

Lagrangian Reachability

Jacek Cyranka^{1*}, Md. Ariful Islam^{2*}, Greg Byrne³, Paul Jones⁴,
Scott A. Smolka⁴, Radu Grosu⁵

¹ Rutgers University {jcyranka@gmail.com}

² Carnegie Mellon University {mdarifui@cs.cmu.edu}

³ Metron Inc., Reston, VA {gr3g.byrne@gmail.com}

⁴ US Food and Drug Administration {paul.jones@fda.hhs.gov}

⁵ Stony Brook University {sas@cs.stonybrook.edu}

⁶ Vienna University of Technology {radu.grosu@tuwien.ac.at}

Abstract. We introduce LRT, a new Lagrangian-based ReachTube computation algorithm that conservatively approximates the set of reachable states of a nonlinear dynamical system. LRT makes use of the Cauchy-Green stretching factor (SF), which is derived from an over-approximation of the gradient of the solution-flows. The SF measures the discrepancy between two states propagated by the system solution from two initial states lying in a well-defined region, thereby allowing LRT to compute a reachtube with a ball-overestimate in a metric where the computed enclosure is as tight as possible. To evaluate its performance, we implemented a prototype of LRT in C++/Matlab, and ran it on a set of well-established benchmarks. Our results show that LRT compares very favorably with respect to the CAPD and Flow* tools.

1 Introduction

Bounded-time reachability analysis is an essential technique for ensuring the safety of emerging systems, such as cyber-physical systems (CPS) and controlled biological systems (CBS). However, computing the reachable states of CPS and CBS is a very difficult task as these systems are most often nonlinear, and their state-space is uncountably infinite. As such, these systems typically do not admit a closed-form solution that can be exploited during their analysis.

For CPS/CBS, one can therefore only compute point solutions (trajectories) through numerical integration and for predefined inputs. To cover the infinite set of states reachable by the system from an initial region, one needs to conservatively extend (symbolically surround) these pointwise solutions by enclosing them in *reachtubes*. Moreover, the starting regions of these reachtubes have to cover the initial states region.

The class of continuous dynamical systems we are interested in this paper are nonlinear, time-variant ordinary differential equations (ODEs):

$$x'(t) = F(t, x(t)), \tag{1a}$$

$$x(t_0) = x_0, \tag{1b}$$

* Both authors contributed equally to this work.

where $x: \mathbb{R} \rightarrow \mathbb{R}^n$. We assume that F is a smooth function, which guarantees short-time existence of solutions. The class of time-variant systems includes the class of time-invariant systems. Time-variant equations may contain additional terms, e.g. an excitation variable, and/or periodic forcing terms.

For a given initial time t_0 , set of initial states $\mathcal{X} \subset \mathbb{R}^n$, and time bound $T > t_0$, our goal is to compute conservative *reachtube* of (1), that is, a sequence of time-stamped sets of states $(R_1, t_1), \dots, (R_k, t_k)$ satisfying:

$$\text{Reach}((t_0, \mathcal{X}), [t_{i-1}, t_i]) \subset R_i \text{ for } i = 1, \dots, k,$$

where $\text{Reach}((t_0, \mathcal{X}), [t_{i-1}, t_i])$ denotes the set of reachable states of (1) in the interval $[t_{i-1}, t_i]$. Whereas there are many sets satisfying this requirement, of particular interest to us are reasonably tight reachtubes; i.e. reachtubes whose over-approximation is the tightest possible, having in mind the goal of proving that a certain region of the phase space is (un)safe, and avoiding false positives. In practice and for the sake of comparison with other methods, we compute a discrete-time reachtube; as we discuss, a continuous reachtube can be obtained using our algorithm.

Existing tools and techniques for conservative reachtube computation can be classified by the time-space approximation they perform into three categories: (1) Taylor-expansion in time, variational-expansion in space (wrapping-effect reduction) of the solution set (CAPD [4,32,33], VNode-LP [26,25]), (2) Taylor-expansion in time and space of the solution set (Cosy Infinity [21,3,22], Flow* [5,6]), and (3) Bloating-factor-based and discrepancy-function-based [10,11]. The last technique computes a conservative reachtube using a *discrepancy function* (DF) that is derived from an over-approximation of the *Jacobian of the vector field* (usually given by the RHS of the differential equations) defining the continuous dynamical system.

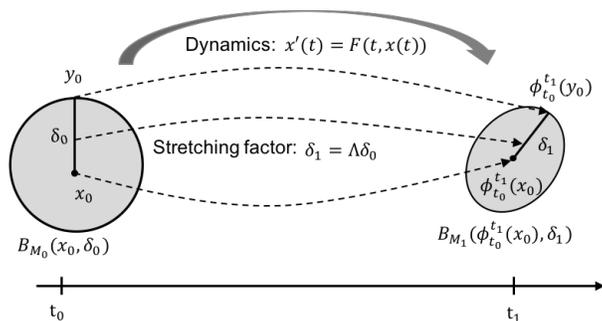


Fig. 1: An overview of LRT. The figure shows one execution step of the LRT described in detail in Section 3. The dashed arrows reflect the solution flow ϕ and the evolution of state discrepancy.

This paper proposes an alternative (and orthogonal to [10,11]) technique for computing a conservative reachtube, by using a *stretching factor* (*SF*) that is derived from an over-approximation of the *gradient of the solution-flows*, also

referred to as the *sensitivity matrix* [8,9], and the *deformation tensor*. An illustration of our method is given in Figure 1. $B_{M_0}(x_0, \delta_0)$ is a well-defined initial region given as a ball in metric space M_0 centered at x_0 of radius δ_0 . The SF A measures the discrepancy of two states x_0, y_0 in B_{M_0} propagated by the solution flow induced by (1), i.e. $\phi_{t_0}^{t_1}$. We can thus use the SF to bound the infinite set of reachable states at time t_1 with the ball-overestimate $B_{M_1}(\phi_{t_0}^{t_1}(x_0), \delta_1)$ in an appropriate metric (which may differ from the initial M_0), where $\delta_1 = A \cdot \delta_0$. Similar to [10], this metric is based on a weighted norm, yielding a tightest-possible enclosure of the reach-set [7,19,17,20]. For two-dimensional system, we present an analytical method to compute M_1 , but for higher dimensional system, we solve a semi-definite optimization problem. Analytical formulas derived for 2d case allow for faster computation. We point out that the output provided by LRT can be used to compute a validated bound for the so-called *finite-time Lyapunov exponent* ($FTLE = \frac{1}{T} \ln(SF)$) for a whole set of solutions. FTLE are widely computed in e.g. climate research in order to detect Lagrangian coherent structures.

We call this approach and its associated the LRT, for *Lagrangian Reachtube* computation. The LRT uses analogues of Cauchy-Green deformation tensors (CGD) from finite strain theory (FST) to determine the SF of the solution-flows, after each of its time-step iterations. The LRT algorithm is described thoroughly in Section 3.

To compute the gradient of the flows, we currently make use of the CAPD C^1 routine, which propagates the initial ball (box) using interval arithmetic. The CAPD library has been certified to compute a conservative enclosure of the true solution, and it has been used in many peer-reviewed computer proofs of theorems in dynamical systems, including [12,15,31].

To evaluate the LRT's performance, we implemented a prototype in C++/Matlab and ran it on a set of eight benchmarks. Our results show that the LRT compares very favorably to a direct use of CAPD and Flow* (see Section 4), while still leaving room for further improvement. In general, we expect the LRT to behave favorably on systems that exhibit long-run stable behavior, such as orbital stability.

We did not compare the LRT with the DF-based tools [11,10], although we would have liked to do this very much. The reason is that the publicly-available DF-prototype has not yet been certified to produce conservative results. Moreover, the prototype only considers time-invariant systems.

The rest of the paper is organized as follows. Section 2 reviews finite-strain theory and the Cauchy-Green-deformation tensor for flows. Section 3 presents the LRT, our main contribution, and proves that it conservatively computes the reachtube of a Cauchy system. Section 4 compares our results to CAPD and Flow* on six benchmarks from [6,10], the forced Van der Pol oscillator (time-variant system) [30], and the Mitchell Schaeffer cardiac cell model [23]. Section 5 offers our concluding remarks and discusses future work.

2 Background on Flow Deformation

In this section we present some background on the LRT. First, in Section 2.1 we briefly recall the general FST, as in the LRT we deal with matrices analogous to Cauchy-Green deformation tensors. Second, in Section 2.2 we show how the Cauchy-Green deformation tensor can be used to measure discrepancy of two initial states propagated by the flow inducted by Eq. (1).

2.1 Finite Strain Theory and Lagrangian Description of the Flow

In classical continuum mechanics, *finite strain theory (FST)* deals with the deformation of a continuum body in which both rotation and strain can be arbitrarily large. Changes in the configuration of the body are described by a displacement field. Displacement fields relate the initial configuration with the deformed configuration, which can be significantly different. FST has been applied, for example, in stress/deformation analysis of media like elastomers, plastically-deforming materials, and fluids, modeled by constitutive models (see e.g., [13], and the references provided there). In the *Lagrangian* representation, the coordinates describe the deformed configuration (in the material-reference-frame spatial coordinates), whereas in the *Eulerian* representation, the coordinates describe the undeformed configuration (in a fixed-reference-frame spatial coordinates).

Notation In this section we use the standard notation used in the literature on FST. We use X to denote the position of a particle in the Eulerian coordinates, and x to denote the position of a particle in the Lagrangian coordinates. The Lagrangian coordinates depend on the initial (Eulerian) position, and the time t , so we use $x(X, t)$ to denote the position of a particle in Lagrangian coordinates.

The *displacement field* from the initial configuration to the deformed configuration in Lagrangian coordinates is given by the following equation:

$$u(X, t) = x(X, t) - X. \quad (2)$$

The dependence $\nabla_X u$ of the displacement field $u(X, t)$ on the initial condition X is called the *material displacement gradient tensor*, with

$$\nabla_X u(X, t) = \nabla_X x(X, t) - I, \quad (3)$$

where $\nabla_X x$ is called the *deformation gradient tensor*.

We now investigate how an initial perturbation $X + dX$ in the Eulerian coordinates evolves to the deformed configuration $dx(X + dX, t)$ in Lagrangian coordinates by using (2). This is called a *relative displacement vector*:

$$dx(X + dX, t) = x(X + dX, t) - x(X, t) = u(X + dX, t) + dX - u(X, t),$$

As a consequence, for small dX we obtain the following approximate equality:

$$dx(X + dX, t) \approx u(X + dX, t) - u(X, t). \quad (4)$$

Now let us compute $u(X+dX, t)$ by expressing $du(X+dX, t)$ as with $x(X+dX, t)$ above. One obtains:

$$u(X + dX, t) = u(X, t) + du(X + dX, t) = u(X, t) + \nabla_X u(X + dX, t)dX.$$

Now by replacing $u(X + dX, t)$ in Equation (4) above, one obtains the following result:

$$dx(X + dX, t) \approx \nabla_X x(X + dX, t)dX.$$

Several rotation-independent tensors have been introduced in the literature. Classical examples include the right Cauchy-Green deformation tensor:

$$C = (\nabla_X x)^T \cdot \nabla_X x. \quad (5)$$

2.2 Cauchy-Green Deformation Tensor for Flows

Notation By $[x]$ we denote a product of intervals (a box), i.e. a compact and connected set $[x] \subset \mathbb{R}^n$. We will use the same notation for interval matrices. By $\|\cdot\|_2$ we denote the *Euclidean norm*, by $\|\cdot\|_\infty$ we denote the max norm, we use the same notation for the inducted operator norms. Let $B(x, \delta)$ denote the closed ball centered at x with the radius δ . It will be clear from the context in which metric space we consider the ball. By $\phi_{t_0}^{t_1}$ we denote the flow inducted by (1), by $D_x \phi_{t_0}^{t_1}$ we denote the partial derivative in x of the flow with respect to the initial condition, at time t_1 , which we call the *gradient of the flow*, also referred to as the *sensitivity matrix* [8,9].

Let us now relate the finite strain theory presented in Section 2.1 to the study of flows inducted by the ODE (1). For expressing deformation in time of a continuum we first consider the set of initial conditions (e.g. a ball), which is being evolved (deformed) in time by the flow ϕ . For the case of flows we have that the positions in Eulerian coordinates are coordinates of the initial condition (denoted here using lower case letters with subscript 0, i.e. x_0, y_0). The equivalent of $x(X, t)$ – the Lagrangian coordinates of X at time t is $\phi_{t_0}^t(x_0)$. Obviously, the equivalent of $u(X, t)$ is $(\phi_{t_0}^t(x_0) - x_0)$, and the deformation gradient $\nabla_X x$ here is just the derivative of the flow with respect to the initial condition $D_x \phi_{t_0}^t$ (sensitivity matrix).

In this section we show that deformation tensors arise in a study of discrepancy of solutions of (1). First, we provide some basic lemmas that we use in the analysis of our reachtube-computation algorithm. We work with the metric spaces that are based on weighted norms. We aim at finding weights, such that the inducted *matrix measures* (also known as *logarithmic norms* for particular cases) [7,19,17,20] provides to be smaller than those in the Euclidean norm for the computed gradients of solutions.

Definition 1. *Given positive-definite symmetric matrix $M \in \mathbb{R}^{n \times n}$ we define the M -norm of \mathbb{R}^n vectors by*

$$\|y\|_M = \sqrt{y^T M y}. \quad (6)$$

Given the decomposition

$$M = C^T C,$$

the matrix norm inducted by (6) is

$$\|A\|_M = \sqrt{\lambda_{max}((C^T)^{-1} \cdot A^T \cdot M \cdot A \cdot C^{-1})}, \quad (7)$$

where $\lambda_{max}(\cdot)$ denotes the maximal eigenvalue of the matrix A .

Observe that the square-root is well defined, as $\lambda_{max} > 0$, M is symmetric positive definite, and hence, the matrix $(C^T)^{-1} \cdot A^T \cdot M \cdot A \cdot C^{-1}$ is also symmetric positive definite.

Lemma 1. Consider the Cauchy problem (1). Let $x_0, y_0 \in \mathbb{R}^n$ be two initial conditions at time t_0 . Let $M \in \mathbb{R}^{n \times n}$ be a positive-definite symmetric matrix and $C^T C = M$ be its decomposition. For $t_1 \geq t_0$, it holds that

$$\|\phi_{t_0}^{t_1}(x_0) - \phi_{t_0}^{t_1}(y_0)\|_M \leq \sqrt{\lambda_{max}((C^T)^{-1} D_x \phi_{t_0}^{t_1}(\xi)^T M D_x \phi_{t_0}^{t_1}(\xi) C^{-1})} \|x_0 - y_0\|_M \quad (8)$$

where $\xi = \omega x_0 + (1 - \omega)y_0$ for some $\omega \in [0, 1]$. For the particular case of the Euclidean norm, (8) takes the form

$$\|\phi_{t_0}^{t_1}(x_0) - \phi_{t_0}^{t_1}(y_0)\|_2 \leq \sqrt{\lambda_{max}((D_x \phi_{t_0}^{t_1}(\xi))^T \cdot D_x \phi_{t_0}^{t_1}(\xi))} \|x_0 - y_0\|_2. \quad (9)$$

A proof can be found in Appendix A.

Remark 1. Let $\xi \in \mathbb{R}$ be a given vector. Observe that $(D_x \phi_{t_0}^{t_1}(\xi))^T \cdot D_x \phi_{t_0}^{t_1}(\xi)$ appearing in (9) is the right *Cauchy-Green deformation tensor* (5) for two given initial vectors x_0 and y_0 . We call the value $\sqrt{\lambda_{max}}$ appearing in (9) *Cauchy-Green stretching factor* for given initial vectors x_0 and y_0 , which is necessarily positive as the CG deformation tensor is positive definite $\sqrt{\lambda_{max}(\cdot)} > 0$.

Lemma 1 is used when both of the discrepancy of the solutions at time t_1 as well as the initial conditions is measured in the same M -norm. In the practical Lagrangian Reachtube Algorithm the norm used is being changed during the computation. Hence we need another version of Lemma 1, where the norm in which the discrepancy of the initial condition is measured differs from the norm in which the discrepancy of the solutions at time t_1 is measured.

Lemma 2. Consider the Cauchy problem (1). Let $x_0, y_0 \in \mathbb{R}^n$ be two initial conditions at time t_0 . Let $M_0, M_1 \in \mathbb{R}^{n \times n}$ be positive-definite symmetric matrices, and $C_0^T C_0 = M_0$, $C_1^T C_1 = M_1$ their decompositions respectively. For $t_1 \geq t_0$, it holds that

$$\begin{aligned} & \|\phi_{t_0}^{t_1}(x_0) - \phi_{t_0}^{t_1}(y_0)\|_{M_1} \\ & \leq \sqrt{\lambda_{max}((C_0^T)^{-1} \cdot (D_x \phi_{t_0}^{t_1}(\xi))^T \cdot M_1 \cdot D_x \phi_{t_0}^{t_1}(\xi) \cdot C_0^{-1})} \|x_0 - y_0\|_{M_0}, \quad (10) \end{aligned}$$

where $\xi = \omega x_0 + (1 - \omega)y_0$ for some $\omega \in [0, 1]$.

A proof can be found in Appendix A.

Remark 2. Given a positive-definite symmetric matrix M . We call the value appearing in (10) $(C_0^T)^{-1} \cdot (D_x \phi_{t_0}^{t_1}(\xi))^T \cdot M_1 \cdot D_x \phi_{t_0}^{t_1}(\xi) \cdot C_0^{-1}$ as the M_0/M_1 -*deformation tensor*, and the value $\sqrt{\lambda_{max}}$ as the M_0/M_1 -*stretching factor*.

The idea behind using weighted norms in our approach is that the stretching factor in M -norm (10) is expected to be smaller than that in the Euclidean norm (9). Ultimately, this permits a tighter reachtube computation, whose complete procedure is presented in Section 3.3.

3 Lagrangian Reachtube Computation

3.1 Reachtube Computation: Problem-Statement

In this section we provide first some lemmas that we then use to show that our method-and-algorithm produces a conservative output, in the sense that it encloses the set of solutions starting from a set of initial conditions. Precisely, we define what we mean by conservative enclosures.

Definition 2. *Given an initial set \mathcal{X} , initial time t_0 , and the target time $t_1 \geq t_0$. We call the following compact sets:*

- $\mathcal{W} \subset \mathbb{R}^n$ a conservative, reach-set enclosure, if $\phi_{t_0}^{t_1}(x) \in \mathcal{W}$ for all $x \in \mathcal{X}$.
- $\mathcal{D} \subset \mathbb{R}^{n \times n}$ a conservative, gradient enclosure, if $D_x \phi_{t_0}^{t_1}(x) \in \mathcal{D}$ for all $x \in \mathcal{X}$.

Following the notation used in [10], and extending the corresponding definitions to our time variant setting, we introduce the notion of reachability as follows: Given an initial set $\mathcal{X} \subset \mathbb{R}^n$ and a time t_0 , we call a state x in \mathbb{R}^n as *reachable* within a time interval $[t_1, t_2]$, if there exists an initial state $x_0 \in \mathcal{X}$ at time t_0 and a time $t \in [t_1, t_2]$, such that $x = \phi_{t_0}^t(x_0)$. The set of all reachable states in the interval $[t_1, t_2]$ is called the *reach set* and is denoted by $\text{Reach}((t_0, \mathcal{X}), [t_1, t_2])$.

Definition 3 ([10] Def. 2.4). *Given an initial set \mathcal{X} , initial time t_0 , and a time bound T , a $((t_0, \mathcal{X}), T)$ -reachtube of System (1) is a sequence of time-stamped sets $(R_1, t_1), \dots, (R_k, t_k)$ satisfying the following: (1) $t_0 \leq t_1 \leq \dots \leq t_k = T$, (2) $\text{Reach}((t_0, \mathcal{X}), [t_{i-1}, t_i]) \subset R_i, \forall i = 1, \dots, k$.*

Definition 4. *Whenever the initial set \mathcal{X} and the time horizon T are known from the context we will skip the $(\mathcal{X}, [t_0, T])$ part, and will simply use the name (conservative) reachtube over-approximation of the flow defined by (1).*

Observe that we do not address here the question what is the exact structure of the solution set at time t initiating at \mathcal{X} . In general it could have, for instance, a fractal structure. We aim at constructing an over-approximation for the solution set, and its gradient, which is amenable for rigorous numeric computations.

In the theorems below we show that the method presented in this paper can be used to construct a reachtube over-approximation \mathcal{R} . The theorems below is the foundation of our novel *Lagrangian Reachtube Algorithm (LRT)* presented in Section 3.3. In particular, the theorems below provide estimates we use for constructing \mathcal{R} . First, we present a theorem for the discrete case.

3.2 Conservative Reachtube Construction

Theorem 1. *Let $t_0 \leq t_1$ be two time points. Let $\phi_{t_0}^{t_1}(x)$ be the solution of (1) with the initial condition (t_0, x) at time t_1 , let $D_x \phi_{t_0}^{t_1}$ be the gradient of the flow. Let $M_0, M_1 \in \mathbb{R}^{n \times n}$ be positive-definite symmetric matrices, and $C_0^T C_0 = M_0$, $C_1^T C_1 = M_1$ be their decompositions respectively. Let $\mathcal{X} = B_{M_0}(x_0, \delta_0) \subset \mathbb{R}^n$ be a set of initial states for the Cauchy problem (1) (ball in M_0 -norm with the center at x_0 , and radius δ_0). Assume that there exists a compact, conservative enclosure $\mathcal{D} \subset \mathbb{R}^{n \times n}$ for the gradients, such that:*

$$D_x \phi_{t_0}^{t_1}(x) \in \mathcal{D} \text{ for all } x \in \mathcal{X}. \quad (11)$$

Suppose $\Lambda > 0$ is such that:

$$\Lambda \geq \sqrt{\lambda_{\max} \left((C_0^T)^{-1} D^T M_1 D C_0^{-1} \right)}, \text{ for all } D \in \mathcal{D}. \quad (12)$$

Then it holds that:

$$\phi_{t_0}^{t_1}(x) \in B_{M_1}(\phi_{t_0}^{t_1}(x_0), \Lambda \cdot \delta_0).$$

Proof. Let x_0 be the center of the ball of initial conditions $\mathcal{X} = B_{M_0}(x_0, \delta_0)$, and let us pick $x \in \mathcal{X}$. From Lemma 4 the discrepancy of the solutions initiating at x_0 and x at time t_1 is bounded in M_1 -norm by:

$$\|\phi_{t_0}^{t_1}(x_0) - \phi_{t_0}^{t_1}(x)\|_{M_1} \leq \delta_0 \sqrt{\lambda_{\max} \left((C_0^T)^{-1} \cdot (D_x \phi_{t_0}^{t_1}(\xi))^T \cdot M_1 \cdot D_x \phi_{t_0}^{t_1}(\xi) \cdot C_0^{-1} \right)},$$

where $\xi = \omega x_0 + (1 - \omega)x$ for some $\omega \in [0, 1]$. Obviously, $\xi \in B_{M_0}(x_0, \delta_0)$. Hence, $D_x \phi_{t_0}^{t_1}(\xi) \in \mathcal{D}$. Moreover, if $\Lambda > 0$ satisfies (12), then

$$\Lambda \geq \sqrt{\lambda_{\max} \left((C_0^T)^{-1} \cdot (D_x \phi_{t_0}^{t_1}(\xi))^T \cdot M_1 \cdot D_x \phi_{t_0}^{t_1}(\xi) \cdot C_0^{-1} \right)},$$

and $\phi_{t_0}^{t_1}(x) \in B_{M_1}(\phi_{t_0}^{t_1}(x_0), \Lambda \delta_0)$. As x was chosen in an arbitrary way, we are done. \square

The next theorem is the variant of Theorem 1 for obtaining a continuous reachtube.

Theorem 2. *Let $\phi_{t_0}^{t_1}(x)$ be the solution of (1) with the initial condition (t_0, x) at time t_1 , let $D_x \phi_{t_0}^{t_1}$ be the gradient of the flow. Let $M_0, M_1 \in \mathbb{R}^{n \times n}$ be positive definite symmetric matrices, and $C_0^T C_0 = M_0$, $C_1^T C_1 = M_1$ be their decompositions respectively. Let $\mathcal{X} = B_{M_0}(x_0, \delta_0) \subset \mathbb{R}^n$ be a set of initial conditions for the Cauchy problem (1) (ball in M_0 -norm). Assume that there exists $\{\mathcal{D}_t\}_{t \in [t_0, t_1]}$ - a compact t -parametrized set, such that*

$$\begin{aligned} \mathcal{D}_t &\subset \mathbb{R}^{n \times n} \text{ for } t \in [t_0, t_1], \\ D_x \phi_{t_0}^t(x) &\in \mathcal{D}_t \text{ for all } x \in \mathcal{X}, \text{ and } t \in [t_0, t_1]. \end{aligned}$$

If $\Lambda > 0$ is such that for all $D: D \in \mathcal{D}_t$ for some $t \in [t_0, t_1]$

$$\Lambda \geq \sqrt{\lambda_{\max}((C_0^T)^{-1}D^T M_1 D C_0^{-1})}$$

Then for all $t \in [t_0, t_1]$ it holds that

$$\phi_{t_0}^t(x) \in B_{M_1}(\phi_{t_0}^t(x_0), \Lambda \cdot \delta_0). \quad (13)$$

Proof. It follows from the proof of Theorem 1 applied to all times $t \in [t_0, t_1]$. \square

Corollary 1. Let $T \geq t_0$. Assume that there exists $\{\mathcal{D}_t\}_{t \in [t_0, T]}$ – a compact t -parametrized set, such that

$$\begin{aligned} \mathcal{D}_t &\subset \mathbb{R}^{n \times n} \text{ for } t \in [t_0, T], \\ D_x \phi_{t_0}^t(x) &\in \mathcal{D}_t \text{ for all } x \in \mathcal{X}, \text{ and } t \in [t_0, T]. \end{aligned}$$

Then the existence of a $((t_0, X), T)$ -reachtube of the system described in Eq. (1) in sense of Definition 3, i.e. a sequence of time-stamped sets $(R_1, t_1), \dots, (R_k, t_k)$ is provided by an application of Lemma 2. We provide an algorithm computing the reachtube in Section 3.3.

Proof. Immediate application of Theorem 2 shows that if the first segment (R_1, t_1) is defined

$$(R_1, t_1) := \bigcup_{t \in [t_0, t_1]} B_{M_1}(\phi_{t_0}^t(x_0), \Lambda \cdot \delta_0),$$

then it satisfies

$$\text{Reach}((t_0, \mathcal{X}), [t_0, t_1]) \subset (R_1, t_1),$$

which is exactly provided by (13). The j -th segment (R_j, t_j) for $j = 2, \dots, k$ is obtained by replacing in Theorem 2 the time interval $[t_0, t_1]$ with the interval $[t_{j-1}, t_j]$ (Observe that the norm may be different in each step). \square

3.3 LRT: A Rigorous Lagrangian Computation Algorithm

We now present a complete description of our algorithm. In the next section we prove its correctness. First let us comment on how in practice we compute a representable enclosure for the gradients (11).

- First, given an initial ball $B_{M_0}(x_0, \delta_0)$ we compute its representable over-approximation, i.e., a product of intervals (a box in canonical coordinates $[X] \subset \mathbb{R}^n$), such that $B_{M_0}(x_0, \delta_0) \subset [X]$.
- Next, using the C^1 -CAPD algorithm [33,32] all trajectories initiating in $[X]$ are rigorously propagated forward in time, in order to compute a conservative enclosure for $\{D_x \phi_{t_0}^{t_1}(\xi) \mid \xi \in [X]\}$, the gradients. We use the notation $[D_x \phi_{t_0}^{t_1}([X])] \subset \mathbb{R}^{n \times n}$, to denote a representable enclosure (an interval matrix) for the set of gradients $\{D_x \phi_{t_0}^{t_1}(\xi) \mid \xi \in [X]\}$.

The norm of an interval vector $\| [x] \|$ is defined as the supremum of the norms of all vectors within the bounds $[x]$. For an interval set $[x] \subset \mathbb{R}^n$ we denote by $\mathbb{R}^n \supset B_M([x], r) := \bigcup_{x \in [x]} B_M(x, r)$, the union of the balls in M -norm of radius r having the center in $[x]$. Each product of two interval matrices is overestimated by using the interval-arithmetic operations.

Definition 5. We will call the rigorous tool the C^1 -CAPD algorithm, which is currently used to generate conservative enclosures for the gradient in the LRT.

The output of the LRT are discrete-time reachtube over-approximation cross-sections $\{t_0, t_1, \dots, t_k\}$, $t_k = t_0 + kT$, i.e., reachtube over-approximations $B_{M_j}(x_j, \delta_j)$ of the flow induced by (1) at time t_j . We note that the algorithm can be easily modified to provide a validated bounds for the finite-time Lyapunov exponent. We use the discrete-time output for the sake of comparison. However, as a byproduct, a continuous reachtube over-approximation is obtained by means of *rough enclosures* (Fig. 2) produced by the rigorous tool used and by applying Theorem 2. The implementation details of the algorithm can be found in Section 4.

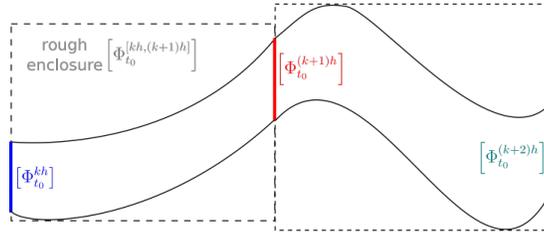


Fig. 2: Fine bounds provided at equally spaced time steps (colored), and coarse bounds provided for the intermediate times by the rough enclosure (dotted boxes).

We are now ready to give the formal description of the LRT: (1) its inputs, (2) its outputs, and (3) its computation.

LRT: The Lagrangian Reachtube Algorithm

Input:

- ODEs (1): time-variant ordinary differential equations,
- T : time horizon, t_0 : initial time, k : number of steps, $h = T/k$: time step
- M_0 : initial positive-definite symmetric matrix defining the norm (6),
- $[x_0] \subset \mathbb{R}^n$: the initial bounds for the position of the center of the ball at t_0 ,
- $\delta_0 > 0$: the radius of the ball (in M_0 norm) about x_0 at t_0 .

Output:

- $\{[x_j]\}_{j=1}^k \subset \mathbb{R}^{n \times k}$: interval enclosures for ball centers x_j at time $t_0 + jh$,

- $\{M_j\}_{j=1}^k$: norms defining metric spaces for the ball enclosures,
- $\{\delta_j\}_{j=1}^k \in \mathbb{R}_+^k$: radiuses of the ball enclosures at x_j for $j = 1, \dots, k$.⁷

Begin LRT

1. Set $t_1 = t_0 + h$. Propagate the center of the ball $[x_0]$ forward in time by the time-step h , using the rigorous tool. The result is a conservative enclosure for the solutions $[\phi_{t_0}^{t_1}([x_0])]$ and the gradients $[D_x \phi_{t_0}^{t_1}([x_0])]$.
2. Choose a matrix $D \in [D_x \phi_{t_0}^{t_1}([x_0])]$, and compute a symmetric positive-definite matrix M_1 , and its decomposition $M_1 = C_1^T C_1$, such that, it minimizes the stretching factor for D . In other words, it holds that:

$$\sqrt{\lambda_{max} \left((C_1^T)^{-1} D^T M_1 D C_1^{-1} \right)} \leq \sqrt{\lambda_{max} \left((\tilde{C}^T)^{-1} D^T \tilde{M} D \tilde{C}^{-1} \right)}, \quad (14)$$

for all positive-definite symmetric matrices \tilde{M} . In the actual code we find the minimum with some resolution, i.e., we compute M_1 , such that it is close to the optimal in the sense of (18), using the procedure presented in subsection 3.6.

3. Decide whether to change the norm of the ball enclosure from M_0 to M_1 (if it leads to a smaller stretching factor). If the norm is to be changed keep M_1 as it is, otherwise $M_1 = M_0$,
4. Compute an over-approximation for $B_{M_0}([x_0], \delta_0)$, which is representable in the rigorous tool employed by the LRT, and can be used as input to propagate forward in time all solutions initiating in $B_{M_0}([x_0], \delta_0)$. This is a product of intervals in canonical coordinates $[X] \subset \mathbb{R}^n$, such that:

$$B_{M_0}([x_0], \delta_0) \subset [X].$$

We compute the over-approximation using the interval arithmetic expression:

$$C_0^{-1} (C_0 \cdot [x_0] + [-\delta_0, \delta_0]^n)$$

5. Rigorously propagate $[X]$ forward in time, using the rigorous tool over the time interval $[t_0, t_1]$. The result is a continuous reachtube, providing bounds for $[\Phi_{t_0}^t([X])]$, and $[D_x \Phi_{t_0}^t([X])]$ for all $t \in [t_0, t_1]$. We employ an integration algorithm with a fixed time-step h . As a consequence "fine" bounds are obtained for $t = t_1$. We denote those bounds by:

$$[\phi_{t_0}^{t_1}([X])], \quad [D_x \phi_{t_0}^{t_1}([X])].$$

⁷ Observe that the radius is valid for the M_j norm, $B_{M_j}([x_j], \delta_j) \subset \mathbb{R}^n$ for $j = 1, \dots, k$ is a conservative output, i.e. $B_{M_j}([x_j], \delta_j)$ is an over-approximation for the set of states reachable at time t_1 starting from any state (t_0, x) , such that $x \in \mathcal{X}$

$$\text{Reach}((t_0, \mathcal{X}), t_j) \subset B_{M_j}([x_j], \delta_j), \text{ for } j = 1, \dots, k.$$

We remark that for the intermediate time-bounds, i.e., for:

$$[\phi_{t_0}^t([X])], \text{ for } t \in (t_0, t_1), \quad (15)$$

the so-called *rough enclosures* can be used. These provide *coarse* bounds, as graphically illustrated in Figure 2 in the appendix.

6. Compute interval matrix bounds for the M_0/M_1 Cauchy-Green deformation tensors:

$$\left[\left((C_0^T)^{-1} \cdot [D_x \phi_{t_0}^{t_1}([X])]^T \cdot C_1^T \right) \cdot (C_1 \cdot [D_x \phi_{t_0}^{t_1}([X])] \cdot C_0^{-1}) \right], \quad (16)$$

where C_0^T, C_0 are s.t. $C_0^T C_0 = M_0$, and C_1^T, C_1 are s.t. $C_1^T C_1 = M_1$. The interval matrix operations are executed in the order given by the brackets.

7. Compute a value $\Lambda > 0$ (M_0/M_1 stretching factor) as an upper bound for the square-root of the maximal eigenvalue of each (symmetric) matrix in (16):

$$\Lambda \geq \sqrt{\lambda_{max}(C)},$$

for all $C \in \left[\left((C_0^T)^{-1} \cdot [D_x \phi_{t_0}^{t_1}([X])]^T \cdot C_1^T \right) \cdot (C_1 \cdot [D_x \phi_{t_0}^{t_1}([X])] \cdot C_0^{-1}) \right]$.

This quantity Λ can be used for the purpose of computation of validated bound for the *finite-time Lyapunov exponent*

$$FTLE(B_{M_0}([x_0], \delta_0)) = \frac{1}{t_1 - t_0} \ln(\Lambda_{prev} \cdot \Lambda),$$

where Λ_{prev} is the product of all stretching factors computed in previous steps.

8. Compute the new radius for the ball at time $t_1 = t_0 + h$:

$$\delta(t_1) = \Lambda \cdot \delta(t_0),$$

9. Set the new center of the ball at time t_1 as follows:

$$[x_1] = [\phi_{t_0}^{t_1}([x_0])].$$

10. Set the initial time to t_1 , the bounds for the initial center of the ball to $[x_1]$, the current norm to M_1 , the radius in M_1 -norm to $\delta_1 = \Lambda \cdot \delta_0$, and the ball enclosure for the set of initial states to $B_{M_1}([x_1], \Lambda \cdot \delta(t_0))$. If $t_1 \geq T$ terminate. Otherwise go back to 1.

End LRT

3.4 LRT-Algorithm Correctness Proof

In this section we provide a proof that the LRT, our new reachtube-computation algorithm, is an overapproximation of the behavior of the system described by Equations (1). This main result is captured by the following theorem.

Theorem 3 (LRT-Conservativity). *Assume that the rigorous tool used in the Lagrangian Reachtube Algorithm (LRT) produces conservative gradient enclosures for system (1) in the sense of Definition 2, and it guarantees the existence of the solutions within time intervals. Assume also that the LRT terminates on the provided inputs.*

Then, the output of the LRT is a conservative reachtube over-approximation of (1) at times $\{t_j\}_{j=0}^k$, that is:

$$\text{Reach}((t_0, \mathcal{X}), t_j) \subset B_{M_j}([x_j], \delta_j), \text{ for } j = 1, \dots, k,$$

bounded solutions exists for all intermediate times $t \in (t_j, t_{j+1})$.

Proof. Let $\mathcal{X} = B_{M_0}([x_0], \delta_0)$ be a ball enclosure for the set of initial states. Without losing generality we analyze the first step of the algorithm. The same argument applies to the consecutive steps (with the initial condition changed appropriately, as explained in the last step of the algorithm).

The representable enclosure $[X] \subset \mathbb{R}^n$ (product of intervals in canonical coordinates) computed in Step 4 satisfies $B_{M_0}([x_0], \delta_0) \subset [X]$. By the assumption the rigorous-tool used produces conservative enclosures for the gradient of the flow induced by ODEs (1). Hence, as the set $[X]$ containing $B_{M_0}([x_0], \delta_0)$ is the input to the rigorous forward-time integration procedure in Step 5, for the gradient enclosure $[D_x \phi_{t_0}^{t_1}([X])]$ computed in Step 5, it holds that

$$\{D_x \phi_{t_0}^{t_1}(x) \text{ for all } x \in \mathcal{X}\} \subset [D_x \phi_{t_0}^{t_1}([X])].$$

Therefore, the set $[D_x \phi_{t_0}^{t_1}([X])]$ can be interpreted as the set \mathcal{D} , i.e., the compact set containing all gradients of the solution at time t_1 with initial condition in $B_{M_0}([x_0], \delta_0)$ appearing in Theorem 1, see (11). As a consequence, the value Λ computed in Step 7 satisfies the following inequality:

$$\Lambda \geq \sqrt{\lambda_{\max}((C_1^T)^{-1} D^T M_0 D C_1^{-1})}, \text{ for all } D \in \mathcal{D}.$$

From Theorem 2 it follows that:

$$\phi_{t_0}^{t_1}(x) \in B_{M_1}(\phi_{t_0}^{t_1}([x_0]), \Lambda \cdot \delta(t_0)) \text{ for all } x \in \mathcal{X},$$

which implies that:

$$\text{Reach}((t_0, \mathcal{X}), t_1) \subset B_{M_1}([\phi_{t_0}^{t_1}([x_0])], \Lambda \cdot \delta(t_0))$$

which proves our overapproximation (conservativity) claim. Existence of the solutions for all times $t \in (t_0, t_1)$ is guaranteed by the assumption about the rigorous tool (in the step 5 of the LRT algorithm we use rough enclosures, see Figure 2). \square

3.5 Wrapping Effect in the Algorithm

A very important precision-loss issue of conservative approximations, and therefore of validated methods for ODEs, is the *wrapping effect*. This occurs when

a set of states is wrapped (conservatively enclosed) within a box defined in a particular norm. The weighted M -norms technique the LRT uses (instead of the standard Euclidean norm) is a way of reducing the effect of wrapping. More precisely, in Step 7 of every iteration, the LRT finds an appropriate norm which minimizes the stretching factor computed from the set of Cauchy-Green deformation tensors.

However, there are other sources of the wrapping effect in the algorithm. The discrete reachtube bounds are in a form of a ball in appropriate metric space, which is an ellipsoidal set in canonical coordinates, see example on Fig. 3a. In Step 4 of the LRT algorithm, a representable enclosure in canonical coordinates for the ellipsoidal reachtube over-approximation is computed. When the ellipsoidal set is being directly wrapped into a box in canonical coordinates (how it is done in the algorithm currently), the wrapping effect is considerably larger than when the ellipsoidal set is wrapped into a rectangular set reflecting the eigen-coordinates. We illustrate the wrapping effect using the following weighted norm (taken from one of our experiments):

$$M = \begin{bmatrix} 7 & -9.5 \\ -9.5 & 19 \end{bmatrix} \quad (17)$$

Figure 3 shows the computation of enclosures for a ball represented in the weighted norm given by M . It is clear from Figure 1(c) that the box enclosure of the ball in the eigen-coordinates (blue rectangle) is much tighter than the box enclosure of it in the canonical coordinates (green square).

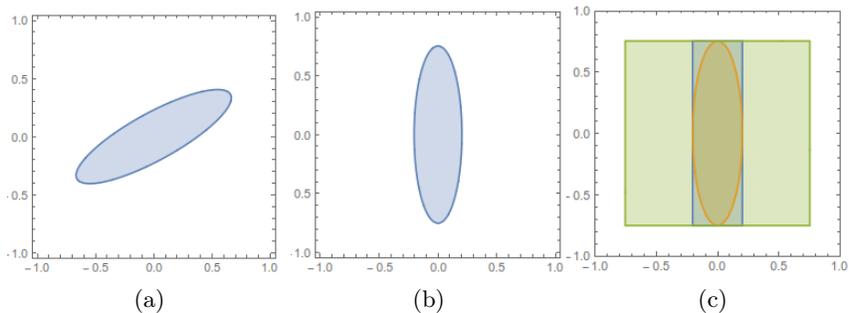


Fig. 3: (a) A ball in the weighted norm given by M of radius 1 (the ellipsoidal set). (b) The ellipsoidal set in its eigen-coordinates (unrotated). (c) Wrapping the ellipsoidal set in a box: blue rectangle in eigen-coordinates and green square in canonical-coordinates.

Step 6 of the LRT is another place where reducing the wrapping effect has the potential to considerably increase the precision of the LRT. This step computes the product of interval matrices, which results in large overapproximations for wide-intervals matrices. In fact, in the experiments considered in Section 4, if the initial-ball radius is large, we observed that the overestimate of the stretching fac-

tor tends to worsen the LRT performance in reachtube construction, when compared to a direct application of CAPD and Flow*. We plan to find workarounds for this problem. One possible solution would be to use matrix decomposition, and compute the eigenvalues of the matrix by using this decomposition.

3.6 Direct computation of the optimal M -norm

The computation of the optimal M -norm enables the estimation of the stretching factor. Step 7 of the LRT finds norm M_1 and decomposes it as $M_1 = C_1^T C_1$, such that, for a gradient matrix \widetilde{D} the following inequality holds for all positive-definite symmetric matrices \widetilde{M} :

$$\sqrt{\lambda_{max}((C_1^T)^{-1} D^T M_1 D C_1^{-1})} \leq \sqrt{\lambda_{max}((\widetilde{C}^T)^{-1} D^T \widetilde{M} D \widetilde{C}^{-1})}, \quad (18)$$

Below we illustrate how to compute M_1 for 2D systems. This can be generalized to higher-dimensional systems.

I) D has complex conjugate eigenvalues $\lambda = \alpha \pm i\beta$. In this case $w \pm iv$ is the associated pair of complex conjugate eigenvectors, where $w, v \in \mathbb{R}^2$. Define:

$$C = [w \ v]^{-1}$$

As a consequence we have the following equations:

$$C D C^{-1} = \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix}, \text{ and } (C^T)^{-1} D^T C^T = \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix},$$

Thus, one obtains the following results:

$$(C^T)^{-1} D^T M A C^{-1} = ((C^T)^{-1} D^T C^T)(C D C^{-1}) = \begin{bmatrix} \alpha^2 + \beta^2 & 0 \\ 0 & \alpha^2 + \beta^2 \end{bmatrix},$$

Clearly, one has that:

$$\lambda_{max}((C^T)^{-1} D^T M D C^{-1}) = \alpha^2 + \beta^2.$$

As the eigenvalues of $(C^T)^{-1} D^T M D C^{-1}$ are equal, it follows from the identity of the determinants that the inequality below holds for any $\widetilde{M} = \widetilde{C}^T \widetilde{C}$:

$$\lambda_{max}((C^T)^{-1} D^T M D C^{-1}) \leq \lambda_{max}((\widetilde{C}^T)^{-1} D^T \widetilde{M} D \widetilde{C}^{-1})$$

II) D has two distinct real eigenvalues $\lambda_1 \neq \lambda_2$ In this case we do not find a positive-definite symmetric matrix. However, we can find a rotation matrix, defining coordinates in which the stretching factor is smaller than in canonical coordinates (M -norm). Let $B \in \mathbb{R}^{2 \times 2}$ be the eigenvectors matrix of D . Denote:

$$B^{-1} D B = \widetilde{D} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}.$$

Let R be the rotation matrix

$$R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}, \quad c, s \neq 0, \quad \text{hence, } R^{-1}\tilde{D}R = \begin{bmatrix} \frac{\lambda_1 c^2 + \lambda_2 s^2}{c^2 + s^2} & \frac{(\lambda_1 - \lambda_2)cs}{c^2 + s^2} \\ -\frac{(\lambda_1 - \lambda_2)cs}{c^2 + s^2} & \frac{\lambda_1 c^2 + \lambda_2 s^2}{c^2 + s^2} \end{bmatrix}.$$

$$\text{For } s, c = 1 \text{ we have } R^T \tilde{D}^T (R^T)^{-1} R^{-1} \tilde{D} R = \begin{bmatrix} \left(\frac{\lambda_1 + \lambda_2}{2}\right)^2 & 0 \\ 0 & \left(\frac{\lambda_1 + \lambda_2}{2}\right)^2 \end{bmatrix}.$$

Therefore, we may set $C = (BR)^{-1}$, which for $\lambda_1 < \lambda_2$ results in $\lambda_{max}(C^T D^T (C^T)^{-1} C^{-1} D C) < \lambda_{max}(D^T D)$, because $\left(\frac{\lambda_1 + \lambda_2}{2}\right)^2 < \lambda_2^2$.

III) $D \in \mathbb{R}^{n \times n}$, where $n > 2$. In this case we call the Matlab engine part of our code. Precisely, we use the external linear-optimization packages [24,16]. We initially set $\gamma = (\lambda_1 \lambda_2 \cdots \lambda_n)^{1/n}$. Then, using the optimization package, we try to find M_1 and its decomposition, such that:

$$\sqrt{\lambda_{max}((C_1^T)^{-1} D^T M_1 D C_1^{-1})} \leq \gamma, \quad (19)$$

If we are not successful, we increase γ until an M_1 satisfying (19) is found.

4 Implementation and Experimental Evaluation

Prototype Implementation Our implementation is based on interval arithmetic, i.e. all variables used in the algorithm are over intervals, and all computations performed are executed using interval arithmetic. The main procedure is implemented in C++, which includes header files for the CAPD tool (implemented in C++ as well) to compute rigorous enclosures for the center of the ball at time t_1 in step 1, and for the gradient of the flow at time t_1 in step 5 of the LRT algorithm (see Section 3.3).

To compute the optimal norm and its decomposition for dimensions higher than 2 in step 2 of the LRT algorithm, we solve a semidefinite optimization problem. We found it convenient to use dedicated Matlab packages for that purpose [24,16], in particular for Case 3 in Section 3.6.

To compute an upper bound A for the square-root of the maximal eigenvalue of all symmetric matrices in some interval bounds, we used the VERSOFT package [27,28] implemented in Intlab [29]. To combine C++ and the Matlab/Intlab part of the implementation, we use an engine that allows one to call Matlab code within C++ using a special makefile. The source code, numerical data, and readme file describing compilation procedure for LRT can be found online [14].

We remark that the current implementation is a proof of concept; in particular, it is not optimized in terms of the runtime—a direct CAPD implementation is an order of magnitude faster. We will investigate ways of significantly improving the runtime of the implementation in future work.

BM	dt	T	ID	LRT		Flow*		(direct) CAPD		nLRT	
				(F/I)V	(A/I)V	(F/I)V	(A/I)V	(F/I)V	(A/I)V	(F/I)V	(A/I)V
B(2)	0.01	20	0.02	7.7e-5	0.09	7.9e-5	0.15	Fail	Fail	Fail	Fail
I(2)	0.01	20	0.02	4.3e-9	0.09	5e-9	0.12	7.4e-9	0.10	1.45	1.31
L(3)	0.001	2	1.4e-6	1.6e5	9.0e3	2.1e13	7.1e12	4.5e4	1.7e3	1.2e22	4.9e19
F(2)	0.01	40	2e-3	1.2e-42	1.01	NA	NA	6.86e-4	0.18	5.64e7	3.21e5
M(2)	0.001	4	2e-3	0.006	0.22	0.29	0.54	0.29	0.52	4.01	2.24
R(4)	0.01	20	1e-2	2.2e-19	0.07	2.4e-15	0.31	2.9e-11	0.23	Fail	Fail
O(7)	0.01	4	2e-4	71.08	34.35	272	5.1e4	4.3e3	620	Fail	Fail
P(12))	5e-4	0.1	0.01	5.25	4.76	290	64.6	280	62.4	19.4	6.2

Table 1: Performance comparison with Flow* and CAPD. We use the following abbreviations: B(2)-Brusselator, I(2)-Inverse Van der Pol oscillator, L(3)-Lorenz attractor, F(2)-Forced Van der Pol oscillator, M(2)-Mitchell Schaeffer cardiac cell model, R(4)-Robot arm, O(7)-Biology model, A(12)-Polynomial system (number inside parenthesis denotes dimension). T: time horizon, dt: time step, ID: initial diameter in each dimension, (F/I)V: ratio of final and initial volume, (A/I)V: ratio of average and initial volume. NA: Not applicable, Fail: Volume blow-up.

Experimental Evaluation We compare the results obtained by LRT with direct CAPD and Flow* on a set of standard benchmarks [6,10]: the Brusselator, inverse-time Van der Pol oscillator, the Lorenz equations [18], a robot arm model [2], a 7-dimensional biological model [10], and a 12-dimensional polynomial [1] system. Additionally, we consider the forced Van der Pol oscillator [30] (a time-variant system), and the Mitchell Schaeffer [23] cardiac cell model.

Our results are given in Table 1, and were obtained on a Ubuntu 14.04 LTS machine, with an Intel Core i7-4770 CPU 3.40GHz x 8 processor and 16 GB memory. The results presented in the columns labeled *(direct) CAPD* and *Flow** were obtained using the CAPD software package and Flow*, respectively. The internal parameters used in the codes can be checked online [14]. The comparison metric that we chose is ratio of the final and initial volume, and ratio of the average and initial volume. We compute directly volumes of the reachtube over-approximations in form of rectangular sets obtained using CAPD, and Flow* software. Whereas, the volumes of the reachtube over-approximations obtained using the LRT algorithm are approximated by the volumes of the tightest rectangular enclosure of the ellipsoidal set, as illustrated on Fig. 3c. The results in *nLRT* column were obtained using a naive implementation of the LRT algorithm, in which the metric space is chosen to be globally Euclidean, i.e., $M_0 = M_1 = \dots = M_k = Id$. For MS model the initial condition (i.c.) is in stable regime. For Lorenz i.c. is a period 2 unstable periodic orbit. For all the other benchmark equations the initial condition was chosen as in [10].

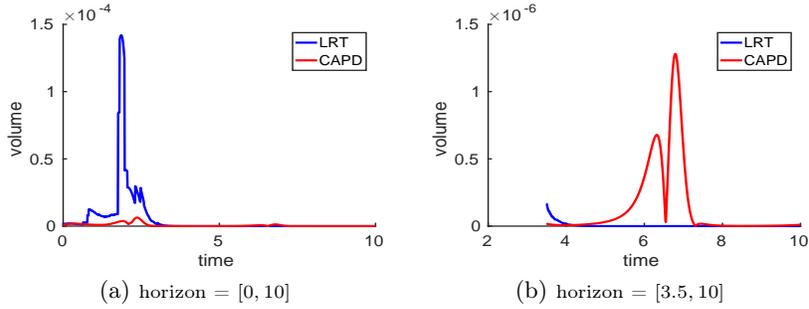


Fig. 4: Volume comparison for forced Van der Pol Oscillator

Some interesting observations about our experiments are as follows. Fig. 4 illustrates that the volumes of the LRT reachtube of the forced Van der Pol oscillator increase significantly for some initial time-steps compared to CAPD reachtube (nevertheless reduce in the long run). This initial increase is related to the computation of new coordinates, which are significantly different from the previous coordinates, resulting in a large stretching factor in step 7 of the LRT algorithm. Namely, we observed that it happens when the coordinates are switched from Case II to Case I, as presented in Section 3.6. We, however, observe that this does not happen in the system like MS cardiac model (see Fig. 5(a)). We believe those large increases of stretching factors in systems like fVDP can be avoided by smarter choices of the norms. We will further investigate those possibilities in future work.

Table 1 shows that LRT does not perform well for Lorentz attractor compared to CAPD (see also Fig. 5(b)), as the CAPD tool is current state of the art for such chaotic systems. LRT, however, performs much better than Flow* for this example. In all other examples, the LRT algorithm behaves favorably (outputs tighter reachtube, compare (F/I)V) in the long run.

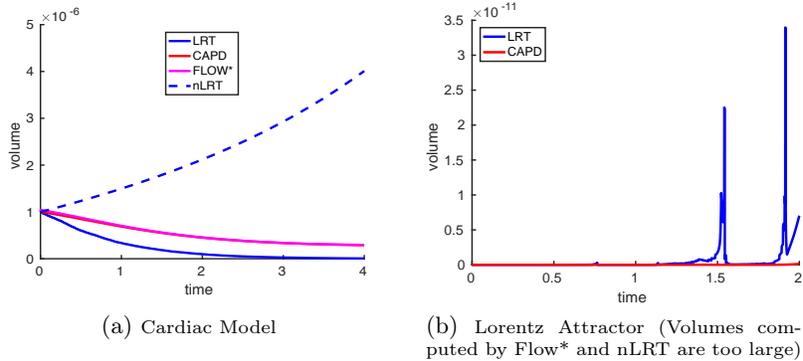


Fig. 5: Volume comparison in two important benchmarks

5 Conclusions

We presented LRT, a rigorous procedure for computing ReachTube overapproximations based on Cauchy-Green stretching factor computation, in Lagrangian

coordinates. We plan to pursue further research on our algorithm. One appealing possibility is to extend LRT to hybrid systems widely used in research on cardiac dynamics, and many other fields of study.

We also plan to implement LRT with forward-in-time integration and conservative-enclosures computation of the gradient, by just using a simple independent code (instead of CAPD). By such a code we mean a rigorous integration procedure based, for example, on the Taylor's method. This code would directly compute an interval enclosure for the Cauchy-Green deformation tensors in an appropriate metric space. It would then be interesting to compare the wrapping-reduction performance of such a code with the procedure described above.

Acknowledgments Research supported in part by the following grants: NSF IIS-1447549, NSF CPS-1446832, NSF CPS-1446725, NSF CNS-1445770, NSF CNS-1430010, AFOSR FA9550-14-1-0261 and ONR N00014-13-1-0090.

References

1. J. Anderson and A. Papachristodoulou. A decomposition technique for nonlinear dynamical system analysis. *IEEE Transactions on Automatic Control*, 57(6):1516–1521, 2012.
2. D. Angeli, E. D. Sontag, and Y. Wang. A characterization of integral input-to-state stability. *IEEE Transactions on Automatic Control*, 45(6):1082–1097, 2000.
3. M. Berz and K. Makino. Verified integration of odes and flows using differential algebraic methods on high-order taylor models. *Reliable Computing*, 4(4):361–369, 1998.
4. M. Capiński, J. Cyranka, Z. Galias, T. Kapela, M. Mrozek, P. Pilarczyk, D. Wilczak, P. Zgliczyński, and M. Żelawski. CAPD - computer assisted proofs in dynamics, a package for rigorous numerics, <http://capd.ii.edu.pl/>. Technical report, Jagiellonian University, Kraków, 2016.
5. X. Chen, E. Abraham, and S. Sankaranarayanan. Taylor model flowpipe construction for non-linear hybrid systems. In *Proceedings of the 2012 IEEE 33rd Real-Time Systems Symposium, RTSS '12*, pages 183–192, Washington, DC, USA, 2012. IEEE Computer Society.
6. X. Chen, E. Ábrahám, and S. Sankaranarayanan. *Flow*: An Analyzer for Non-linear Hybrid Systems*, pages 258–263. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
7. G. Dahlquist. *Stability and Error Bounds in the Numerical Integration of Ordinary Differential Equations*. Transactions of the Royal Institute of Technology. Almqvist & Wiksells, Uppsala, 1958.
8. A. Donzé. *Breach, A Toolbox for Verification and Parameter Synthesis of Hybrid Systems*, pages 167–170. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
9. A. Donzé and O. Maler. *Systematic Simulation Using Sensitivity Analysis*, pages 174–189. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
10. C. Fan, J. Kapinski, X. Jin, and S. Mitra. Locally optimal reach set over-approximation for nonlinear systems. In *Proceedings of the 13th International Conference on Embedded Software, EMSOFT '16*, pages 6:1–6:10, New York, NY, USA, 2016. ACM.
11. C. Fan and S. Mitra. *Bounded Verification with On-the-Fly Discrepancy Computation*, pages 446–463. Springer International Publishing, Cham, 2015.

12. Z. Galias and P. Zgliczyński. Computer assisted proof of chaos in the Lorenz equations. *Physica D: Nonlinear Phenomena*, 115(3):165 – 188, 1998.
13. K. Hashiguchi and Y. Yamakawa. *Introduction to Finite Strain Theory for Continuum Elasto-Plasticity*. Wiley Series in Computational Mechanics. Wiley, 2012.
14. M. A. Islam and J. Cyranka. LRT prototype implementation. http://www.cs.cmu.edu/~mdarifui/cav_codes.html, 2017.
15. T. Kapela and P. Zgliczyński. The existence of simple choreographies for the N-body problem—a computer-assisted proof. *Nonlinearity*, 16(6):1899, 2003.
16. J. Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE, 2005.
17. W. Lohmiller and J.-J. E. Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, June 1998.
18. E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.
19. S. M. Lozinskii. Error estimates for the numerical integration of ordinary differential equations, part i. *Izv. Vyss. Uceb. Zaved. Matematika*, 6:52–90, 1958.
20. J. Maidens and M. Arca. Reachability analysis of nonlinear systems using matrix measures. *IEEE Transactions on Automatic Control*, 60(1):265–270, Jan 2015.
21. K. Makino and M. Berz. Cosy infinity version 9. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 558(1):346–350, 2006.
22. K. Makino and M. Berz. Rigorous integration of flows and odes using taylor models. *Symbolic Numeric Computation*, pages 79–84, 2009.
23. C. C. Mitchell and D. G. Schaeffer. A two-current model for the dynamics of cardiac membrane. *Bulletin of mathematical biology*, 65(5):767–793, 2003.
24. A. MOSEK. The mosek optimization tools version 3.2 (revision 8) user’s manual and reference, 2002.
25. N. S. Nedialkov. Interval tools for odes and daes. In *12th GAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics (SCAN 2006)*, pages 4–4, Sept 2006.
26. N. S. Nedialkov. Vnode-lp—a validated solver for initial value problems in ordinary differential equations. In *Technical Report CAS-06-06-NN*. 2006.
27. J. Rohn. Bounds on eigenvalues of interval matrices. *ZAMM-Zeitschrift fur Angewandte Mathematik und Mechanik*, 78(3):S1049, 1998.
28. J. Rohn. Versoft: Guide. Technical report, Technical Report, 2011.
29. S. Rump. *Developments in Reliable Computing*, chapter INTLAB - INTerval LABoratory, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999.
30. B. Van Der Pol. Vii. forced oscillations in a circuit with non-linear resistance.(reception with reactive triode). *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 3(13):65–80, 1927.
31. D. Wilczak and P. Zgliczyński. Heteroclinic connections between periodic orbits in planar restricted circular three body problem. Part ii. *Communications in Mathematical Physics*, 259(3):561–576, 2005.
32. D. Wilczak and P. Zgliczyński. C^r -Lohner algorithm. *Schedae Informaticae*, 2011(Volume 20), 2012.
33. P. Zgliczynski. C^1 Lohner Algorithm. *Foundations of Computational Mathematics*, 2(4):429–465, 2002.

A Proofs of the Lemmas in Section 2

Lemma 3. Consider the Cauchy problem (1). Let $x_0, y_0 \in \mathbb{R}^n$ be two initial conditions at time t_0 . Let $M \in \mathbb{R}^{n \times n}$ be a positive-definite symmetric matrix and $C^T C = M$ be its decomposition. For $t_1 \geq t_0$, it holds that

$$\|\phi_{t_0}^{t_1}(x_0) - \phi_{t_0}^{t_1}(y_0)\|_M \leq \sqrt{\hat{\lambda} \left((C^T)^{-1} (D_x \phi_{t_0}^{t_1}(\xi))^T M D_x \phi_{t_0}^{t_1}(\xi) C^{-1} \right)} \|x_0 - y_0\|_M$$

where $\xi = \omega x_0 + (1 - \omega)y_0$ for some $\omega \in [0, 1]$. For the particular case of the Euclidean norm, (8) takes the form

$$\|\phi_{t_0}^{t_1}(x_0) - \phi_{t_0}^{t_1}(y_0)\|_2 \leq \sqrt{\lambda_{max} \left((D_x \phi_{t_0}^{t_1}(\xi))^T \cdot D_x \phi_{t_0}^{t_1}(\xi) \right)} \|x_0 - y_0\|_2.$$

Proof. Let $\xi(\omega) = \omega x_0 + (1 - \omega)y_0$. From

$$\int_0^1 D_x \phi_{t_0}^{t_1}(\xi(\omega)) d\omega = \frac{1}{x_0 - y_0} (\phi_{t_0}^{t_1}(x_0) - \phi_{t_0}^{t_1}(y_0)),$$

and the well known mean value theorem for integrals, it holds that

$$\phi_{t_0}^{t_1}(x_0) - \phi_{t_0}^{t_1}(y_0) = D_x \phi_{t_0}^{t_1}(\hat{\xi})(x_0 - y_0)$$

for some $\hat{\omega} \in [0, 1]$, $\xi(\hat{\omega}) = \hat{\xi}$. From taking norms in above equation we obtain

$$\|\phi_{t_0}^{t_1}(x_0) - \phi_{t_0}^{t_1}(y_0)\| = \|D_x \phi_{t_0}^{t_1}(\hat{\xi})(x_0 - y_0)\| \leq \|D_x \phi_{t_0}^{t_1}(\hat{\xi})\| \|x_0 - y_0\|.$$

Replacing $\|D_x \phi_{t_0}^{t_1}(\hat{\xi})\|_2$ with the inducted Euclidean matrix norm we obtain (9). If we use the weighted M -norm (6) we have for the matrix norm (7)

$$\|\phi_{t_0}^{t_1}(x_0) - \phi_{t_0}^{t_1}(y_0)\|_M \leq \sqrt{\lambda_{max} \left((C^T)^{-1} (D_x \phi_{t_0}^{t_1}(\hat{\xi}))^T M D_x \phi_{t_0}^{t_1}(\hat{\xi}) C^{-1} \right)} \|x_0 - y_0\|_M$$

□

Lemma 4. Consider the Cauchy problem (1). Let $x_0, y_0 \in \mathbb{R}^n$ be two initial conditions at time t_0 . Let $M_0, M_1 \in \mathbb{R}^{n \times n}$ be positive-definite symmetric matrices, and $C_0^T C_0 = M_0$, $C_1^T C_1 = M_1$ their decompositions respectively. For $t_1 \geq t_0$, it holds that

$$\begin{aligned} & \|\phi_{t_0}^{t_1}(x_0) - \phi_{t_0}^{t_1}(y_0)\|_{M_1} \\ & \leq \sqrt{\lambda_{max} \left((C_0^T)^{-1} \cdot (D_x \phi_{t_0}^{t_1}(\xi))^T \cdot M_1 \cdot D_x \phi_{t_0}^{t_1}(\xi) \cdot C_0^{-1} \right)} \|x_0 - y_0\|_{M_0}, \end{aligned}$$

where $\xi = \omega x_0 + (1 - \omega)y_0$ for some $\omega \in [0, 1]$.

Proof. Let $\xi = \omega x_0 + (1 - \omega)y_0$ for some $\omega \in [0, 1]$. We use the equality $\phi_{t_0}^{t_1}(x_0) - \phi_{t_0}^{t_1}(y_0) = D_x \phi_{t_0}^{t_1}(\xi)(x_0 - y_0)$ derived in the proof of Lemma 1. Let us denote $A := D_x \phi_{t_0}^{t_1}(\xi)$, and $w = (x_0 - y_0)$. It holds that

$$\begin{aligned} \|\phi_{t_0}^{t_1}(x_0) - \phi_{t_0}^{t_1}(y_0)\|_{M_1} &= \|Aw\|_{M_1} = \sqrt{(Aw)^T M_1 (Aw)} = \sqrt{w^T (A^T M_1 A) w} = \\ &= \sqrt{w^T C_0^T ((C_0^T)^{-1} A^T M_1 A C_0^{-1}) C_0 w} \leq \sqrt{\lambda_{max} ((C_0^T)^{-1} A^T M_1 A C_0^{-1})} \sqrt{w^T C_0^T C_0 w} \\ &= \sqrt{\lambda_{max} ((C_0^T)^{-1} A^T M_1 A C_0^{-1})} \|w\|_{M_0} \quad \square \end{aligned}$$