on a standard laptop (all the computations were performed on an Intel i5-3320M@2.6 GHz with 8 GB of RAM). The magenta part corresponds to the inner approximation $\mathbb{X}^-$ of $\mathbb{X}$. From (19), we know that $\mathbb{X}^-$ (magenta) is a negative invariant set. The outer approximation $\mathbb{X}^+$ corresponds to the union of the yellow and the magenta zones. Note that $\mathbb{X}^+$ may not be negative invariant.

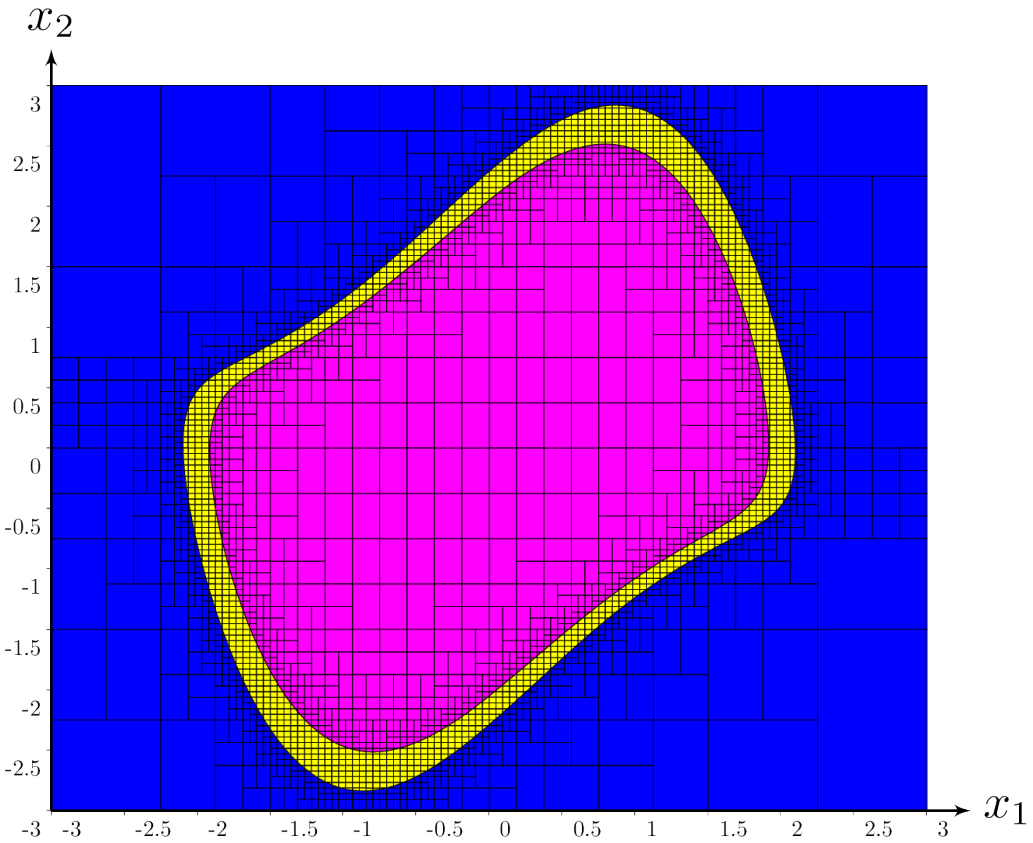

Figure 8. Approximation of the greatest negative invariant set included in the frame box $\mathbb{A}$.

6.2. Forward Reach Set

Given the system (38), the forward reach set [48] of a set $\mathbb{A}$ is defined by

$$\begin{array}{rcl} \text{Fwd}(\mathbb{A}) & = & \overline{\left\{\mathbf{x} \mid \exists t \geq 0, \boldsymbol{\varphi}_\gamma(-t,\mathbf{x}) \in \mathbb{A}\right\}} \\ & = & \overline{\left\{\mathbf{x} \mid \forall t \geq 0, \boldsymbol{\varphi}_\gamma(-t,\mathbf{x}) \in \overline{\mathbb{A}}\right\}} \\ & \overset{(37)}{=} & \overleftarrow{f}^*\left(\overline{\mathbb{A}}\right) \end{array} \tag{39}$$

We obtain in less than 4 s, with $\mathbb{A} = \left\{(x_1,x_2) \in \mathbb{R}^2 \mid (x_1-1.2)^2 + (x_2-1)^2 \leq 0.3^2\right\}$, the approximation illustrated by Figure 9. The frame box is $[-3,3] \times [-3,3]$. Note that we were able to get a non-empty inner approximation of $\text{Fwd}(\mathbb{A})$ that was not possible with existing interval base methods such as [49]. Similar results could have been obtained using flow* [50], but to have the guarantee to enclose the whole trajectory, we need to deal with an infinite horizon, whereas the Taylor-based method (used in Flow*) is devoted to predict the trajectory for a limited time horizon.

6.3. Backward Reach Set

Given a set $\mathbb{A}$, the backward reach set is defined by

$$\begin{array}{rcl} \text{Bwd}(\mathbb{A}) & = & \overline{\left\{\mathbf{x} \mid \exists t \geq 0, \boldsymbol{\varphi}_\gamma(t,\mathbf{x}) \in \mathbb{A}\right\}} \\ & = & \overline{\left\{\mathbf{x} \mid \forall t \geq 0, \boldsymbol{\varphi}_\gamma(t,\mathbf{x}) \in \overline{\mathbb{A}}\right\}} \\ & \overset{(23)}{=} & \overrightarrow{f}^*\left(\overline{\mathbb{A}}\right) \end{array} \tag{40}$$

For the system (38), we get in less than 4 s the approximation of $\text{Bwd}(\mathbb{A})$, as illustrated by Figure 10. The frame box is $[-3,3] \times [-3,3]$.

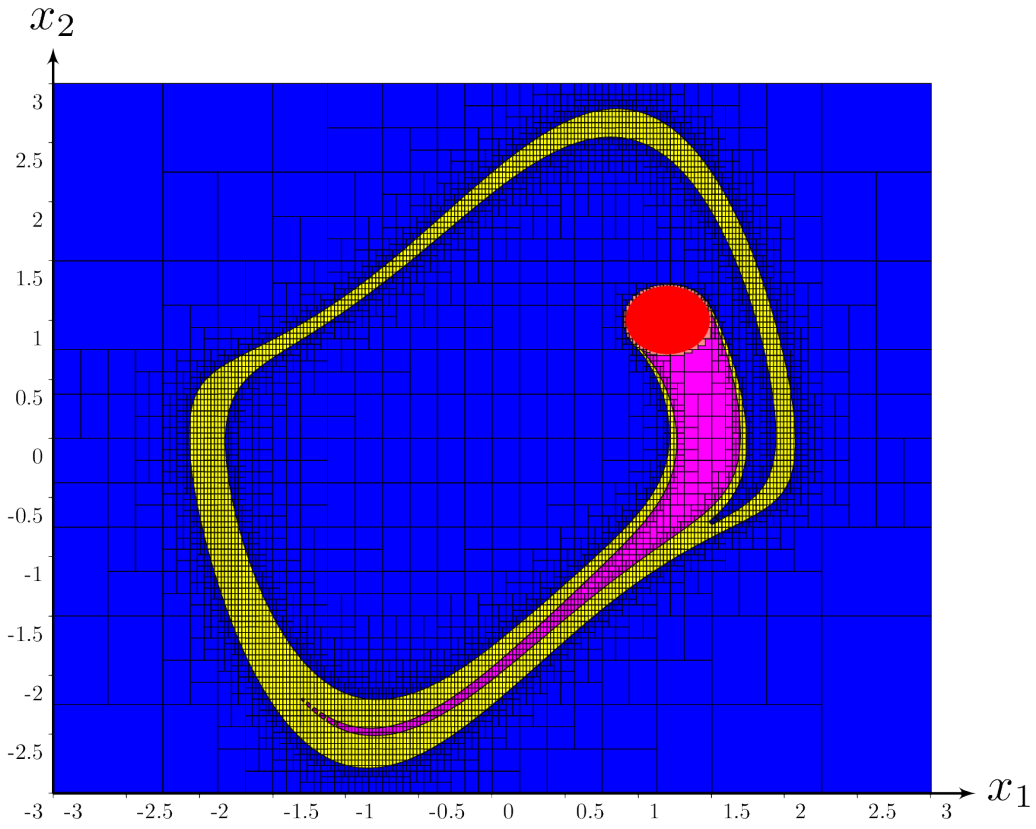

Figure 9. Forward reach set associated with the red disk $\mathbb{A}$ for an infinite time horizon.

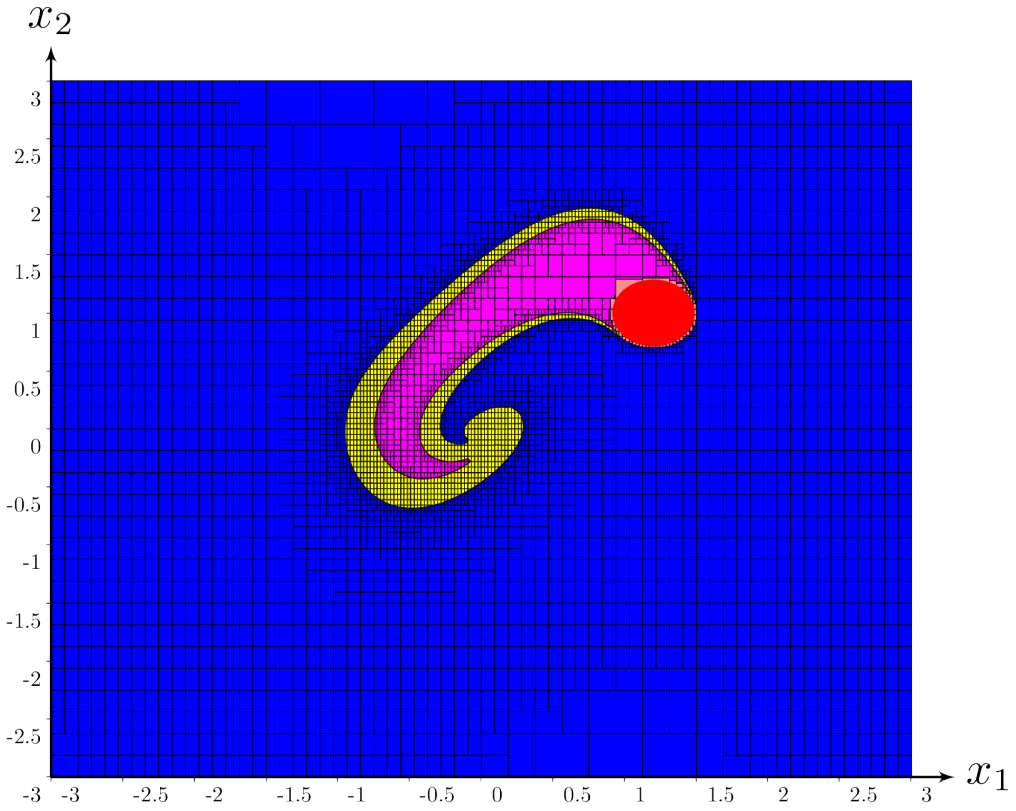

Figure 10. Forward reach set associated with the red disk $\mathbb{A}$.

6.4. Control Forward Reach Set

Consider the system:

$$\mathcal{S} : \dot{\mathbf{x}}(t) = \boldsymbol{\gamma}(\mathbf{x}(t), u),\ u \in \{u_0, u_1\} \tag{41}$$

where u is the control that can be chosen asynchronously inside the set $\{0, 1\}$. Given an initial state set $\mathbb{A}$, we want to compute the set $\mathbb{X}$ of all states that can be reached from $\mathbb{A}$ [51]. We have

$$\mathbb{X} \quad = \quad (\text{Forw}_{u=u_1} \circ \text{Forw}_{u=u_0})^{\infty}(\mathbb{A}) \ . \tag{42}$$

Equivalently, $\mathbb{X}$ can be defined as the limit of the sequence

$$\begin{array}{rcl} \mathbb{X}(k+1) & = & \text{Forw}_{u=u_1}(\text{Forw}_{u=u_0}(\mathbb{X}(k))) \\ \mathbb{X}(0) & = & \mathbb{A} \end{array} , \tag{43}$$

thus,

$$\begin{array}{rcl} \overline{\mathbb{X}}(k+1) & = & \overleftarrow{f_1}^*\left(\overline{\text{Forw}_{u=u_0}(\mathbb{X}(k))}\right) \\ & = & \overleftarrow{f_1}^*\left(\overline{\overleftarrow{f_0}^*\left(\overline{\mathbb{X}(k)}\right)}\right) \\ & = & \overleftarrow{f_1}^* \circ \overleftarrow{f_0}^*\left(\overline{\mathbb{X}(k)}\right). \end{array} \tag{44}$$

Therefore, we have:

$$\overline{\mathbb{X}} = \lim_{k\to\infty} \overline{\mathbb{X}}(k) = \left(\overleftarrow{f_1}^* \circ \overleftarrow{f_0}^*\right)^*\left(\overline{\mathbb{A}}\right) = \left(\overleftarrow{f_1} \circ \overleftarrow{f_0}\right)^*\left(\overline{\mathbb{A}}\right) \tag{45}$$

and finally,

$$\mathbb{X} = \overline{\left(\overleftarrow{f_1} \circ \overleftarrow{f_0}\right)^*\left(\overline{\mathbb{A}}\right)}. \tag{46}$$

We consider as an example the car on the hill system [47] where

$$\begin{cases} \dot{x}_1 = & x_2 \\ \dot{x}_2 = & -9.81\sin(0.55\sin(1.2x_1) - 0.6\sin(1.1x_1)) \\ & -0.7x_2 + u \end{cases}$$

with the set $\mathbb{A} = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 0.5^2\}$ and $u \in \{-1, 1\}$. We get in less than 10 s the approximation illustrated by Figure 11, where the frame box is $[-1, 6] \times [-4, 4]$.

6.5. Minimal Robust Positive Invariant Set

The example is a continuous-time version of the example taken from [52]. We consider the system described by

$$\begin{cases} \dot{x}_1 = & 0.2x_1 + 0.2x_2 - x_1 \\ \dot{x}_2 = & -0.2x_1 + 0.5x_2 + \omega - x_1 \end{cases}$$

where $\omega \in [\omega] = [-1, 1]$ is the perturbation. The system has the form $\dot{\mathbf{x}}(t) = \boldsymbol{\gamma}(\mathbf{x}(t), \omega)$. We want to compute the smallest set $\mathbb{X}$ containing $\mathbf{0}$ such that the system cannot escape. This set corresponds to the *minimal robust positive*ly invariant set [53] and is known to be difficult to compute. Moreover, no method exists in the literature to get a guaranteed inner approximation for nonlinear continuous-time systems. Now, this problem is similar to the previous one except that the control u is now replaced by a perturbation ω, and we can use the same method. In 2 s, we obtain the approximation illustrated by Figure 12 where the frame box is $[-3, 3] \times [-3, 3]$.

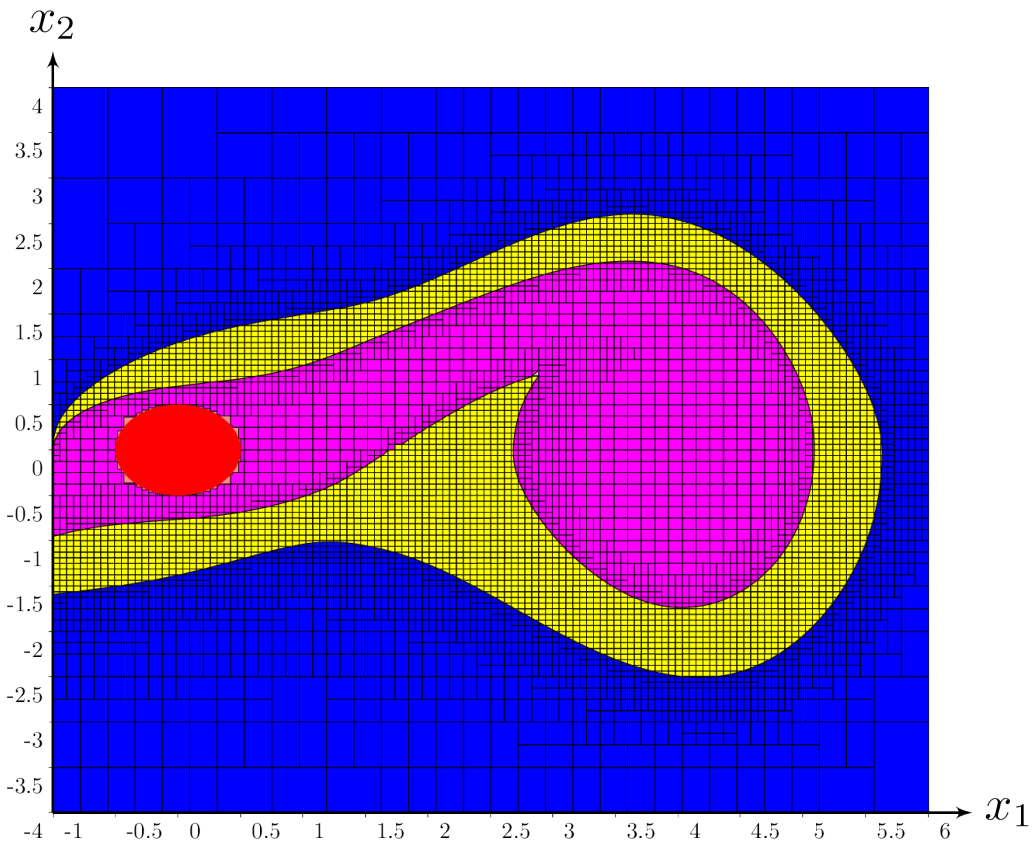

Figure 11. Inner and outer approximation of the control forward reach set.

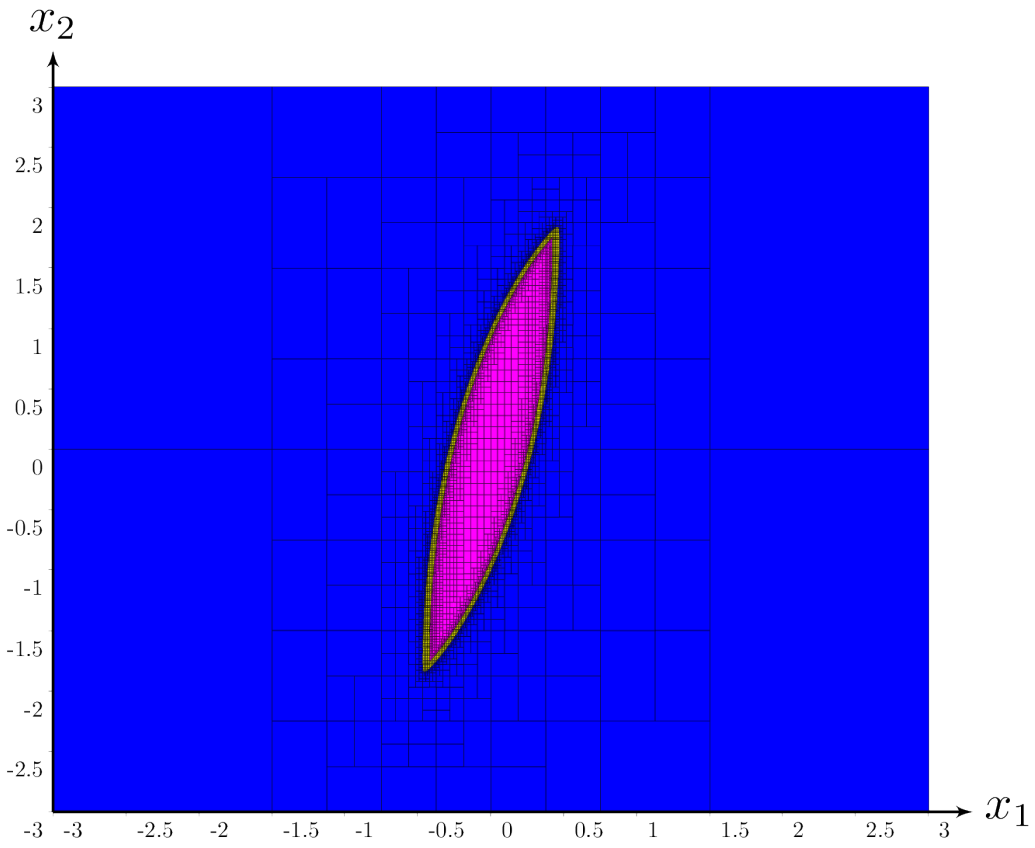

Figure 12. Approximation of the smallest positive invariant set containing **0**.

6.6. Path Planning

Given the system (38) and three sets

$$\begin{cases} \mathbb{A}: & (x_1-0.8)^2+(x_2-1.3)^2 \leq (0.4)^2 \\ \mathbb{B}: & [1.35,1.45]\times[-0.2,0] \\ \mathbb{C}: & [0.74,1.2]\times[-1.5,-1.06] \end{cases}.$$

Let us find the set $\mathbb{X}$ of all points corresponding to a path that starts from $\mathbb{A}$, avoids $\mathbb{B}$, and reaches $\mathbb{C}$. It corresponds to a path planning problem [54,55] for which interval analysis has been shown to be particularly efficient [56–58]. We have

$$\begin{aligned} \mathbb{X} &= \text{Forw}_{\gamma|\mathbb{B}}(\mathbb{A}) \cap \text{Back}_{\gamma|\mathbb{B}}(\mathbb{C}) \\ &= \overleftarrow{f_{\gamma|\mathbb{B}}}^*\left(\overline{\mathbb{A}}\right) \cap \overrightarrow{f_{\gamma|\mathbb{B}}}^*\left(\overline{\mathbb{C}}\right) \end{aligned} \tag{47}$$

and thus our methodology applies. The result, depicted on Figure 13 left, was obtained in less than 38 s for a search box $[-3,3]\times[-3,3]$. The three images on the right show on several zooms around $\mathbb{A}$,$\mathbb{B}$,$\mathbb{C}$ that a non-empty inner approximation was obtained, which was not possible with existing solvers.

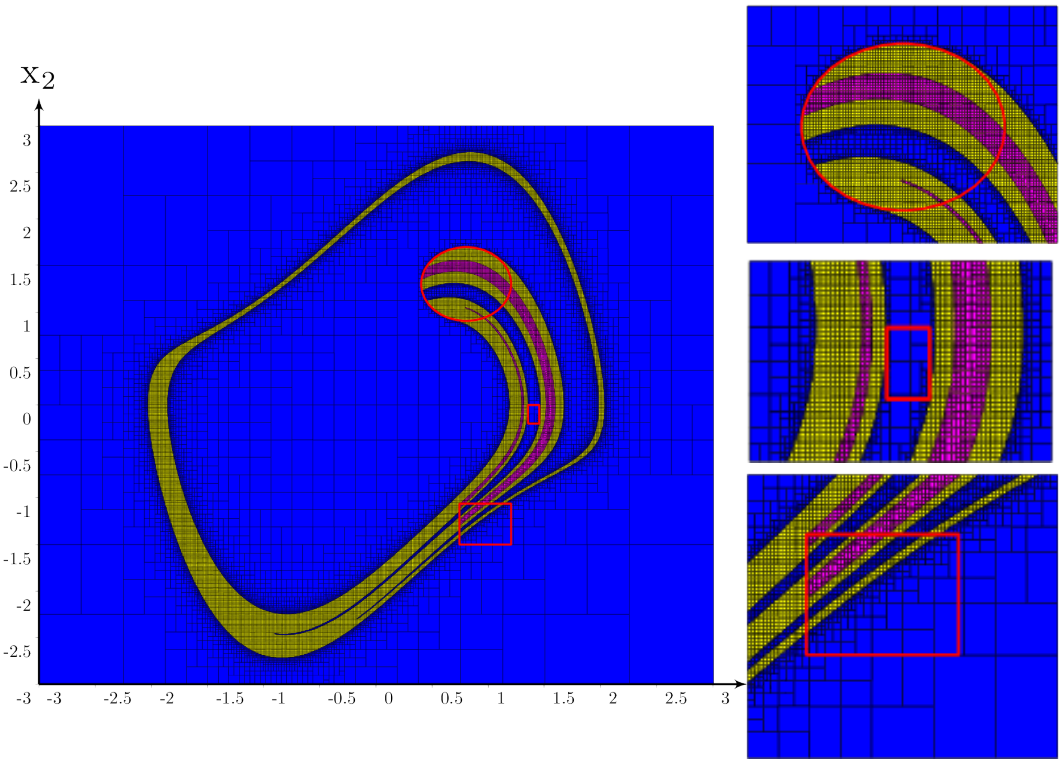

Figure 13. Paths starting from $\mathbb{A}$, avoiding $\mathbb{B}$, and reaching $\mathbb{C}$.

7. Conclusions

In this paper, we have proposed a new approach to compute invariant sets of continuous-time dynamical systems. Our contributions are the following:

- A link between Kleene algebra and invariant sets. This allowed us to derive a simple fixed point method able to compute guaranteed inner and outer approximations of invariant sets.
- The treatment of toy examples for which no other existing approach is able to deal with.

The approach uses the fact that a suited automorphism has been found for a dynamical system described by a deterministic state equation.

Moreover, our formalism allowed us to compute sets that can be defined as combinations (intersection, union, complementary, image by automorphism) of invariant sets. This combination can be interpreted as a first step toward what could be called an *invariant algebra*, i.e., an algebra the atoms of which are positive invariant sets of dynamical systems. This algebra transforms a complex problem such as the *reach and avoid problem* without developing a complex algorithm with properties that are difficult to analyze. Instead, our algebra yields a simple expression operating in our invariant algebra.

Our approach can directly be extended to discrete time system and the algorithm, based on a formal expression, remains unchanged. Only atoms (i.e., the automorphisms) have to be adapted. For a discrete time system of the form $\mathbf{x}(k+1) = \gamma(\mathbf{x}(k))$, the automorphism is even simpler, since it could be $\mathbb{X} \mapsto \gamma(\mathbb{X}) = \{\mathbf{y} | \exists \mathbf{x} \in \mathbb{X}, \mathbf{y} = \gamma(\mathbf{x})\}$. Thus, we get an approach similar to that proposed in [59] where the set invariance is used to prove properties of discrete-time dynamical systems in a context of temporal logic.

In our approach, we have chosen a structure that is a Kleene algebra. It captures many properties we have when we deal with invariant sets. Now, some properties are forgotten by the Kleene algebra. For instance, for our continuous-time systems, we have the property $\overrightarrow{f} \circ \overrightarrow{f} \supset \overrightarrow{f}$. From this property, we may get some other simplifications that could be used to increase the efficiency of the resolution.

The Python code with all examples is made available at (1 March 2022): https://www.ensta-bretagne.fr/lemezo/pyinvariant/pyinvariant.html.

Author Contributions: Writing, L.J., D.M. and T.L.M.; software and examples T.L.M.; review and editing B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the French Government Defense procurement and technology agency, Direction Générale de l'Armement (DGA).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: https://www.ensta-bretagne.fr/lemezo/pyinvariant/pyinvariant.html (accessed on 1 March 2022).

Acknowledgments: This work has been supported by the French Government Defense procurement and technology agency (DGA).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jaulin, L. *Automation for Robotics*; ISTE: Eugene, OR, USA, 2015.
2. Jaulin, L. *Mobile Robotics*; ISTE: Eugene, OR, USA, 2015.
3. Blanchini, F.; Miani, S. *Set-Theoretic Methods in Control*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007.
4. Esik, Z.; Fahrenberg, U.; Legay, A.; Quaas, K. An Algebraic Approach to Energy Problems II, The Algebra of Energy Functions. *Acta Cybern.* **2017**, *23*, 229–268. [CrossRef]
5. Wan, J.; Vehi, J.; Luo, N. A numerical approach to design control invariant sets for constrained nonlinear discrete-time systems with guaranteed optimality. *J. Glob. Optim.* **2009**, *44*, 395–407. [CrossRef]
6. Olaru, S.; Dona, J.D.; Seron, M.; Stoican, F. Positive invariant sets for fault tolerant multisensor control schemes. *Int. J. Control* **2010**, *83*, 2622–2640. [CrossRef]
7. Althoff, D.; Althoff, M.; Scherer, S. Online Safety Verification of Trajectories for Unmanned Flight with Offline Computed Robust Invariant Sets. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany, 28 September–2 October 2015.
8. Asarin, E.; Dang, T.; Maler, O. The d/dt tool for verification of hybrid systems. In Proceedings of the International Conference on Computer Aided Verification, Los Angeles, CA, USA, 21–24 July 2002; pp. 365–370.
9. Tabuada, P. *Verification and Control of Hybrid Systems: A Symbolic Approach*; Springer: Berlin/Heidelberg, Germany, 2009.
10. Mitchell, I. Comparing forward and backward reachability as tools for safety analysis. In *Hybrid Systems: Computation and Control*; Bemporad, A., Bicchi, A., Buttazzo, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 428–443.
11. Tahir, F.; Jaimoukha, M. Low-Complexity Polytopic Invariant Sets for Linear Systems Subject to Norm-Bounded Uncertainty. *IEEE Trans. Autom. Control* **2015**, *60*, 1416–1421. [CrossRef]
12. Roux, P.; Jobredeaux, R.; Garoche, P.; Feron, E. A generic ellipsoid abstract domain for linear time invariant systems. In Proceedings of the Hybrid Systems: Computation and Control, HSCC'12, Beijing, China, 17–19 April 2012; pp. 105–114.
13. Bertsekas, D. Infinite-time reachability of state-space regions by using feedback Control. *IEEE Trans. Autom. Control* **1972**, *17*, 604–613. [CrossRef]
14. Athanasopoulos, N.; Smpoukis, K.; Jungers, R. Invariant Sets Analysis for Constrained Switching Systems. *IEEE Control Syst. Lett.* **2017**, *1*, 256–261. [CrossRef]
15. Allamigeon, X.; Gaubert, S.; Goubault, E.; Putot, S.; Stott, N. A fast method to compute disjunctive quadratic invariants of numerical programs. *Acm Trans. Embed. Comput. Syst.* **2017**, *6*, 1–19. [CrossRef]
16. Guernic, C.L.; Girard, A. Reachability analysis of linear systems using support functions. *Nonlinear Anal.* **2009**, *4*, 250–262. [CrossRef]
17. Ghorbal, K.; E.Goubault.; Putot, S. A logical product approach to zonotope intersection. In *Computer Aided Verification, CAV 2010*; Touili, T., Cook, B., Jackson, P., Eds.; Springer: Edinburgh, UK, 2010.
18. Asarin, E.; Dang, T.; Girard, A. Hybridization methods for the analysis of non-linear systems. *Acta Inform.* **2007**, *7*, 451–476. [CrossRef]
19. Girard, A. Computation and stability analysis of limit cycles in piecewise linear hybrid systems. In Proceedings of the 1st IFAC Conference on Analysis and Design of Hybrid Systems, St Malo, France, 16–18 June 2003; pp. 181–186.
20. Sankaranarayanan, S.; Dang, T.; Ivancic, F. Symbolic model checking of hybrid systems using template polyhedra. *Tacas Lect. Notes Comput. Sci.* **2008**, *4963*, 188–202.
21. Saint-Pierre, P. Approximation of the viability kernel. *Appl. Math. Optim.* **1994**, *29*, 187–209. [CrossRef]
22. Bobiti, R.; Lazar, M. Automated-Sampling-Based Stability Verification and DOA Estimation for Nonlinear Systems. *IEEE Trans. Autom. Control* **2018**, *63*, 3659–3674. [CrossRef]
23. Goubault, E.; Mullier, O.; Putot, S.; Kieffer, M. Inner approximated reachability analysis. In Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control, HSCC'14, Berlin, Germany, 15–17 April 2014; pp. 163–172.
24. Wilczak, D.; Zgliczynski, P. Cr-Lohner algorithm. *Schedae Inform.* **2011**, *20*, 9–46.

25. Goubault, E.; Putot, S. Forward Inner-Approximated Reachability of Non-Linear Continuous Systems. In Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control, Pittsburgh, PA, USA, 18–20 April 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 1–10. [CrossRef]
26. Giesl, P.; Hafstein, S. Review on computational methods for Lyapunov functions. *Discret. Contin. Dyn. Syst.* **2015**, *8*, 2291–2331.
27. Henrion, D.; Korda, M. Convex computation of the region of attraction of polynomial control systems. *IEEE Trans. Autom. Control* **2013**, *59*, 297–312. [CrossRef]
28. Ratschan, S.; She, Z. Providing a Basin of Attraction to a Target Region of Polynomial Systems by Computation of Lyapunov-like Functions . *SIAM J. Control Optim.* **2010**, *48*, 4377–4394. [CrossRef]
29. Oustry, A.; Tacchi, M.; Henrion, D. Inner approximations of the maximal positively invariant set for polynomial dynamical systems. *IEEE Control Syst. Lett.* **2019**, *3*, 733–738. [CrossRef]
30. Boczko, E.; Kalies, W.; Mischaikow, K. Polygonal approximation of flows. *Topol. Its Appl.* **2007**, *154*, 2501–2520. [CrossRef]
31. Mézo, T.L.; Jaulin, L.; Zerr, B. An interval approach to compute invariant sets. *IEEE Trans. Autom. Control* **2017**, *62*, 4236–4243. [CrossRef]
32. Conley, C. *Isolated Invariant Sets and the Morse Index*; American Mathematical Society: Providence, RI, USA, 1991.
33. Frehse, G. PHAVer: Algorithmic Verification of Hybrid Systems. *Int. J. Softw. Tools Technol. Transf.* **2008**, *10*, 23–48. [CrossRef]
34. Frehse, G.; Le Guernic, C.; Donzé, A.; Cotton, S.; Ray, R.; Lebeltel, O.; Ripado, R.; Girard, A.; Dang, T.; Maler, O. SpaceEx: Scalable Verification of Hybrid Systems. In Proceedings of the 23rd International Conference on Computer Aided Verification (CAV), Snowbird, UT, USA, 14–20 July 2011.
35. LaValle, S. *Planning Algorithm*; Cambridge University Press: Cambridge, UK, 2006.
36. Kalies, W.; Mischaikow, K.; Vandervorst, R. Lattice Structures for Attractors. *Found. Comput. Math.* **2016**, *16*, 1151–1191. [CrossRef]
37. Davey, B.A.; Priestley, H.A. *Introduction to Lattices and Order*; Cambridge University Press: Cambridge, UK, 2002; ISBN 0521784514.
38. Monniaux, D. The pitfalls of verifying floating-point computations. *Acm Trans. Program. Lang. Syst.* **2008**, *30*, 1–12. [CrossRef]
39. Mézo, T.L.; Jaulin, L.; Zerr, B. Bracketing the solutions of an ordinary differential equation with uncertain initial conditions. *Appl. Math. Comput.* **2018**, *318*, 70–79.
40. Mézo, T.L.; Jaulin, L.; Zerr, B. Bracketing backward reach sets of a dynamical system. *Int. J. Control* **2019**, *93*, 2528–2540. [CrossRef]
41. Kozen, D. On Kleene algebras and closed semirings. In *International Symposium on Mathematical Foundations of Computer Science*; Springer: Berlin/Heidelberg, Germany, 1990; pp. 26–47.
42. Kozen, D. A completeness theorem for Kleene algebras and the algebra of regular events. In Proceedings of the Sixth Annual IEEE Symposium on Logic in Computer Science, LICS, Amsterdam, The Netherlands, 15–18 July 1991; pp. 214–225.
43. Moore, R.E. *Interval Analysis*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1966.
44. Lhommeau, M.; Hardouin, L.; Cottenceau, B.; Jaulin, L. Interval Analysis and Dioid: Application to Robust Controller Design for Timed Event Graphs. *Automatica* **2004**, *40*, 1923–1930. [CrossRef]
45. Rohou, S.; Jaulin, L.; Mihaylova, M.; Bars, F.L.; Veres, S. Guaranteed Computation of Robots Trajectories. *Robot. Auton. Syst.* **2017**, *93*, 76–84. [CrossRef]
46. Monnet, D.; Ninin, J.; Jaulin, L. Computing an inner and an outer approximation of the viability kernel. *Reliab. Comput.* **2016**, *22*, 138–148.
47. Lhommeau, M.; Jaulin, L.; Hardouin, L. Capture Basin Approximation using Interval Analysis. *Int. J. Adapt. Control Signal Process.* **2011**, *25*, 264–272. [CrossRef]
48. Dang, T.; Guernic, C.L.; Maler, O. Computing reachable states for nonlinear biological models. *Comput. Methods Syst. Biol.* **2009**, *5688*, 126–141.
49. Ramdani, N.; Nedialkov, N. Computing Reachable Sets for Uncertain Nonlinear Hybrid Systems using Interval Constraint Propagation Techniques. *Nonlinear Anal. Hybrid Syst.* **2011**, *5*, 149–162. [CrossRef]
50. Chen, X. Reachability Analysis of Non-LinearHybrid Systems Using Taylor Models. Ph.D. Thesis, University of Aachen, Aachen, Germany, 2015.
51. Girard, A.; Guernic, C.L.; Maler, O. Efficient computation of reachable sets of linear time-invariant systems with inputs. *Hybrid Syst. Comput. Control* **2006**, *3927*, 257–271.
52. Meslem, N.; Loukkas, N.; Martinez, J. Using set invariance to design robust interval observers for discrete time linear systems. *Int. J. Robust Nonlinear Control* **2018**, *28*, 3623–3639. [CrossRef]
53. Rakovic, S.V.; Kerrigan, E.C.; Kouramas, K.I.; Mayne, D.Q. Invariant approximations of the minimal robust positively invariant set. *IEEE Trans. Autom. Control* **2005**, *50*, 406–410. [CrossRef]
54. Lozano-Pérez, T. Spatial Planning: A Configuration Space Approach. *IEEE Trans. Comput.* **1983**, *32*, 108–120. [CrossRef]
55. Rungger, M.; Zamani, M. SCOTS: A Tool for the Synthesis of Symbolic Controllers. In Proceedings of the HSCC 2016, Vienna, Austria, 12–14 April 2016. [CrossRef]
56. Jaulin, L. Path Planning Using Intervals and Graphs. *Reliab. Comput.* **2001**, *7*, 1–15. [CrossRef]
57. Pepy, R.; Kieffer, M.; Walter, E. Reliable Robust Path Planning with Application to Mobile Robots. *Int. J. Appl. Math. Comput. Sci.* **2009**, *19*, 413–424. [CrossRef]

58. Crépon, P.; Panchea, A.; Chapoutot, A. Reliable Motion Plannning for a Mobile Robot. In Proceedings of the IEEE International Conference on Robotic Computing, Laguna Hills, CA, USA, 31 January–2 February 2018.
59. Belta, C.; Yordanov, B.; Gol, E. *Formal Methods for Discrete-Time Dynamical Systems*; Studies in Systems, Decision and Control; Springer International Publishing: Berlin/Heidelberg, Germany, 2017.