An Interval Approach to Solve an Initial Value Problem

Thomas Le Mézo[1,a)], Luc Jaulin[1] and Benoît Zerr[1]

[1]*ENSTA-Bretagne LabSTICC 2 rue François Verny 29806 Brest France*

[a)]Corresponding author: thomas.le_mezo@ensta-bretagne.fr

Abstract. This paper proposes an original guaranteed interval-based method to solve an Initial Value Problem (IVP) for ordinary differential equations (ODE). Our method uses the geometrical properties of the vector field given by the ODE and a state space discretization to compute an enclosure of the trajectories set that verifies the IVP problem.

INTRODUCTION

Guaranteed integration is an elementary component for the study of dynamical systems and has been an important subject of research the last decades [1]. It is used from guaranteed system simulation [2] to the computation of viability kernel [3].

In this paper, we propose a method to solve guaranteed integration problems which is orthogonal to literature as we will not use the Picard operator nor any interval counterpart of integration method (Euler [4], Runge-Kutta [5] or Taylor [6]). The first section formalizes the problem; then the notion of *pathbubble* and new contractors are introduced; a method to solve the problem is presented; and a final section shows how the algorithm can be used on two test-cases.

FORMALIZATION OF THE PROBLEM

Let consider a Box Initial Value Problem (BIVP) $\mathcal{S}$ defined by the following equation:

$$\mathcal{S}: \begin{cases} \dot{\mathbf{x}}(t) & = \mathbf{f}(\mathbf{x}(t)), \\ \mathbf{x}(t_0) & = \mathbf{x}_0, \\ \mathbf{x}_0 & \in [\mathbf{x}_0], \\ t_0 & \in [t_0]. \end{cases} \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector and $\mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^n$ is the evolution function of $\mathcal{S}$ and where $[t_0]$ is an interval and $[\mathbf{x}_0]$ is a box. We can note that both here time and initial condition are uncertain.

Problem We want to find the set $\mathbb{S}$ of points solution of (1) such that:

$$\mathbb{S} = \{(t, \mathbf{x}) \mid \exists t_0 \in [t_0], \exists \mathbf{x}_0 \in [\mathbf{x}_0] \mid \mathbf{x} = \Phi(t - t_0, \mathbf{x}_0)\}, \tag{2}$$

where Φ is the flow map of $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$.

Hypothesis We know a search box $[\mathbf{r}]$ of $\mathbb{R}^n$ such that for all $\mathbf{x}(t)$ solution of the BIVP, $\forall t, \mathbf{x}(t) \in [\mathbf{r}]$.

PATHBUBBLE AND PROPAGATION CONTRACTORS

We introduce now the *pathbubble* object which is composed of three set : a *bubble* $\mathbb{X}$ which is a compact of $\mathbb{R}^n$, an input set $\mathbb{I} \subset \partial\mathbb{X}$ and an output set $\mathbb{O} \subset \partial\mathbb{X}$ where $\partial\mathbb{X}$ is the boundary of $\mathbb{X}$. A *pathbubble* will be noted $(\mathbb{X}, \mathbb{I}, \mathbb{O})$ and is a mathematical objects that will be used to support paths.

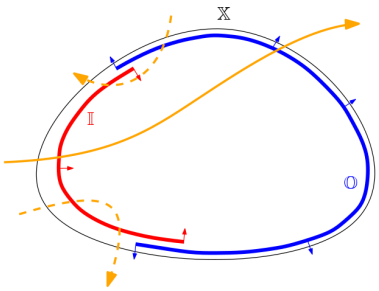

FIGURE 1. A *pathbubble* $(\mathbb{X}, \mathbb{I}, \mathbb{O})$ with a possible path (plain orange path) and an incompatible paths (dotted orange paths) according to the input and output set.

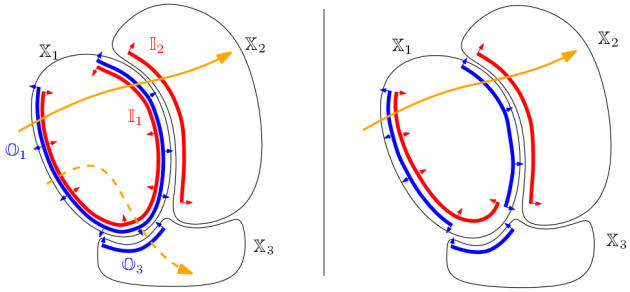

FIGURE 2. Illustration of the principle of continuity propagation contractor of a *pathbubbles* with its neighbors. Left, before the contraction, Right after the contraction. The plain orange path is a valid whereas the dashed path is incompatible with the border conditions. The corresponding contraction is $\mathbb{O}_1 := (\mathbb{I}_2 \cup \mathbb{I}_3 \cup \ldots) \cap \mathbb{O}_1$ and $\mathbb{I}_1 := (\mathbb{O}_2 \cup \mathbb{O}_3 \cup \ldots) \cap \mathbb{I}_1$.

Fig. 1 illustrates a *pathbubble* with its input set (in red) and output set (in blue). Dashed orange paths are incompatible with this *pathbubble* because they enter or leave the *pathbubble* in a wrong sense whereas the plain orange path.

Let $\mathcal{Q}$ be a paving of $\mathbb{R}^n$. We can define two contractors that will be applied to the *pathbubble* $(\mathbb{X}, \mathbb{I}, \mathbb{O})$, where $\mathbb{X} \in \mathcal{Q}$, in order to contract $\mathbb{I}$ and $\mathbb{O}$.

Continuity propagation contractor This contractor has the property to match the input set ($\mathbb{I}$) of a *pathbubble* to the output sets ($\mathbb{O}$) of its neighbor *pathbubbles*.

$$\mathbb{I}_{\mathbb{X}} = \bigcup_{\mathbb{Y} \in \mathcal{N}(\mathbb{X})} \mathbb{O}_{\mathbb{Y}} \cap \partial\mathbb{X} \tag{3}$$

where $\mathbb{S}_{\mathbb{X}}$ is the set (input or output) associated with the *pathbubble* $(\mathbb{X}, \mathbb{I}, \mathbb{O})$ and $\mathcal{N}(\mathbb{X})$ are the neighbor *pathbubbles* of $\mathbb{X}$ *i.e.* $\mathcal{N}(\mathbb{X}) = \{\mathbb{Y} \in \mathcal{Q} | \mathbb{X} \cap \mathbb{Y} \neq \emptyset \text{ or singleton}\}$. Fig. 2 shows an example of a *pathbubble* continuity contraction.

Flow contractor This contractor propagate the input set $\mathbb{I}$ of a *pathbubble* to its output set according to the geometrical direction given by the cone of consistent trajectories, $\mathbf{f}(\mathbb{X})$ which can be computed using interval arithmetics, and do also the backward operation (output to input). The corresponding contraction is:

$$\mathbb{O} = (\mathbb{I} + \{\mathbf{y} = \beta \cdot \mathbf{f}(\mathbf{x}), \mathbf{x} \in \mathbb{X}, \beta \geq 0\}) \cap \partial\mathbb{X}. \tag{4}$$

Fig. 3 shows the principle of a *pathbubble* flow contraction.

METHOD

In this section, we will provide an algorithm to solve our BIVP.

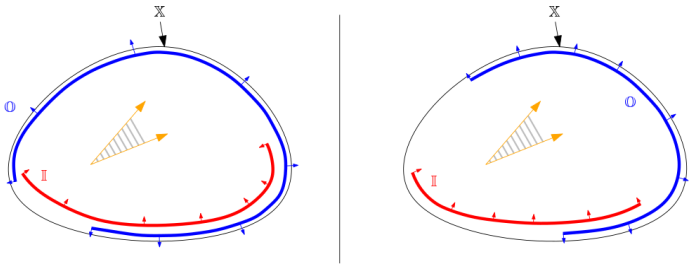

FIGURE 3. Illustration of the principle of flow propagation contractor of a *pathbubble*. Left, before the contraction, right after the contraction. The cone of possible path is shown in gray. A forward and a backward contraction has been realized for this example.

Algorithm We propose the following algorithm to compute an enclosure of $\mathcal{I}$:

1. Cover $\mathbb{R}^n$ with a paving $\mathcal{Q}$.
2. For each $\mathbb{X}$ of $\mathcal{Q}$, initialize the *pathbubble* with $\mathbb{I} = \partial\mathbb{X}$ and $\mathbb{O} = \partial\mathbb{X}$ except for paving that intersects the initial condition which will be initialized with $\mathbb{I} = \partial\mathbb{X}$ and $\mathbb{O} = \emptyset$.
3. For each *pathbubble*, apply the continuity contractor to match neighbor *pathbubble* output sets ($\mathbb{O}$) to the *pathbubble* input set ($\mathbb{I}$).
4. For each *pathbubble*, apply the flow propagation contractor.
5. Go to step 3 until a fix point is reached *i.e.* output and input sets of any *pathbubble* stay identical by applying contractors.
6. Bisect all boxes of $\mathcal{Q}$ associated with a non-empty *pathbubble* (i.e. $\mathbb{I}$ and $\mathbb{O} \neq \emptyset$).
7. Go to Step 3.

We can notice that the main idea of the method is that following vector field directions is sufficient to compute an outer approximation set. By introducing the time variable as a component of the state vector, we can generalize the approach: in the following test-case section, we will use this point of view.

TEST-CASES

In this section, we will illustrate our algorithm on two test-cases. We used a single core i5-2520M CPU@2.50GHz processor for all the computation. To solve our BIVP, we will use here a *pathbubble* where $\mathbb{X}$ is a box which can be represented on a computer.

Simple function integration

The first problem we will consider is to solve the following system

$$\dot{x} = -\sin x + e, \tag{5}$$

with $e \in [-0.1, 0.1]$. The initial condition is $t_0 \in [0.3, 0.5]$ and $x_0 \in [-0.5, 0.5]$ and the search set $\mathbf{r} \in [0, 5] \times [-2, 2]$.

The algorithm was stopped after 13 iterations which corresponds to less than 1 second of computation. At $t = 5.0$, the size of the x interval is inferior to $[-0.126, 0.126]$ as shown on Fig. 4.

Car on the hill problem

Consider the car on the hill problem [7] given by the following equation:

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -9.81 \sin\left(\frac{\left(\frac{1.1}{1.2}\sin(x_1) - 1.2\sin(1.1x_1)\right)}{2.0}\right) - 0.7x_2 + 2.0 \end{cases} \tag{6}$$

which corresponds to the evolution of a car on a hill (see Fig. 5). We started the car at the initial condition $\mathbf{x}_0 \in [-1, 1] \times [-1, 1]$ and stopped the algorithm after 18 iterations which takes less than 11 seconds. Fig. 5 shows the result of the algorithm in the state space. Due to uncertainty of the initial condition, we can see that there is two possibilities for the car when it arrives at the last third hill: to continue ahead and exit the figure or falls back into the valley toward an equilibrium. We can see, on Fig. 5, that our method is able to separate the two possible trajectories of the car.

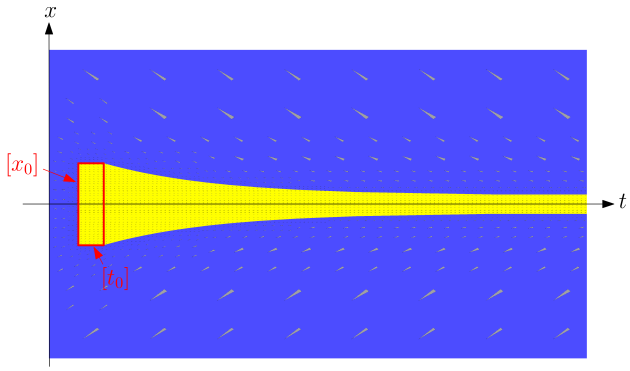

FIGURE 4. Results of the algorithm in the state space. The yellow shape corresponds to the outer approximation of valid trajectories and gray cones correspond to the vector field in each *pathbubblees*. The initial condition is the red box and the search set $\mathbf{r} \in [0,5] \times [-2,2]$.

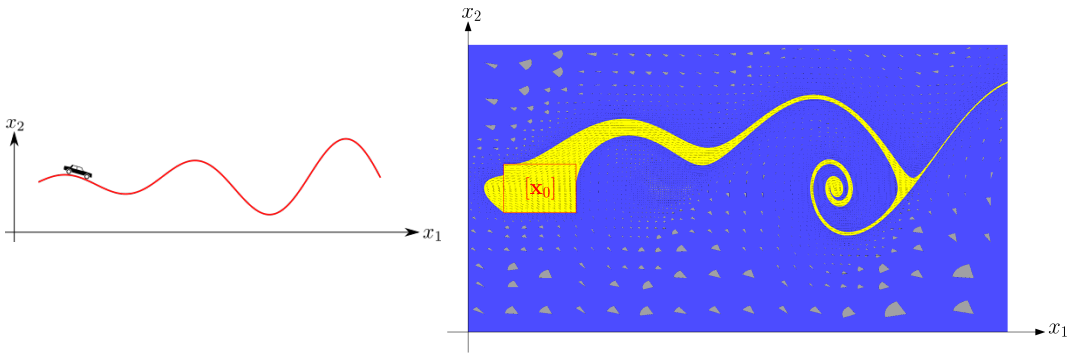

FIGURE 5. Right: The shape of the car on the hill problem. Left:Results of the algorithm in the state space. The search set is $\mathbf{r} \in [-2,13] \times [-6,6]$, the color code is the same as Fig. 4. The Initial condition is on the left part of the figure

CONCLUSION

In this paper, we have proposed a new approach to compute an enclosure set of paths that verify a Box Initial Valued Problem for Ordinary Differential Equation. The main idea of our method is to follow the geometrical directions given by the vector field of the ODE. Whereas existing approaches which need a discretization of time, our method is only based on a discretization of the space state. The principle of our method has been illustrated on the guaranteed integration of the car on the hill problem and on a sinusoidal function.

REFERENCES

[1] M. Berz, *Computational differentiation: techniques, applications, and tools*, 89 (Society for Industrial & Applied, 1996).

[2] M. Konečnỳ, W. Taha, J. Duracz, A. Duracz, and A. Ames, "Enclosing the behavior of a hybrid system up to and beyond a zeno point," in *Cyber-Physical Systems, Networks, and Applications (CPSNA), 2013 IEEE 1st International Conference on* (IEEE, 2013), pp. 120–125.

[3] M. Lhommeau, L. Jaulin, and L. Hardouin, International Journal of Adaptive Control and Signal Processing **25**, 264–272 (2011).

[4] R. E. Moore and F. Bierbaum, *Methods and applications of interval analysis*, Vol. 2 (SIAM, 1979).

[5] J. A. dit Sandretto and A. Chapoutot, Reliable Computing **22**, p. 79 (2016).

[6] N. Revol, K. Makino, and M. Berz, Journal of Logic and Algebraic Programming **64**, 135–154 (2005).

[7] M. Lhommeau, L. Jaulin, and L. Hardouin, International Journal of Adaptative Control and Signal Processing **25**, 264–272 (2011).