Forward Inner-Approximated Reachability of Non-Linear Continuous Systems

Eric Goubault
LIX, Ecole Polytechnique, CNRS,
Université Paris-Saclay, 91128 Palaiseau,
France
goubault@lix.polytechnique.fr

Sylvie Putot
LIX, Ecole Polytechnique, CNRS,
Université Paris-Saclay, 91128 Palaiseau,
France
putot@lix.polytechnique.fr

ABSTRACT

We propose an approach for computing inner-approximations (also called under-approximations) of reachable sets of dynamical systems defined by non-linear, uncertain, ordinary differential equations. This is a notoriously difficult problem, much more intricate than outer-approximations (also called over-approximations), for which there exist well known solutions, mostly based on Taylor models. The few methods developed recently for inner-approximation mostly rely on backward flowmaps, and extra ingredients, either coming from optimization, or involving topological criteria, are required. Our solution, in comparison, builds on rather inexpensive set-based methods, namely a generalized mean-value theorem combined with Taylor models outer-approximations of the flow and its Jacobian with respect to the uncertain inputs and parameters. We demonstrate with a C/C++ prototype implementation that our method is both efficient and precise on classical examples. The combination of such forward inner and outer Taylor-model based approximations can be used as a basis for the verification and falsification of properties of cyber-physical systems.

Keywords

Inner-approximation; Taylor models; affine arithmetic; modal intervals; reachability

1. INTRODUCTION

We propose an approach to compute inner-approximating flowpipes, that under-approximate reachable sets of uncertain continuous systems described by ordinary differential equations, which are widely used for modeling all sorts of physical, biological and even economic or social systems.

While outer-approximations describe states that may be reached, inner-approximations represent states that are actually reachable from one of the initial states. They are thus a very useful complement to the more classical outer-approximations, since they allow to show that some executions of the system will actually reach a target or a bad state. Also, the combination of inner and outer-approximations allows for judging the quality of the abstractions involved. But methods for inner-approximated reachability are far less developed, and especially in the non-linear case, since most methods in the non-linear case rely on conservative linearizations, which necessarily produce outer approximations.

In this work, we concentrate on the inner-approximation of the reachable sets of the continuous part of hybrid systems. We consider general systems of parametric ODEs, i.e. possibly non-linear, or even non-polynomial, of the form :

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

where the continuous variables x belong to a state-space domain $\mathcal{D} \subseteq \mathbb{R}^n$, the (constant) parameters p belong to the uncertainty domain $\mathcal{P} \subseteq \mathbb{R}^p$, and $f : \mathcal{D} \times \mathcal{P} \times \mathbb{R}^+ \to \mathcal{D}$ is assumed sufficiently smooth on $\mathcal{D} \subseteq \mathbb{R}^n$ (at least $\mathcal{C}^1$, and sometimes more when we will use higher Taylor models, see Section 2.4).

Introducing the new state variable $z = (x, p, t)$ with $\dot{z} = (\dot{x}, 0, 1)$, and defining $\mathcal{Z} = \mathcal{D} \times \mathcal{P} \times \mathbb{R}^+$, the equation (1) can be rewritten with all uncertainties embedded in the initial state vector :

$$\dot{z}(t) = f(z) \tag{2}$$

In the sequel, we will write x_i and f_i for the ith component $(i = 1, \ldots, n)$ of the state vector x and of the function f.

Contributions:.

This paper extends the work of [13, 14], where were proposed an approach for direct forward inner-approximated reachability of discrete dynamical systems, and a few hints to handle continuous and hybrid systems. We are computing here inner-approximations of the flow of uncertain initial value problems, as defined in Section 2. There are two main ingredients involved in our method. The first ingredient is that in order to derive inner-approximations, we only need forward outer-approximations of some dynamics, which we can compute using classical Taylor models, introduced in Section 2.4. But we need to outer-approximate not only the set of reachable states of the dynamics but also of the variational equations, including the dynamics of the Jacobian of the flow with respect to the uncertain initial values.

The second main ingredient of our method is a generalized mean value theorem, that we introduce in Section 2.3, applied to the flow of the uncertain ODE. The generalized mean value theorem relies itself on modal intervals, a simple extension of classical interval arithmetic (see Section 2.2), to inner-approximate the image of an input set by a

HSCC'17, April 18 - 20, 2017, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-4590-3/17/04…$15.00

DOI: http://dx.doi.org/10.1145/3049797.3049811

non-linear vector-valued function. Note that (Kaucher) interval methods, known to be conservative, are locally used to derive inner ranges, but that this conservatism does not propagate: the inner range is always computed from the outer-approximated Taylor models.

In many ways, all this is remarkably simple, with respect to other existing methods (using backward propagation of the flow of the dynamics), that we discuss in the paragraph devoted to related work. Our method is not much more complex than a classical Taylor model approach for outer-approximations. But we have to consider a larger dynamical system, since we have to consider also the dynamics of the Jacobian with respect to the initial values. Thus, if the original system contains n equations, we must compute Taylor models for n^2 equations. However, the Taylor coefficients of the Jacobian can easily be derived from the Taylor models of the original equations, as we show in Section 3.

Finally, we carry out some experiments with our prototype implementation and provide comparisons to existing work.

Related work:.

There are numerous methods for the computation of outer-approximating (or over-approximating) flowpipes and reachable sets of ODEs, either linear [9], or non-linear [6, 25, 26], linear in the presence of uncertain parameters [8] or non-linear with uncertain parameters [1]. This is also supported by several tools, that can often also consider hybrid systems, see e.g. NLToolBox [27], SpaceEx [7], Flow* [3], CORA [2] and VNODE-LP [24] to mention but a few.

Inner-approximations have been far less studied, except in the case of linear systems, see e.g. [9], or using ellipsoidal methods [21] (and the corresponding tool [20]). In [18], the authors compute inner-approximations of the viability kernel by iterating backward (inner-approximated) reachability problems, using the ellipsoidal methods of [21]. The methods we propose here for inner-approximating reachable sets can be used for inner-approximating viability kernels as well.

The main existing method for under-approximating (or inner-approximating as we put it here) flowpipes of non-linear systems is a backward method, described in [4]. The method starts with a general compact and connected set of states X_0 described by a system of polynomial inequalities, and constructs a Taylor model for the backward flowmap Φ of the dynamics. Then, any *connected* set Ω which contains a point x which is mapped by Φ into X_0 is an inner-approximation of the reachable set of states X if Ω does not intersect the boundary of X. The method of [4] relies then on two computational ingredients. First, it builds a Taylor model for the backward flowmap (it is of the same order of complexity as for any forward outer-approximation, or for our inner-approximation method). Then, a candidate inner-approximation Ω that does not intersect the boundary of X is given by a set of polynomial constraints, derived from the Taylor model for the backward flowmap, and the constraints defining the initial set of states X_0. The method of [4] has then to test connectedness, which is intractable in general but can be semi-decided using clever interval methods.

A similar backward approach has been proposed in [28]. It is similar in that it also constructs an outer-approximation of the backward flowmap. But their authors construct an outer-approximation of the boundary of the reachable set to find inner-approximations. This is done using interval methods and a careful subdivision of the state-space, which might be very costly given that the boundary of the reachable set of highly non-linear ODEs might be extremely complicated to approximate.

Finally, the authors have recently discovered Section 4 of the work [11], which contains ideas that look similar to ours. The main differences seem to be that we are considering more general parameterized dynamical systems, which will later allow us to handle guard conditions for hybrid systems, and that we have a different scheme for bounding the remainder in our inner-approximated Taylor models. But we could not assess the practical differences since the description in [11] is sketchy, and contains no real experiment.

2. PRELIMINARIES

We introduce here the main ingredients used in our approach. Section 2.2 is devoted to generalized intervals and Kaucher arithmetic, which are instrumental in extending the mean-value theorem to obtain an inner-approximation of the range of a function over interval inputs, as described in Section 2.3. Finally briefly introduce Taylor methods for enclosing flows of ODEs in Section 2.4.

2.1 Outer and inner interval approximations

Classical intervals [23] are used in many situations to rigorously compute with interval domains instead of reals, usually leading to outer approximations of function ranges over boxes. We denote the set of classical intervals by $\mathbb{IR} = \{[\underline{x}, \overline{x}],\ \underline{x} \in \mathbb{R}, \overline{x} \in \mathbb{R}, \underline{x} \leqslant \overline{x}\}$

In what follows, uncertain quantities defined in intervals (inputs) are noted in bold, outer-approximating interval enclosures are noted in bold and enclosed within inward facing brackets, and inner-approximating intervals are noted in bold and enclosed within outward facing brackets.

An outer-approximating extension of a function $f : \mathbb{R}^n \to \mathbb{R}$ is a function $[\boldsymbol{f}] : \mathbb{IR}^n \to \mathbb{IR}$ such that for all $\boldsymbol{x}$ in $\mathbb{IR}^n, \text{range}(f, \boldsymbol{x}) = \{f(x), x \in \boldsymbol{x}\} \subseteq [\boldsymbol{f}](\boldsymbol{x})$. The natural interval extension consists in replacing real operations by their interval counterparts in the expression of the function. A generally more accurate extension relies on the mean-value theorem, linearizing the function to compute. Suppose the function f is differentiable over the interval $\boldsymbol{x} = [a, b]$. Then, the mean-value theorem implies that for any choice of $x_0 \in \boldsymbol{x}$, then we have $\forall x \in \boldsymbol{x},\ \exists c \in \boldsymbol{x},\ f(x) = f(x_0) + f'(c)(x - x_0)$.

If we can bound the range of the gradient of f over $\boldsymbol{x}$, by $\text{range}(f', \boldsymbol{x}) \subseteq [\boldsymbol{f}'](\boldsymbol{x})$, then we can derive the following interval enclosure, usually called the mean-value extension: for any $x_0 \in \boldsymbol{x}$, $\text{range}(f, \boldsymbol{x}) \subseteq f(x_0) + [\boldsymbol{f}'](\boldsymbol{x})(\boldsymbol{x} - x_0)$.

Classical interval computations can be interpreted as quantified propositions. Consider for example $f(x) = x^2 - x$, its exact range over $x = [2, 3]$ is $[2, 6]$. The natural interval extension of f, evaluated on $[2, 3]$, is $[\boldsymbol{f}]([2, 3]) = [2, 3]^2 - [2, 3] = [1, 7]$, which can be interpreted as the proposition $(\forall x \in [2, 3])\,(\exists z \in [1, 7])\,(f(x) = z)$. The mean-value extension gives $f(2.5) + [\boldsymbol{f}']([2, 3]) \times ([2, 3] - 2.5) = [1.25, 6.25]$, and can be interpreted similarly as an outer-aproximation.

Inner-approximations determine a set of values proved to belong to the range of the function over some input box. The fact that some $]\boldsymbol{z}[\in \mathbb{IR}$ satisfies $]\boldsymbol{z}[\subseteq \text{range}(f, \boldsymbol{x})$, i.e., is an inner-approximation of the range of f over $\boldsymbol{x}$, can again be written using quantifiers : $(\forall z \in]\boldsymbol{z}[)\,(\exists x \in \boldsymbol{x})\,(f(x) = z)$.

2.2 Modal intervals and Kaucher arithmetic

The results and notations introduced in this section are mostly based on the work of Goldsztejn *et al.* on modal intervals [10]. Let us first introduce generalized intervals, i.e., intervals whose bounds are not ordered, and Kaucher arithmetic [17] on these intervals.

The set of generalized intervals is denoted by $\mathbb{IK} = \{\boldsymbol{x} = [\underline{x}, \overline{x}],\ \underline{x} \in \mathbb{R}, \overline{x} \in \mathbb{R}\}$. Related to a set of real numbers $\{x_0 \in \mathbb{R},\ \underline{x} \leqslant x_0 \leqslant \overline{x}\}$, one can consider two generalized intervals, $[\underline{x}, \overline{x}]$, which is called *proper*, and $[\overline{x}, \underline{x}]$, which is called *improper*. We define the operations dual $[a, b] = [b, a]$ and pro $[a, b] = [\min(a, b), \max(a, b)]$.

DEFINITION 1 ([10]). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function and $\boldsymbol{x} \in \mathbb{IK}^n$, which we can decompose in $\boldsymbol{x}_\mathcal{A} \in \mathbb{IR}^p$ and $\boldsymbol{x}_\mathcal{E} \in (\text{dual } \mathbb{IR})^q$ with $p + q = n$. A generalized interval $\boldsymbol{z} \in \mathbb{IK}$ is $(f, \boldsymbol{x})$-interpretable if*

$$(\forall x_\mathcal{A} \in \boldsymbol{x}_\mathcal{A})\,(Q_z z \in \textit{pro } \boldsymbol{z})\,(\exists x_\mathcal{E} \in \textit{pro } \boldsymbol{x}_\mathcal{E}), (f(x) = z) \quad (3)$$

where $Q_z = \exists$ if $(\boldsymbol{z})$ is proper, and $Q_z = \forall$ otherwise.

When all intervals in (3) are proper, we retrieve the interpretation of classical interval computation, which gives an outer approximation of range$(f, \boldsymbol{x})$, that is $(\forall x \in \boldsymbol{x})\,(\exists z \in [\boldsymbol{z}])\,(f(x) = z)$. When all intervals are improper, (3) becomes an inner-approximation of range$(f, \boldsymbol{x})$, that is $(\forall z \in]\text{pro } \boldsymbol{z}[)\,(\exists x \in \text{pro } \boldsymbol{x})\,(f(x) = z)$.

Kaucher arithmetic [17] returns intervals that are interpretable as inner-approximations in some simple cases. Kaucher addition extends addition on classical intervals by $\boldsymbol{x} + \boldsymbol{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$ and $\boldsymbol{x} - \boldsymbol{y} = [\underline{x} - \overline{y}, \overline{x} - \underline{y}]$.

For multiplication, let us decompose $\mathbb{IK}$ in $\mathcal{P} = \{\boldsymbol{x} = [\underline{x}, \overline{x}],\ \underline{x} \geqslant 0 \wedge \overline{x} \geqslant 0\}$, $-\mathcal{P} = \{\boldsymbol{x} = [\underline{x}, \overline{x}],\ \underline{x} \leqslant 0 \wedge \overline{x} \leqslant 0\}$, $\mathcal{Z} = \{\boldsymbol{x} = [\underline{x}, \overline{x}],\ \underline{x} \leqslant 0 \leqslant \overline{x}\}$, and dual $\mathcal{Z} = \{\boldsymbol{x} = [\underline{x}, \overline{x}],\ \underline{x} \geqslant 0 \geqslant \overline{x}\}$. When restricted to proper intervals, the Kaucher multiplication coincides with the classical interval multiplication. Kaucher multiplication $\boldsymbol{x} \times \boldsymbol{y}$ extends the classical multiplication for all possible combinations of $\boldsymbol{x}$ and $\boldsymbol{y}$ belonging to these sets. We refer to [17] for full details, and only give below an intuitive explanation of one of these cases. Let us interpret the result of the multiplication $\boldsymbol{z} = \boldsymbol{x} \times \boldsymbol{y}$ when $\boldsymbol{y} \in$ dual $\mathcal{Z}$, and $\boldsymbol{x} \in \mathcal{Z}$, which is $\boldsymbol{z} = \boldsymbol{x} \times \boldsymbol{y} = 0$. Proposition 1 will express the fact that the result can be interpreted as in Definition 1. Interval $\boldsymbol{z}$ can a priori either be proper or improper, let us consider the improper case. We obtain an inner-approximation of the range of the multiplication: according to the quantifiers in Definition 1, computing $\boldsymbol{z} = \boldsymbol{x} \times \boldsymbol{y}$ consists in finding $\boldsymbol{z}$ such that for all $x \in \boldsymbol{x}$, for all $z \in$ pro $\boldsymbol{z}$, there exists $y \in$ pro $\boldsymbol{y}$ such that $z = x \times y$. If $\boldsymbol{x}$ contains zero, which is the case when $\boldsymbol{x} \in \mathcal{Z}$, then $\boldsymbol{z}$ is necessarily 0. Indeed, a property that holds for all $x \in \boldsymbol{x}$, holds in particular for $x = 0$, from which we deduce that for all $z \in$ pro $\boldsymbol{z}$, (there exists $y \in$ pro $\boldsymbol{y}$) $z = 0$.

The important feature of Kaucher arithmetic is that it defines a generalized interval natural extension (see [10]) :

PROPOSITION 1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function, given by an arithmetic expression where each variable appears syntactically only once (and with dregree 1). Then for $\boldsymbol{x} \in \mathbb{IK}^n$, $f(\boldsymbol{x})$, computed using Kaucher arithmetic, is $(f, \boldsymbol{x})$-interpretable.*

Kaucher arithmetic can thus be used in some cases to compute an inner-approximation of range$(f, \boldsymbol{x})$. But the restriction to functions f with single occurrences of variables, that is with no dependency, prevents its direct use. A mean-value extension allows us to overcome this limitation.

2.3 Generalized interval mean value extension

In the general case of a differentiable function f, the mean-value theorem can be extended to define a generalized interval mean value extension (see [10]) :

THEOREM 1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable, $\boldsymbol{x} \in \mathbb{IK}^n$ an improper interval, and suppose that for each $i \in \{1, \ldots, n\}$, we can compute $[\boldsymbol{\Delta}_i] \in \mathbb{IR}$ such that*

$$\left\{ \frac{\partial f}{\partial x_i}(x),\ x \in \textit{pro } \boldsymbol{x} \right\} \subseteq [\boldsymbol{\Delta}_i]. \quad (4)$$

Then, for any $\tilde{x} \in$ pro $\boldsymbol{x}$, the following interval, evaluated with Kaucher arithmetic, is $(f, \boldsymbol{x})$-interpretable :

$$\tilde{f}(\boldsymbol{x}) = f(\tilde{x}) + \sum_{i=1}^{n} [\boldsymbol{\Delta}_i](\boldsymbol{x}_i - \tilde{x}_i). \quad (5)$$

When using (5) for inner-approximation, we can only get a subset of all possible cases in the Kaucher multiplication, we list them and the corresponding multiplication rules below: $(\boldsymbol{x} \in \mathcal{P}) \times (\boldsymbol{y} \in \text{dual } \mathcal{Z}) = [\underline{x}\underline{y}, \overline{x}\overline{y}]$, $(\boldsymbol{x} \in -\mathcal{P}) \times (\boldsymbol{y} \in \text{dual } \mathcal{Z}) = [\overline{x}\underline{y}, \underline{x}\overline{y}]$, and $(\boldsymbol{x} \in \mathcal{Z}) \times (\boldsymbol{y} \in \text{dual } \mathcal{Z}) = 0$. Indeed, for an improper $\boldsymbol{x}$ and $\tilde{x} \in$ pro $\boldsymbol{x}$, then $(\boldsymbol{x} - \tilde{x})$ is in dual $\mathcal{Z}$. The outer-approximation $[\boldsymbol{\Delta}_i]$ of the Jacobian is a proper interval, thus in $\mathcal{P}$, $-\mathcal{P}$ or $\mathcal{Z}$, and we can deduce from the multiplication rules that the inner-approximation is non empty only when $[\boldsymbol{\Delta}_i]$ does not contain 0.

EXAMPLE 1. *Let f be defined by $f(x) = x^2 - x$, for which we want to compute an inner-approximation of the range over $\boldsymbol{x} = [2, 3]$. Due to the two occurrences of x, $f(\text{dual}\,\boldsymbol{x})$, computed with Kaucher arithmetic, is not $(f, \boldsymbol{x})$-interpretable. The interval $\tilde{f}(\boldsymbol{x}) = f(2.5) + \boldsymbol{f}'([2, 3])(\boldsymbol{x} - 2.5) = 3.75 + [3, 5](\boldsymbol{x} - 2.5)$ given by its mean-value extension, computed with Kaucher arithmetic, is $(f, \boldsymbol{x})$-interpretable. For $\boldsymbol{x} = [3, 2]$, using the multiplication rule for $\mathcal{P} \times$ dual $\mathcal{Z}$, we get $\tilde{f}(\boldsymbol{x}) = 3.75 + [3, 5]([3, 2] - 2.5) = 3.75 + [3, 5][0.5, -0.5] = 3.75 + [1.5, -1.5] = [5.25, 2.25]$, that can be interpreted as: $\forall z \in [2.25, 5.25],\ \exists x \in [2, 3],\ z = f(x)$. Thus, $[2.25, 5.25]$ is an inner-approximation of* range$(f, [2, 3])$.

In Section 3, we will be using Theorem 1 with f being the solution of the uncertain dynamical system (2): for this, we need to be able to outer-approximate, at any time t, $f(\tilde{x}), \tilde{x} \in$ pro $\boldsymbol{x}$, and its Jacobian with respect to the (uncertain) initial value of the system, $\left\{ \frac{\partial f}{\partial x_i}(x),\ x \in \text{pro } \boldsymbol{x} \right\}$. Computing an enclosure of the solution of an initial value problem is the objective of Section 2.4.

2.4 Enclosing the flow of an uncertain ODE with interval Taylor methods

Consider the uncertain dynamical system (2), where $z = (x, p, t)$ and with initial condition $z(t_0) \in \mathcal{Z}\prime$ at time $t_0 \geq 0$. Let us denote $\mathcal{Z}(t; t_0, \mathcal{Z}\prime)$ the set of solutions of (2) at time t for initial conditions in $\mathcal{Z}\prime$ at t_0. We define a time grid $t_0 < t_1 < \ldots < t_N$, and assume $\mathcal{Z}\prime = \boldsymbol{z}_0 = [\underline{z}_0, \overline{z}_0]$ at time $t_0 \geq 0$.

Interval Taylor methods for guaranteed set integration, see [25] for a review, compute flowpipes that are guaranteed to contain the reachable set of solutions $\mathcal{Z}(t; t_0, \mathcal{Z}\prime)$ of (2) for all time t in $[t_j, t_{j+1}]$. They first verify the existence and uniqueness of the solution using the Banach fixed point theorem and the Picard-Lindelöf operator, and compute an

a priori rough enclosure $[\boldsymbol{r}_{j+1}]$ of $\mathcal{Z}(t)$ for all t in $[t_j, t_{j+1}]$. A tighter enclosure for the set of reachable values for t in $[t_j, t_{j+1}]$ is then computed using a Taylor series expansion of order k of the solution at t_j, where $[\boldsymbol{r}_{j+1}]$ is used to enclose the remaining term :

$$[\boldsymbol{z}](t, t_j, [\boldsymbol{z}_j]) = [\boldsymbol{z}_j] + \sum_{i=1}^{k-1} \frac{(t-t_j)^i}{i!} f^{[i]}([\boldsymbol{z}_j]) + \frac{(t-t_j)^k}{k!} f^{[k]}([\boldsymbol{r}_{j+1}]), \quad (6)$$

where the Taylor coefficients $f^{[i]}$, which are the $i-1$th Lie derivative of f along vector field f, are defined inductively, and can be computed by automatic differentiation as follows (for all $k = 1, \ldots, n$) :

$$f_k^{[1]} = f_k \quad (7)$$

$$f_k^{[i+1]} = \sum_{j=1}^{n} \frac{\partial f_k^{[i]}}{\partial z_j} f_j \quad (8)$$

Let us quickly recall how Equation (6) is obtained. Let $z(t)$ be a solution to Equation (2) starting at time 0 at point z_0. By definition :

$$\frac{dz}{dt}(t) = f(z(t)) = f^{[1]}(z(t))$$

and more generally, we can prove by induction on l that $\frac{d^{(l+1)}z}{dt^{(l+1)}}(t) = f^{[l+1]}(z(t))$, since by induction hypothesis :

$$\begin{aligned} \frac{d^{(l+1)}z}{dt^{(l+1)}}(t) &= \frac{d}{dt}\left(t \mapsto f^{[l]}(z(t))\right) \\ &= \sum_{j=1}^{n} \dot{z}_j(t) \frac{\partial f^{[l]}}{\partial z_j}(z(t)) \\ &= \sum_{j=1}^{n} f_j(z(t)) \frac{\partial f^{[l]}}{\partial z_j}(z(t)) = f^{[l+1]}(z(t)) \end{aligned}$$

Equation (6) is then a direct consequence from Taylor-Lagrange expansion, for sufficiently smooth functions f.

Finally, we use enclosure $[\boldsymbol{z}_{j+1}] = [\boldsymbol{z}](t_{j+1}, t_j, [\boldsymbol{z}_j])$ as initial solution set at time t_{j+1} to derive the interval Taylor model on the next time step.

If evaluated plainly in interval arithmetic, scheme (6) yields enclosures of increasing width. A classical way to control the loss of accuracy due to the loss of correlation in interval arithmetic, called wrapping effect, is a method introduced by Lohner, that uses QR-factorization [25]. Alternatively, we choose here to control wrapping using affine arithmetic [5] instead of interval arithmetic in this evaluation.

3. FORWARD INNER REACHABILITY

As in Section 2.4, we consider the uncertain dynamical system (2), where $z = (x, p, t)$ and with initial condition $z(t_0) \in \mathcal{Z}_\prime = \boldsymbol{z}_0 = [\underline{z}_0, \overline{z}_0]$ at time $t_0 \geq 0$, and we denote $\mathcal{Z}(t; t_0, \boldsymbol{z}_0)$ the set of solutions $\{z(t, z_0),\ z_0(t_0) \in \boldsymbol{z}_0\}$ of (2) at time t. We have seen in Section 2.4, that for a time grid $t_0 < t_1 < \ldots < t_N$, we can compute on each time interval $[t_j, t_{j+1}]$, a flowpipe (6) that is guaranteed to contain the reachable set of solutions of (2) for all time t in $[t_j, t_{j+1}]$. We now want to compute also an inner-approximating flowpipe of this reachable set, that is for all t in $[t_j, t_{j+1}]$, a range $]\boldsymbol{z}[(t, t_j, [\boldsymbol{z}_j])$ such that all values inside that range are sure to be reached at time t by an execution of system (2). For that, we will apply Theorem 1, at all time t, to the function $z_0 \mapsto z(t, z_0)$ from $\mathbb{R}^n$ to $\mathbb{R}^n$ solution of the IVP (2).

In Section 3.1, we give the main lines of the computation of inner-approximated flowpipes, and state the algorithm. We then detail and comment each of its steps. In Section 3.2, we show how we use the classical interval Picard-Lindelöf iteration method to get rough enclosures of the solution and its Jacobian on each time step, that we use for computing the remainders of the Taylor models. In Section 3.3, we build the Taylor models, and show that we can compute the Taylor model of the Jacobian as if we were simply deriving the Taylor model for the solution of the initial ODE, which makes its construction very simple and efficient. Finally, in Section 3.4, we comment the actual computation of the inner-approximating flow-pipe, and show how a loss of accuracy in the outer-approximation results in a loss of accuracy in the inner-approximation, and even possibly to an empty inner-approximation.

3.1 Principle of the algorithm

On each time interval, in order to compute an inner range of the solution of the uncertain system, we need an outer-enclosure of the solution starting from a point in the initial set, $z(t, \tilde{z_0})$ for some $\tilde{z_0} \in \boldsymbol{z}_0$ (Equation (15) in Algorithm 1), an enclosure of the solution by the system $z(t, z_0)$ over range $z_0 in \boldsymbol{z}_0$ (Equation (16)), and an enclosure of its Jacobian with respect to z_0, evaluated over range $\boldsymbol{z}_0$ (Equation (17)). The Jacobian is defined $J_{ij}(t, \boldsymbol{z}_0) = \frac{\partial z_i}{\partial z_{0,j}}(t, \boldsymbol{z}_0)$, for i and j between 1 and n, and where z_i is the i-th component of the vector flow function z, and $z_{0,j}$ the j-th component of the vector of initial conditions z_0.

We compute these outer-approximations by applying the Taylor method of Section 2.4 to $z(t, \tilde{z_0})$ and $J(t, z_0)$ where $z_0 \in \boldsymbol{z}_0$ and with initial condition $J(t_0) = Id$ the identity matrix : $z(t, \tilde{z_0})$ satisfies system (2) with $z(t_0) = \tilde{z_0} \in \boldsymbol{z}_0$, so that we can directly use the Taylor expansion (6) on each time interval $[t_j, t_{j+1}]$ to compute $\mathcal{Z}(t; t_0, \tilde{z_0})$. The coefficients of the Jacobian matrix of the flow satisfy :

$$\dot{J}_{ij}(t, z_0) = \sum_{k=1}^{n} \frac{\partial f_i}{\partial z_k}(z).J_{kj}(t, z_0) \quad (9)$$

that can be rewritten

$$\dot{J}(t, z_0) = \mathrm{Jac}_z f(z(t, z_0)).J(t, z_0). \quad (10)$$

with $J(t_0) = Id$ (these are the "variational equations" used in particular in [29] for improving outer-approximations of continuous dynamical systems). We will denote by $\nabla_z f$ the function of variables z_i and J_{ij} (linear in J_{ij}) which is the right hand side of Equation 10. Hence, its pq entry is :

$$(\nabla_z f)_{pq}(z, J) = \sum_{k=1}^{n} \frac{\partial f_p}{\partial z_k}(z) J_{kq} \quad (11)$$

A Taylor expansion can thus be used to outer-approximate the solution of (10) noted $\mathcal{J}(t; t_0, \boldsymbol{z}_0)$ on each time interval $[t_j, t_{j+1}]$, using the outer-approximation for $z(t, z_0)$ given by Taylor expansion (6). Equations (9) together with Equations (1) define a system of $n(n+1)$ ordinary differential equations in $n(n+1)$ variables (z and J). We call the corresponding vector field F and write similarly, by an abuse of notation, for H a function in variables z and J, $H^{[i]}$ the $i-1$th Lie derivative of H along the (augmented) vector

field H. More explicitly, it is defined inductively as follows :

$$H^{[1]} = H \quad (12)$$

$$H^{[i+1]} = (H^{[i]})^{[2]} \quad (13)$$

where the first Lie derivative is :

$$H^{[2]} = \sum_{i=1}^{n} \frac{\partial H}{\partial z_i} f_i + \sum_{k,l=1}^{n} \frac{\partial H}{\partial J_{kl}} \left(\sum_{s=1}^{n} \frac{\partial f_k}{\partial z_s} J_{sl} \right) \quad (14)$$

Let us briefly detail how we obtained Equation (14). Consider a solution $t \mapsto z_i(t)$ and $t \mapsto J_{ij}(t)$ of Equations (9) and (1). The Lie derivative of H is the time derivative of $\tilde{H}(t) = H(z_1(t), \ldots, z_n(t), J_{11}(t), \ldots, J_{nn}(t))$:

$$\dot{\tilde{H}}(t) = \sum_{i=1}^{n} \tfrac{\partial H}{\partial z_i}(t)\dot{z}_i(t) + \sum_{k,l=1}^{n} \tfrac{\partial H}{\partial J_{kl}}(t)\dot{J}_{kl}(t)$$

We get the formula (14) by replacing $\dot{z}_i$ by its expression in Equation (1) and $\dot{J}_{kl}$ by its expression in Equation (9).

We summarize the procedure for computing the inner and outer reachable sets in pseudo-Algorithm 1.

In the next sections, we detail the different steps of Algorithm 1 and illustrate them on a running example:

EXAMPLE 2. *We consider the Brusselator equation:*

$$f(x) = \begin{pmatrix} 1 - 2x_1 + \frac{3}{2}x_1^2 x_2 \\ x_1 - \frac{3}{2}x_1^2 x_2 \end{pmatrix}$$

with $x = (x_1, x_2)$, over the time interval $[0, h]$ $\left(h = \frac{1}{20}\right)$, and with initial conditions $[x_0] = ([2, 2.15], [0.1, 0.15])$.

The Jacobian that appears in Equation (10) is :

$$Jac_z f(z(t, z_0)) = \begin{pmatrix} -2 + 3x_1x_2 & \frac{3}{2}x_1^2 \\ 1 - 3x_1x_2 & -\frac{3}{2}x_1^2 \end{pmatrix}$$

3.2 Step 1: computing the rough enclosures

In order to compute the kth term in Equations (15) and (16) we need to compute $[\boldsymbol{r}_j]$ (respectively $[\boldsymbol{R}_j]$), i.e. a priori enclosures of the components of the solutions z and J over the time interval $[t_j, t_{j+1}]$. This is done following the classical approach [25] relying on the interval Picard-Lindelöf method. This goes as follows. First note that Equation (1) can be rewritten as the integral equation

$$z(t) = z_0 + \int_{t_j}^{t_{j+1}} f(z(s))\mathrm{d}s \quad (19)$$

and define F the functional which to function z associates the right-hand side of Equation (19). Under the condition that f is Lipschitz, F admits a unique fixpoint, solution to Equations (1) and (19). The interval version $F^\sharp$ of the Picard-Lindelöf operator F enjoys the same property and is derived using the obvious rough interval approximation of the integral : $F^\sharp([z]) = z_0 + [t_j, t_{j+1}][\boldsymbol{f}]([z])$ (where $[z]$ will denote ultimately the "rough" enclosure of the solutions to Equation (1) and $[\boldsymbol{f}]$ denotes the interval extension of function f). Simple Jacobi like iteration suffices to reach the fixpoint of $F^\sharp$: $[z]_0 = z_0$, $[z]_{i+1} = F^\sharp([z]_i)$ for all $i \in \boldsymbol{N}$. Convergence can be ensured using outwards rounding in finite precision, numerical acceleration techniques etc.

EXAMPLE 3. *We carry on with the computation of outer-approximations for solutions and Jacobians for the Brusselator on the first time step. We will write $[x_i](t)$ instead of $[x_i](t, 0, [x_0])$ as we are only considering the first time step. We first need to determine the rough enclosures $[\boldsymbol{r}_1]_i$ and $[\boldsymbol{R}_1]_{i,j}$ of the $x_i(t)$ and $J_{ij}(t)$ over $t \in [0, h]$, $x \in [x_0]$ using the interval Picard-Lindelöf method of Section 3.2 :* $[\boldsymbol{r}_1] = \begin{pmatrix} [1.86, 2.15] \\ [0.10, 0.23] \end{pmatrix}$, $[\boldsymbol{R}_1] = \begin{pmatrix} [0.92, 1.00] & [0.00, 0.35] \\ [-0.025, 0.022] & [0.65, 1.00] \end{pmatrix}$.

The remainders for $k = 2$ (first order Taylor model for Equations (15) and (17)) will be determined in Example 4.

Data: a time grid $t_0 < t_1 < \ldots < t_N$, an initial range $\boldsymbol{z}_0$, and some $\tilde{z}_0 \in \boldsymbol{z}_0$
Result: $[\boldsymbol{z}](t, \boldsymbol{z}_0)$ and $]\boldsymbol{z}[(t, \boldsymbol{z}_0)$ over $t = [t_0, t_N]$
Init: $j = 0$, $t_j = t_0$, $[\boldsymbol{z}_j] = \boldsymbol{z}_0$, $[\tilde{z}_j] = \tilde{z}_0$, $[\boldsymbol{J}_j] = Id$
while $j < N - 1$ **do**

Step 1. compute a priori enclosures $[\boldsymbol{r}_{j+1}]$ of $\overline{\mathcal{Z}(t; t_j, \boldsymbol{z}_j)}$ for all t in $[t_j, t_{j+1}]$, $[\tilde{\boldsymbol{r}}_{j+1}]$ of $\mathcal{Z}(t; t_j, \tilde{z}_j)$ for all t in $[t_j, t_{j+1}]$, and $[\boldsymbol{R}_{j+1}]$ of $\mathcal{J}(t; t_j, \boldsymbol{z}_j)$

Step 2. build the Taylor Models valid on $[t_j, t_{j+1}]$:

$$[\boldsymbol{z}](t, t_j, [\boldsymbol{z}_j]) = [\boldsymbol{z}_j] + \sum_{i=1}^{k-1} \frac{(t - t_j)^i}{i!} f^{[i]}([\boldsymbol{z}_j]) + \frac{(t - t_j)^k}{k!} f^{[k]}([\boldsymbol{r}_{j+1}]). \quad (15)$$

$$[\tilde{\boldsymbol{z}}](t, t_j, [\tilde{\boldsymbol{z}}_j]) = [\tilde{\boldsymbol{z}}_j] + \sum_{i=1}^{k-1} \frac{(t - t_j)^i}{i!} f^{[i]}([\tilde{\boldsymbol{z}}_j]) + \frac{(t - t_j)^k}{k!} f^{[k]}([\tilde{\boldsymbol{r}}_{j+1}]). \quad (16)$$

$$[\boldsymbol{J}](t, t_j, [\boldsymbol{z}_j]) = [\boldsymbol{J}_j] + \sum_{i=1}^{k-1} \frac{(t - t_j)^i}{i!} \mathrm{Jac}_z f^{[i]}([\boldsymbol{z}_j]).[\boldsymbol{J}_j] + \frac{(t - t_j)^k}{k!} \mathrm{Jac}_z f^{[k]}([\boldsymbol{r}_{j+1}]).[\boldsymbol{R}_{j+1}] \quad (17)$$

Step 3. deduce an inner-approximation valid for t in $\overline{[t_j, t_{j+1}]}$: if $]\boldsymbol{z}[(t, t_j)$ defined by Equation (18) is an improper interval

$$]\boldsymbol{z}[(t, t_j) = [\tilde{\boldsymbol{z}}](t, t_j, [\tilde{\boldsymbol{z}}_j]) + [\boldsymbol{J}](t, t_j, [\boldsymbol{z}_j]) * ([\overline{z_0}, \underline{z_0}] - \tilde{z_0}) \quad (18)$$

then interval pro $]\boldsymbol{z}[(t, t_j)$ is an inner-approximation of the set of solutions $\{z(t, z_0),\ z_0(t_0) \in \boldsymbol{z}_0\}$ at time t, otherwise the inner-approximation is empty.

Step 4. $[\boldsymbol{z}_{j+1}] = [\boldsymbol{z}](t_{j+1}, t_j, [\boldsymbol{z}_j])$, $\overline{[\tilde{\boldsymbol{z}}_{j+1}]} = [\tilde{\boldsymbol{z}}](t_{j+1}, t_j, [\tilde{\boldsymbol{z}}_j])$, $[\boldsymbol{J}_{j+1}] = [\boldsymbol{J}](t, t_j, [\boldsymbol{z}_j])$

end

Algorithm 1: Computing inner and outer reachable sets

3.3 Step 2: building the Taylor models

Building the Lie derivatives of the Jacobian.

The formulation of Equation (17) relies on the possibility to commute the ith Lie derivative with the calculation of

the Jacobian. Without this, we would have written:

$$[\boldsymbol{J}](t,t_j,[\boldsymbol{z}_j]) = [\boldsymbol{J}_j] + \sum_{i=1}^{k-1} \frac{(t-t_j)^i}{i!}(\mathrm{Jac}_z(f))^{[i]}([\boldsymbol{z}_j]).[\boldsymbol{J}_j] + \frac{(t-t_j)^k}{k!}(\mathrm{Jac}_z(f))^{[k]}([\boldsymbol{r}_{j+1}]).[\boldsymbol{R}_{j+1}] \quad (20)$$

$\mathrm{Jac}_z(f)$ being seen as a function of variables z_i and J_{ij}, which is linear in the J_{ij} as in Equation (11). Equations (17) and (20) are equivalent since the two derivatives (the Jacobian calculation and the Lie derivative) commute.

We prove this equivalence by induction on the number of Lie derivations. For $i = 1$, we have, by definition $\nabla_z(f^{[1]}) = \nabla_z(f) = (\nabla_z(f))^{[1]}$. Suppose now we have, as an induction step $\nabla_z(f^{[i]}) = (\nabla_z(f))^{[i]}$. We now write :

$$\nabla_z(f^{[i+1]})_{pq} = \sum_{k=1}^{n} \frac{\partial f_p^{[i+1]}}{\partial z_k} J_{kq} = \sum_{k=1}^{n} \frac{\partial}{\partial z_k}\left(\sum_{l=1}^{n} \frac{\partial f_p^{[i]}}{\partial z_l} f_l\right) J_{kq}$$

hence,

$$\nabla_z(f^{[i+1]})_{pq} = \sum_{k,l=1}^{n} \left(\frac{\partial^2 f_p^{[i]}}{\partial z_k \partial z_l} f_l + \frac{\partial f_p^{[i]}}{\partial z_l}\frac{\partial f_l}{\partial z_k}\right) J_{kq} \quad (21)$$

On the other hand we have :

$$(\nabla_z(f)_{pq})^{[i+1]} \quad = \quad (\nabla_z(f^{[i]})_{pq})^{[2]}$$

by Definition (14) and by the induction step. Using now Equation (14), and Equation (11) recalled below:

$$(\nabla_z(f^{[i]}))_{pq} = \sum_{k=1}^{n} \frac{\partial f_p^{[i]}}{\partial z_k} J_{kq}$$

we have :

$$(\nabla_z(f)_{pq})^{[i+1]} = \sum_{k,l=1}^{n} \frac{\partial^2 f_p^{[i]}}{\partial z_k \partial z_l} f_l J_{kq} + \sum_{r=1}^{n} \frac{\partial f_p^{[i]}}{\partial z_r}\left(\sum_{t=1}^{n} \frac{\partial f_r}{\partial z_t} J_{tq}\right) \quad (22)$$

which is thus seen to be equal to $\nabla_z(f^{[i+1]})_{pq}$ by Equation (21).

Equation (17) is simpler to compute since we already computed the Lie derivative of f, in Equation (16), the Jacobian calculation being by itself rather inexpensive.

Computing the coefficients of the Taylor models.

We need to outer-approximate the values of some functions, and in particular, all Lie derivatives, which are coefficients in the Taylor models, in Equations (15-17). We have a wide choice from the existing set-based methods, is polynomial. We will use in our running example affine arithmetic [5], as in our prototype. Affine arithmetic was also used in [13] for inner-approximations of discrete dynamical systems. One interest is that we can use the results from [12] to also get good estimates of the joint inner range of the state variables z_j, altogether, when needed.

EXAMPLE 4. *We compute a first-order Taylor model for the Brusselator, using Equation (6) with $k = 2$, and using affine arithmetic to compute $Jac_z(f^{[i]})([\boldsymbol{r}_{j+1}])[\boldsymbol{J}_j]$ and $f^{[i]}([\boldsymbol{z}_j])$. We start with $[x_0] = ([2, 2.5], [0.1, 0.15])$, hence, in affine arithmetic, $[x_0] = \left(\frac{83}{40} + \frac{3}{40}\epsilon_1, \frac{1}{8} + \frac{1}{40}\epsilon_2\right)$. We evaluate $f^{[1]} = f$ using simple rules from affine arithmetic, e.g. :*

$$f^{[1]}([x_0]) = -\frac{119919}{51200} - \frac{1173}{12800}\epsilon_1 + \frac{41361}{256000}\epsilon_2 + \frac{27}{51200}\eta_1 + \frac{3015}{256000}\eta_2 \quad (23)$$

and, e.g. $Jac_z(f^{[1]})([x_0])_{11} = -\frac{391}{320} + \frac{9}{320}\epsilon_1 + \frac{249}{1600}\epsilon_2 + \frac{9}{1600}\eta_3$. The non-linearity of f and its Jacobian produces new symbols in the evaluation with affine arithmetic than ϵ_1, ϵ_2 : we note them using the η letter, instead of ϵ, to make apparent the uncertainty produced by the interpretation in affine arithmetic. Equation (23) evaluates in interval $[-2.6077, -2.0766]$.

To obtain the remainders, we compute $f_i^{[2]}$ and $Jac_z(f_i^{[2]})$:

$$f_1^{[2]} = -2 + 4x_1 + 3x_1x_2 + \frac{3}{2}x_1^3 - 9x_1^2x_2 + \frac{9}{2}x_1^3x_2^2 - \frac{9}{4}x_1^4x_2 \quad (24)$$

$$Jac_z(f^{[2]})_{11}(x_1,x_2)(J_{11},J_{21}) = (3x_2f_1 + 3x_1f_2)J_{11} + (-2 + 3x_1x_2)Jac_z(f^{[1]})_{11}(x_1,x_2)(J_{11},J_{21}) + 3x_1f_1J_{21} + \frac{3}{2}x_1^2 Jac_z(f^{[1]})_{21}(x_1,x_2)(J_{11},J_{21}) \quad (25)$$

where $Jac_z(f^{[1]})_{11}$ and $Jac_z(f^{[1]})_{21}$ are just $Jac_z f(z(t,z_0))_{11}$ and $Jac_z f(z(t,z_0))_{21}$ given in Example 2.

Now again, we are applying affine arithmetic to compute $f^{[2]}([\boldsymbol{r}_1])$ and $Jac_z(f^{[2]})([\boldsymbol{r}_1])[\boldsymbol{R}_1]$ given the rough enclosures $\boldsymbol{r}_1$ and $\boldsymbol{R}_1$ computed in Example 3 and we find :

$$f^{[2]}([\boldsymbol{r}_1]) \quad = \quad \begin{pmatrix} [3.5371, 13.7617] \\ [-11.1499, -1.5368] \end{pmatrix} \quad (26)$$

and $Jac_z(f^{[2]})([\boldsymbol{r}_1])[\boldsymbol{R}_1]$ is the matrix :

$$\begin{pmatrix} [-3.1897, 15.7653] & [-74.8884, -19.3243] \\ [-14.0978, 3.5433] & [16.4500, 68.0696] \end{pmatrix}$$

As a direct consequence, we can evaluate Equations (15) and (17) to get the outer-approximation of z and J at time h :

$$[\boldsymbol{z}](h,t_0,[\boldsymbol{z}_0]) = ([1.88320, 2.05421], [0.15728, 0.20358]) \quad (27)$$

$$[\boldsymbol{J}](h,t_0,[\boldsymbol{z}_0]) = \begin{pmatrix} [0.92545, 0.96808] & [0.20597, 0.32253] \\ [-0.016, 0.02499] & [0.67388, 0.78551] \end{pmatrix} \quad (28)$$

For instance, the first component in Equation (27) is found by using Equation (15) with $k = 2$ and by instantiating the constant coefficient with $[x_0] = ([2, 2.5], [0.1, 0.15])$, the Taylor coefficient in degree one in t with the result of Equation (23), and in degree two with the result of Equation (26). Using a coarser interval abstraction of coefficients in degree zero and one, we indeed find an outer-approximation of the flowpipe until time h : $[\boldsymbol{z}_1](h,t_0,[\boldsymbol{z}_0]) = [2, 2.5] + [-2.6077, -2.0766]\,t + [3.5371, 13.7617]\,\frac{t^2}{2}$ equal at time h to $[1.8740, 2.0634]$, which slightly over-approximates the result of Equation (27) with affine arithmetic.

Computing the center of the inner-approximation.

Equation (16) is required to evaluate Equation (18) and get inner-approximations: we need to propagate a (center) point in the set of initial values through the flowpipe of solutions of ODE (1), at each time step t_j in our time grid.

This center solution is certainly not derivable from the outer-approximation of the flowpipe, e.g. as its midpoint: in order to soundly use the mean-value theorem, this solution must outer-approximate the image by the flow of the initial point.

We use the same Taylor expansion, but with different initial conditions, to compute in (15) an outer-approximation of the solution of system (2) with $z(t_0) = \tilde{z_0}$, used as the center in inner-approximation (18), and in (16) an outer-approximation of the solution of the same system but with uncertain $z(t_0) \in \boldsymbol{z}_0$, used to compute the Taylor coefficients in Equation (17).

EXAMPLE 5. *Starting with the center* $\tilde{x}_0 = (2.075, 0.125)$ *of the initial condition* $[x_0] = ([2, 2.15], [0.1, 0.15])$*, and applying the interval Picard-Lindelöf method of Section 3.2, we find* $x = ([1.9655, 1.9718], [0.1774, 0.1831])$ *at time* h*.*

3.4 Step 3: computing the inner-approximation

The algorithm described in Section 3.1 fully relies on outer-approximations at each step, to deduce an inner-approximation at Step 3. This means that we can soundly compute and implement most of our approach using interval-based methods with outward rounding as classically: outward rounding should be used for the outer approximations of flows and Jacobians (the larger these, the tighter the inner-approx), but the computation by Kaucher aithmetic of improper intervals should be done with inward rounding.

Also, the wider the outer-approximation in Taylor models (15-17), the tighter thus the less accurate the inner-approximation (18): it can even lead to an empty inner-approximation if the result of Equation (18) in Kaucher arithmetic is not an improper interval.

The phenomenon we mentioned above can occur in two ways. First, $[\overline{z_0}, \underline{z_0}] - \tilde{z_0}$ is an improper interval that belongs to dual $\mathcal{Z}$ as defined in Section 2.2. The outer-approximation of the Jacobian matrix, $[\boldsymbol{J}](t, t_j, [\boldsymbol{z}_j])$ is a proper interval. The Kaucher multiplication, as mentioned in Section 2.3, will yield a non-zero improper interval only if $[\boldsymbol{J}](t, t_j, [\boldsymbol{z}_j])$ does not contain 0. And, in this case, the result of this multiplication will depend on the lower bound of the absolute value of the Jacobian (while the same mean-value theorem used for outer-approximation would imply a multiplication of proper intervals that would depend on the upper bound of the absolute value of the Jacobian). The larger this lower bound, the wider the inner-approximation.

Suppose that the Kaucher multiplication yields an improper interval. It is added to proper outer-approximation $[\tilde{\boldsymbol{z}}](t, t_j, [\tilde{\boldsymbol{z}}_j])$ of the solution at time t of the system starting from point $\tilde{\boldsymbol{z}}_0$. Ideally, this should be tight, but if this interval is wider than the improper interval resulting from the Kaucher multiplication, then the sum of the two intervals - computed using the extension of interval addition - will be proper, and the inner-approximation empty.

The quality of the inner-approximation is strongly linked to the quality of the outer-approximation. We can if necessary locally improve the quality by using higher-order Taylor models. Indeed, as we know that the exact reachable set of the uncertain system lies between the inner and outer-approximated flows, we can bound the approximation error at each instant, and use this information to dynamically refine the approximation.

EXAMPLE 6. *Now we can instantiate Equation (18) as follows, for e.g. the first component of* x *and time* $t = h$*, using the result of Example 5 for the outer-approximation of the center at time* h *and Equation (28) for the outer-approximation of the Jacobian at time* h *:*

$$]\boldsymbol{z}[(h,0) = [1.9655, 1.9718] + [0.9254, 0.9680]\,[0.075, -0.075] + [0.2059, 0.3225]\,[0.025, -0.025] \quad (29)$$

Finally, using Kaucher arithmetic (see Section 2.2), we find

$$]\boldsymbol{z}[(h,0) = [1.9655 + 0.92545 \times 0.075 + 0.20597 \times 0.025,\ 1.9718 - 0.9254 \times 0.075 - 0.20597 \times 0.025]$$

whose proper counterpart is $[1.8973, 2.0400]$*. We thus efficiently find a quite tight characterization of the reachable set with a very low order scheme, for the Brusselator at time* h *:*

$$[1.8973, 2.0400] \subseteq z(h, 0, [z_0]) \subseteq [1.88320, 2.05421]$$

Of course, similarly to Example 4 for the outer-approximation of z*, Equation (18) is valid for all times* t *in* $[0, h]$*, hence gives an inner-approximation of the flowpipe for the Brusselator. This is what we will be doing in Section 4.1.*

4. EXPERIMENTS AND BENCHMARKS

We implemented[1] our method relying on the `FILIB++` C++ library [22] for interval computations, the `FADBAD++` package (http://www.fadbad.com) for automatic differentiation, and (a slightly modified version of) the `aaflib` library (http://aaflib.sourceforge.net) for affine arithmetic. Affine arithmetic is used for the coefficients of the Taylor models in order to limit the wrapping effect. Matrix preconditioning is a more classical alternative, but affine arithmetic proved to be both efficient, if we limit the number of noise symbols used, and accurate. It is also very convenient for prototyping, as we rely on the `aaflib` library.

We first demonstrate in Section 4.1 the good behavior of our inner-approximated flowpipes on the quite difficult Brusselator model. Then, in Section 4.2, we provide some comparison to the experimental results of the related work.

4.1 Brusselator

We consider in this section another instance of the Brusselator system, slightly different from the version of Example 2, which has been used in e.g. [4, 28] :

$$\begin{cases} \dot{x}_1 &= 1 + x_1^2 x_2 - 2.5 x_1 \\ \dot{x}_2 &= 1.5 x_1 - x_1^2 x_2 \end{cases}$$

with $x_1(0) \in [0.9, 1]$ and $x_2(0) \in [0, 0.1]$.

We use Taylor models of order 4 in time, and represent in Figure 1, the inner and outer approximated flowpipes for variables x_1 and x_2, up to a maximum time $t = 10$. The inner-approximations are represented in dashed lines, and the outer-approximations in plain lines.

We can note that the width of the inner-approximation (internal dashed lines) decreases at times, and the inner-approximation even becomes empty (for example for variable x_2 around $t = 4$), but the width can still later be non-zero again. This is not a bug: this phenomenon is an illustration of the fact detailed in Section 3.4, that when adding an improper with a proper interval to get the inner range of a variable, we can get a proper interval, which results in an empty inner-approximation (of the variable of interest

[1]available from http://www.lix.polytechnique.fr/Labo/Sylvie.Putot/software.html

- actually, you can note that on this example, the inner-approximations of the two variables do not become empty at the same time). This does not prevent us from carrying on with the computation of the Taylor models : non-empty inner-approximations will be obtained at later times, depending on the behavior of the Jacobian of the flow.

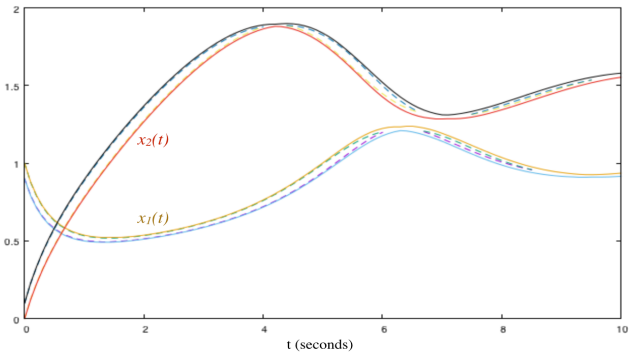

Figure 1: Brusselator (x_1 and x_2), Taylor models order 4

4.2 Comparisons to the related work

We provide in this section some elements of comparisons to the experimental results given in [4, 28]. Let us highlight that it is difficult to compare these methods in a fair manner, as evaluating the compared accuracy of these methods is difficult. Also, our implementation is preliminary (using fixed step of integration for instance), while some of the related work relies on the highly optimized interval solver for initial value problem `VNODE-LP` [24].

Among the examples studied in both [4] and [28], we first selected the version of the Brusselator introduced in Section 4.1 as a representative of the systems of low degree. We chose as second example a biological system of higher degree (7), so as to demonstrate the way our approach scales.

$$\begin{pmatrix} \dot{x_1} \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} -0.4x_1 + ax_3x_4 \\ 0.4x_1 - x_2 \\ x_2 - ax_3x_4 \\ ax_5x_6 - ax_3x_4 \\ -ax_5x_6 + ax_3x_4 \\ 0.5x_7 - ax_5x_6 \\ -0.5x_7 + ax_5x_6 \end{pmatrix} \quad (30)$$

In this system, a is a parameter which is taken equal to 50 in [4] and to 5 in [28]. We will use the corresponding value of the parameter when comparing to the related work.

4.2.1 Comparison to [4]

In [4], the accuracy of computations is measured by the minimum width ratio

$$\gamma_{\min} = \min \frac{\gamma_u(v)}{\gamma_o(v)}, v \in V$$

where V is a set of vectors, and $\gamma_u(v)$ and $\gamma_o(v)$ measure respectively the width of the inner-approximation and outer-approximation in direction $v \in V$. Intuitively, the larger this ratio, the better the approximation. Our method naturally gives inner ranges for the projection of the flow system on its state variables. We thus measure in our case the minimum over the state variables x_i of our system of this ratio. We believe this corresponds to the measure that was used for experiments in [4], as they mention that the vectors are selected along the dimensions (axis-aligned).

Comparison on the Brusselator.

The initial set taken in [4], defined by $x_1 \geq 0.9$, $x_2 \geq 0$, $x_1 + x_2 - 1 \leq 0$, can be projected on $1 \geq x_1 \geq 0.9$, $0.1 \geq x_2 \geq 0$. This outer-approximation of the initial set is quite inaccurate, which results in a lower quality of the inner-approximation that must be taken into account in the comparison to the results of [4]. We could actually also consider initial sets that are not given as boxes but for instance as zonotopes, but we did not investigate this here.

In [4], the authors study the result for $t = 3$ and $t = 4$, with 4th order Taylor models and integration time step $h = 0.02$. We choose, as they do, 4th order Taylor models, and time step $h = 0.02$. Our implementation until $t = 4$ takes a total of 3.2 seconds (to compute both outer and inner approximations), where [4] takes 89 seconds. Our implementation is thus more than an order of magnitude quicker. Note also that with our approach, taking order 3 Taylor models and a larger time step of 0.1, we still obtain results of very similar quality on $\gamma_{\min}$, in 0.25 seconds. Further decreasing the precision starts degrading the quality of results.

In Figure 2, we represent $\gamma_{\min}$ as a function of time, for a time range extended to a maximum time of 10. We observe that at $t = 3$, the relative width of the inner-approximation over the outer-approximation is of order 0.7, which is equal to the value given in Table 1 of [4]. However, this ratio decreases quickly, mostly due to the x_2 component, and at $t = 4$, we get a ratio very close to 0.1, instead of 0.55 as in [4]. Indeed, $t = 4$ is a time at which, as already noted in Section 4.1, our inner-approximation of variable x_2 is temporarily of lower quality, even though the x_1 inner estimate is still of high quality. It is only temporary, as at further times the quality improves, before degrading again.

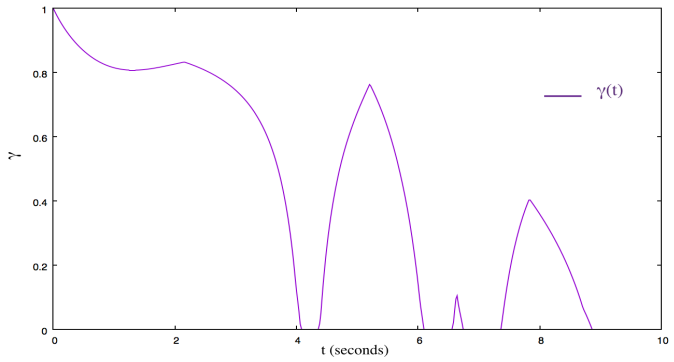

Figure 2: Evolution of $\gamma_{\min}$ with time

In Figure 3, we represent the evolution with time of the widths of the inner- and outer-approximations of x_2, the component that makes $\gamma_{\min}$ decrease drastically around $t = 4$. Whenever the width of the inner-approximation on x_2 decreases to zero, the width of the outer-approximation is also strongly decreasing: the difference between these widths remains almost stable. We also note again that the inner-approximation becoming empty at some point does not impact the behavior of the inner-approximation at further times. Finally, we note that this system looks quite stable, with so-

lution widths that tend to decrease with time. This is a difficult case for the inner-approximation, as its width naturally tends towards 0 due to the problem, as we note on Figure 3. We advocate that inner and outer-approximations should be considered jointly in order to assess the behavior of a system.

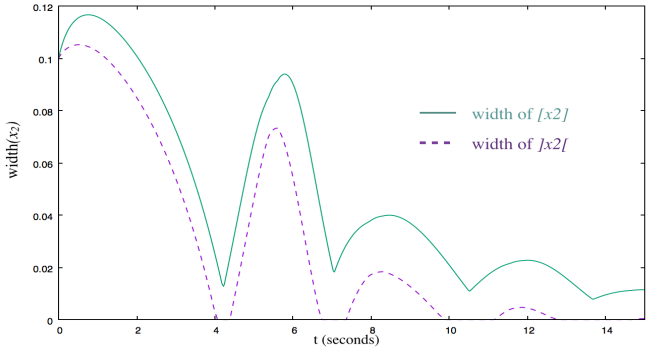

Figure 3: Evolution with time of the width of inner- and outer-approximations on x_2

Comparison on the biological system.

We now consider the biological system, with initial condition $\boldsymbol{x}_0 \in [0.1, 0.1175] \times ... \times [0.1, 0.1175]$, which is an outer-approximation of the simplex taken in [4]. We compute inner and outer approximated flowpipes for time in [0,0.2], with order 5 Taylor Models and a step size of 0.01. The computation completes in 4.7 seconds, and we get as a measure of quality of the approximation $\gamma_{\min}(t = 0.2) \approx 0.65$, which is this time a much better accuracy than the $\gamma_{\min}(t = 0.2) = 0.25$ obtained in 632 seconds in [4]. This seems to confirm that our approaches scales very well to high dimensional systems, with a very good accuracy.

We also measure as an indication of the accuracy the mean value on the components x_i of the distance between the inner and outer approximations x_i^{in} and x_i^{out}, computed as

$$\sum_{i=1}^{n} \frac{\max(\sup(x_i^{out}) - \sup(x_i^{in}), \inf(x_i^{in}) - \inf(x_i^{out}))}{n}.$$

It gives an over-estimation of the error between the inner-approximation and the exact reachable state at time t: this value for $t = 0.2$ is 4.10^{-3}.

4.2.2 Comparison to [28]

Comparison on the Brusselator.

In [28] the authors take $X = [0.3, 0.4] \times [0.5, 0.7]$ for a time frame in [0,1.1], and a time step h=0.05. We compute the inner and outer-approximations, for same time step and with order 3 Taylor Models. In [28], the accuracy of the result is estimated by a parameter ϵ_M that bounds the size of the boxes used to approximated the boundary of the exact reachable set, inside which they will look for an inner-approximation. We believe this parameter comparable in spirit to our measure of the maximum distance between the inner and the outer approximations. We can point 2 differences. First, in our case we compute the distance to the outer-approximation, which will always be greater than the distance to the exact reachable set. Also, our understanding is that there are no guarantees in the method of [28] on the actual distance from the inner-approximation itself to the reachable set, whereas our bounds are guaranteed.

We obtain in 0.8 seconds a distance equal to 0.005 for x_1, and 0.01 for x_2, which is 10 times larger than the estimation 0.001 obtained in 55 seconds in [28].

Comparison on the biological system.

We now consider the biological system, with, as in [28], a box initial condition $\boldsymbol{x}_0 \in [-0.015, 0.001] \times ... \times [-0.015, 0.001]$. In order to compare our results to their backward estimate, we consider the reverse flow, for time in [0,0.2] , and use order 3 Taylor Models and a step size of 0.02. Our computation of the inner and outer approximated flowpipes takes 0.2 seconds. We get inner-approximations that always strictly contain the ones of [28], which takes 0.67 seconds. Our analysis is thus both faster and with better accuracy.

Using as quality measure the measure of [4] that computes the ratio γ of the width of the inner-approximation over the width of the outer-approximation, here componentwise on the variables, and using our outer-approximation in both cases, we obtain for our approximation $(\gamma_1, \ldots, \gamma_n) = (0.970, 0.999, 0.973, 0.938, 0.938, 0.970, 0.971)$, while with the results of [28] we get the lower quality results $(\gamma_1, \ldots, \gamma_n) = (0.85, 0.86, 0.22, 0.84, 0.84, 0.85, 0.85)$.

We also compare our results with the outer-approximation computed by VNODE with a Taylor model of order 5. We represent on Figure 4 the upper bound for each of the 7 variables x_i on the outer-approximation by VNode, our outer-approximation, our inner-approximation, and that of [28], in that order, from left to right. The results confirm that

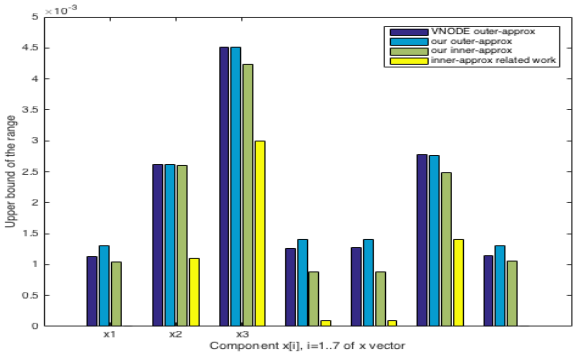

Figure 4: Upper bounds on inner and outer-approximations on the 7 variables of the biological system

our inner-approximation (third bar of each group in the Figure) is generally very close to the outer-approximations (first two bars in each group), while the inner-approximation of the related work (fourth bar) is of lower quality, and on some variables even absent (upper bound equal to zero).

5. CONCLUSION AND FUTURE WORK

We presented an approach to compute inner-approximating flowpipes of uncertain ODEs, that extends in a simple way Taylor-based methods for outer-approximation. The joint computation of the inner and outer-approximating flowpipes gives us a bound on the error to the exact reachable

set at each instant, that we can use if needed to dynamically refine the approximation by using higher order Taylor models or smaller step sizes. Indeed, accurate error estimation, which is usually a difficult task, is a direct outcome of our computation. A natural future extension of the present work, is the inner-approximation of reachable sets in presence of guards and constraints, so as to handle general hybrid systems. Our approach allows us to inner-approximate not only the variables as demonstrated here, but also the projection on whatever function of these variables, not only at specific times like the related work, but as flowpipes, that can be used for detection of intersection. The symbolic information included in our model when we evaluate the Taylor models with affine arithmetic, will allow us to use existing work on the inner-approximation of joint range of functions and constraint solving, as e.g. [15].

Among the interests of inner-approximation is the possibility to falsify properties. Our objective in that respect, is to use the combination of inner and outer approximations to tackle the verification and falsification of temporal properties of uncertain hybrid systems, along the lines of [16,30].

Finally, this work can be applied to other, related problems. First, this can be applied to backward reachability problems, such as the ones treated in e.g. [28], by considering the opposite vector field. Secondly, this can be applied to a particular backward reachability problem : the inner-approximation of region of attractions, for which we should compare our method with existing work [19].

6. ACKNOWLEDGMENTS

The authors were supported by ANR project MALTHY, ANR-13-INSE-0003, DGA project "Complex Robotics Systems Safety" and the academic chair "Complex Systems Engineering" of Ecole polytechnique-ENSTA-Télécom-Thalès-Dassault-DCNS-DGA-FX-FDO-Fondation ParisTech.

7. REFERENCES

[1] M. Althoff. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In *HSCC'13*, pages 173–182. ACM, 2013.

[2] M. Althoff. An introduction to CORA 2015. In *In ARCH 2014 and ARCH 2015*, pages 120–151, 2015.

[3] X. Chen, E. Ábrahám, and S. Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *CAV*, pages 258–263, 2013.

[4] X. Chen, S. Sankaranarayanan, and E. Abraham. Under-approximate flowpipes for non-linear continuous systems. In *FMCAD*, pages 59–66. IEEE/ACM, 2014.

[5] J. Comba and J. Stolfi. Affine arithmetic and its applications to computer graphics. *In SIBGRAPI*, 1993.

[6] T. Dang, O. Maler, and R. Testylier. Accurate hybridization of nonlinear systems. In *HSCC*, 2010.

[7] G. Frehse, C. L. Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. Spaceex: Scalable verification of hybrid systems. In *CAV*, pages 379–395, 2011.

[8] A. Girard. Reachability of uncertain linear systems using zonotopes. In *HSCC'05*. Springer, 2005.

[9] A. Girard, C. L. Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. In *HSCC*, pages 257–271, 2006.

[10] A. Goldsztejn, D. Daney, M. Rueher, and P. Taillibert. Modal intervals revisited: a mean-value extension to generalized intervals. In *QCP'05*, 2005.

[11] A. Goldsztejng and W. Hayes. Rigorous inner approximation of the range of functions. In *SCAN*. IEEE Computer Society, 2006.

[12] E. Goubault, M. Kieffer, O. Mullier, and S. Putot. General inner approximation of vector-valued functions. *Reliable Computing*, 18:117–143, 2013.

[13] E. Goubault, O. Mullier, S. Putot, and M. Kieffer. Inner approximated reachability analysis. In *HSCC*, pages 163–172, 2014.

[14] E. Goubault and S. Putot. Under-approximations of computations in real numbers based on generalized affine arithmetic. In *SAS*, pages 137–152, 2007.

[15] D. Ishii, A. Goldsztejn, and C. Jermann. Interval-based projection method for under-constrained numerical systems. *Constraints*, 17(4):432–460, 2012.

[16] D. Ishii, N. Yonezaki, and A. Goldsztejn. Monitoring temporal properties using interval analysis. *IEICE Transactions*, 2016.

[17] E. Kaucher. Interval analysis in the extended interval space IR. *Comput. (Supplementum) 2*, 1980.

[18] S. Kaynama, J. Maidens, M. Oishi, I. M. Mitchell, and G. A. Dumont. Computing the viability kernel using maximal reachable sets. In *HSCC'12*. ACM, 2012.

[19] M. Korda, D. Henrion, and C. N. Jones. Inner approximations of the region of attraction for polynomial dynamical systems. In *NOLCOS*, 2013.

[20] A. B. Kurzhanski and P. Varaiya. Ellipsoidal toolbox. Technical report, EECS, Berkeley, May.

[21] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis: internal approximation. *Systems & control letters*, 2000.

[22] M. Lerch, G. Tischler, J. W. von Gudenberg, W. Hofschuster, and W. Kramer. filib++, a fast interval library supporting containment computations. *ACM Trans. Math. Soft.*, 2006.

[23] R. E. Moore. *Interval analysis*. 1966.

[24] N. S. Nedialkov. An interval solver for initial value problems in ordinary differential equations.

[25] N. S. Nedialkov, K. Jackson, and G. Corliss. Validated solutions of initial value problems for ordinary differential equations. *Appl. Math. Comp.*, 1999.

[26] M. A. B. Sassi, R. Testylier, T. Dang, and A. Girard. Reachability analysis of polynomial systems using linear programming relaxations. In *ATVA*, 2012.

[27] R. Testylier and T. Dang. NLTOOLBOX: A library for reachability computation of nonlinear dynamical systems. In *ATVA*, pages 469–473, 2013.

[28] B. Xue, Z. She, and A. Easwaran. Under-approximating backward reachable sets by polytopes. In *CAV*, 2016.

[29] P. Zgliczynski. C^1 Lohner algorithm. *Foundations of Computational Mathematics*, 2(4):429–465, 2002.

[30] A. Zutshi, S. Sankaranarayanan, J. V. Deshmukh, J. Kapinski, and X. Jin. Falsification of safety properties for closed loop control systems. In *HSCC*, pages 299–300, 2015.