Contraction, propagation and bisection on a validated simulation of ODE

Julien Alexandre dit Sandretto
Alexandre Chapoutot

Department U2IS
ENSTA ParisTech

SWIM 2016 - Lyon

Contents

Recall on validated simulation

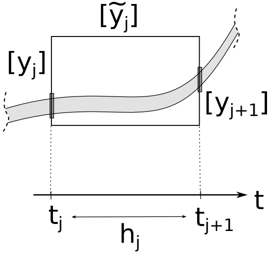
Initial value problem

$$\dot{y} = f(y, p, t) \text{ with } y(0) \in [y_0] \text{ and } p \in [p] \ .$$

Classical approach

Lohner 2-steps method:

1. Find $[\tilde{y}_i]$ and h_i with Picard-Lindelöf operator and Banach's theorem
2. Compute $[y_{i+1}]$ with a validated integration scheme: Taylor (Vnode-LP, CAPD) or Runge-Kutta (DynIbex)

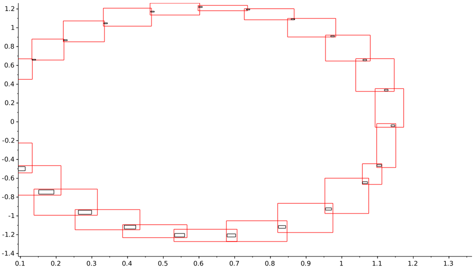


Recall on validated simulation

Validated simulation ⇒ Two lists of boxes

$\mathcal{S} = \{[y_0], \ldots, [y_i], \ldots, [y_N]\}$ and $\tilde{\mathcal{S}} = \{[\tilde{y_0}], \ldots, [\tilde{y_i}], \ldots, [\tilde{y_N}]\}$,
such that, for all p and y_0,

- $y(t_i) \in [y_i]$ with $t_i \in \{0, t_1, \ldots, t_N\}$
- $y(t) \in [\tilde{y}_i]$ for all $t \in [t_i, t_{i+1}]$



red: $\tilde{\mathcal{S}}$
black: $\mathcal{S}$

Contractors

An interval contractor (Ctc)

- $Ctc([x]) \subset [x]$ (contractance property)
- $Ctc([x]) \cap \mathcal{X} = [x] \cap \mathcal{X}$ (completeness property)

A contractor associated to a constraint is a contractor associated to the set $\mathcal{X}$ of all x which satisfy the constraint.

Contractors in simulation

- Picard operator is a contractor
  $Ctc_{picard}([\tilde{y}_j]) := [\tilde{y}_j] \cap [y_j] + \int_0^h f([\tilde{y}_j])ds$
- Validated integration scheme is a contractor
  $Ctc_{integration}([y_{j+1}]) := [y_{j+1}] \cap [y_j] + \int_0^h f([y_j])ds$

($\int$ enclosed by validated Taylor or RK)

Contractors based on tricks

Monotonicity contractor

If $0 \notin f([\tilde{y}_j])$, monotonicity allows $[\tilde{y}_j] = [y_j] \cup [y_{j+1}]$ (and recomputation of $[y_{j+1}]$ with new remainder and fixed point...)

Slicing contractor

If $[y_{j+1}] \supset [y_j]$ (possible attractor in $[y_j]$), slicing of $[y_j]$, integration of slices, and $[y_{j+1}] =$ union of results...

Constraint Satisfaction Differential Problems

Problem

If information is given on $y(t)$ $\Rightarrow$ how take it into account ?

Information on $y(t)$

Given under the form of a temporal set constraints: $y(t^*) \in \mathcal{A}$
Note: $\mathcal{A}$ can be the result of a contraction of $y(t^*)$

w.r.t. a constraint (e.g. $y(t^*)^2 - 3\cos(y(t^*)) < 0$ or $y(t^*) \notin \mathcal{B}$)

For us, a CSDP is a set of temporal set constraints:

$$\{y(t_1) \in \mathcal{A}_1, \ldots, y(t_z) \in \mathcal{A}_z\}$$

Our Approach

Contraction and propagation of already computed simulation:

avoids costly Lohner first step

Constraint Satisfaction Differential Problems

Problem

If information is given on $y(t) \Rightarrow$ how take it into account ?

Information on $y(t)$

Given under the form of a temporal set constraints: $y(t^*) \in \mathcal{A}$
Note: $\mathcal{A}$ can be the result of a contraction of $y(t^*)$

w.r.t. a constraint (e.g. $y(t^*)^2 - 3\cos(y(t^*)) < 0$ or $y(t^*) \notin \mathcal{B}$)

For us, a CSDP is a set of temporal set constraints:

$$\{y(t_1) \in \mathcal{A}_1, \ldots, y(t_z) \in \mathcal{A}_z\}$$

Our Approach

Contraction and propagation of already computed simulation:

avoids costly Lohner first step

Constraint Satisfaction Differential Problems

Problem

If information is given on $y(t)$ $\Rightarrow$ how take it into account ?

Information on $y(t)$

Given under the form of a temporal set constraints: $y(t^*) \in \mathcal{A}$
Note: $\mathcal{A}$ can be the result of a contraction of $y(t^*)$

w.r.t. a constraint (e.g. $y(t^*)^2 - 3\cos(y(t^*)) < 0$ or $y(t^*) \notin \mathcal{B}$)
For us, a CSDP is a set of temporal set constraints:

$$\{y(t_1) \in \mathcal{A}_1, \dots, y(t_z) \in \mathcal{A}_z\}$$

Our Approach

Contraction and propagation of already computed simulation:

avoids costly Lohner first step

How take a temporal set constraint into account ?

$$\boxed{y(t^*) \in \mathcal{A}}$$

1-Add a new integration step

Simulation becomes: $\mathcal{S} = \{[y_0], \ldots, [y_i], [y_k], [y_{i+1}], \ldots, [y_N]\}$
s.t. $t_k = t^*$ and $y(t_k) \in [y_k]$, and same for $\tilde{\mathcal{S}}$
$\Rightarrow$ without Lohner first step because $y(t_k) \in [\tilde{y}_i]$ and $h = t_k - t_i$

2-Contraction

Apply the basic contractor $[y_k] := [y_k] \cap \mathcal{A}$

How take a temporal set constraint into account ?

$$\boxed{y(t^*) \in \mathcal{A}}$$

1-Add a new integration step

Simulation becomes: $\mathcal{S} = \{[y_0], \dots, [y_i], [y_k], [y_{i+1}], \dots, [y_N]\}$
s.t. $t_k = t^*$ and $y(t_k) \in [y_k]$, and same for $\tilde{\mathcal{S}}$
$\Rightarrow$ without Lohner first step because $y(t_k) \in [\tilde{y}_i]$ and $h = t_k - t_i$

2-Contraction

Apply the basic contractor $[y_k] := [y_k] \cap \mathcal{A}$

How take a temporal set constraint into account ?

$$\boxed{y(t^*) \in \mathcal{A}}$$

1-Add a new integration step

Simulation becomes: $\mathcal{S} = \{[y_0], \dots, [y_i], [y_k], [y_{i+1}], \dots, [y_N]\}$
s.t. $t_k = t^*$ and $y(t_k) \in [y_k]$, and same for $\tilde{\mathcal{S}}$
$\Rightarrow$ without Lohner first step because $y(t_k) \in [\tilde{y}_i]$ and $h = t_k - t_i$

2-Contraction

Apply the basic contractor $[y_k] := [y_k] \cap \mathcal{A}$

Remark on bisection

Of boxes

Bisection of $[A]$:
$[A_{left}] := Ctc_{left}([A])$ and $[A_{right}] := Ctc_{right}([A])$

Of states

Bisection of $y(t)$ at $t = t^*$:
$\mathcal{S}_{left} = \mathcal{S}$ and $\mathcal{S}_{right} = \mathcal{S}$, then $\mathcal{S}_{left} : y(t^*) \in Ctc_{left}([y_k])$ and $\mathcal{S}_{right} : y(t^*) \in Ctc_{right}([y_k])$

Remark on bisection

Of boxes

Bisection of $[A]$:
$[A_{left}] := Ctc_{left}([A])$ and $[A_{right}] := Ctc_{right}([A])$

Of states

Bisection of $y(t)$ at $t = t^*$:
$\mathcal{S}_{left} = \mathcal{S}$ and $\mathcal{S}_{right} = \mathcal{S}$, then $\mathcal{S}_{left} : y(t^*) \in Ctc_{left}([y_k])$ and $\mathcal{S}_{right} : y(t^*) \in Ctc_{right}([y_k])$

Propagation

Important to propagate "$y(t^*) \in \mathcal{A}$" to $y(t)$, $\forall t$

Forward propagation

From t^* to t_{end} :

$Ctc_{picard} + Ctc_{integration}$

Remark: Picard first to reduce Lagrange remainder of integration

Backward propagation

From t^* to t_0 :

$Ctc_{picard} + Ctc_{integration}$

with inverse function -f and past stepsize $h = t_i - t_{i-1}$

Fixed point

While sufficient improvement w.r.t. a given threshold:

From t_0 to t_{end} :

Forward propagation

From t_{end} to t_0 :

Backward propagation

Propagation

Important to propagate "$y(t^*) \in \mathcal{A}$" to $y(t)$, $\forall t$

Forward propagation

From t^* to t_{end} :

$Ctc_{picard} + Ctc_{integration}$

Remark: Picard first to reduce Lagrange remainder of integration

Backward propagation

From t^* to t_0 :

$Ctc_{picard} + Ctc_{integration}$

with inverse function -f and past stepsize $h = t_i - t_{i-1}$

Fixed point

While sufficient improvement w.r.t. a given threshold:

From t_0 to t_{end} :

Forward propagation

From t_{end} to t_0 :

Backward propagation

Propagation

Important to propagate "$y(t^*) \in \mathcal{A}$" to $y(t)$, $\forall t$

Forward propagation

From t^* to t_{end} :

$Ctc_{picard} + Ctc_{integration}$

Remark: Picard first to reduce Lagrange remainder of integration

Backward propagation

From t^* to t_0 :

$Ctc_{picard} + Ctc_{integration}$

with inverse function -f and past stepsize $h = t_i - t_{i-1}$

Fixed point

While sufficient improvement w.r.t. a given threshold:

From t_0 to t_{end} :

Forward propagation

From t_{end} to t_0 :

Backward propagation

Propagation

Important to propagate "$y(t^*) \in \mathcal{A}$" to $y(t)$, $\forall t$

Forward propagation

From t^* to t_{end} :

$Ctc_{picard} + Ctc_{integration}$

Remark: Picard first to reduce Lagrange remainder of integration

Backward propagation

From t^* to t_0 :

$Ctc_{picard} + Ctc_{integration}$

with inverse function -f and past stepsize $h = t_i - t_{i-1}$

Fixed point

While sufficient improvement w.r.t. a given threshold:

From t_0 to t_{end} :

Forward propagation

From t_{end} to t_0 :

Backward propagation

Example

Van Der Pol

$$\begin{cases} x' = y \\ y' = 2.0(1.0 - x^2)y - x \end{cases}$$

$x(0) \in [2.0, 2.2]$ and $y(0) \in [0.0, 0.1]$

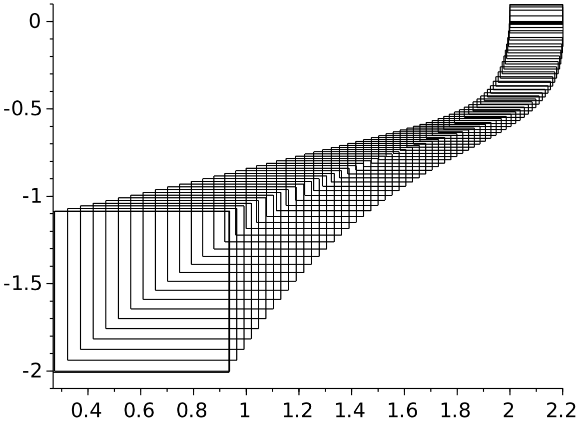
$t_{end} = 2.0$

Bisection then propagation

$x(0) = \{x(0)_{left}, x(0)_{right}\}$

1 simulation then 2 propagations vs 3 simulations

Time : 2+2*1 = 4 sec. vs 2*3 = 6 sec.

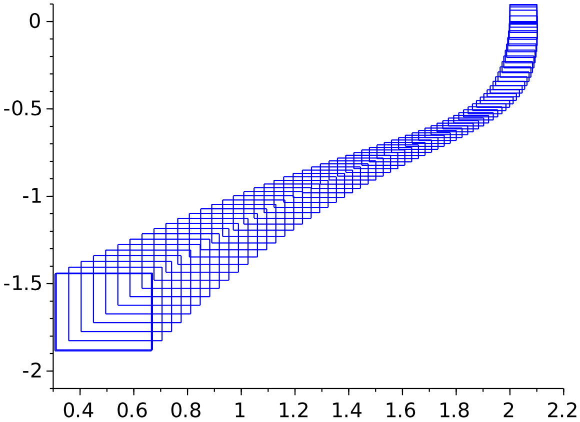


Bisection then propagation

$x(0) = \{x(0)_{left}, x(0)_{right}\}$

1 simulation then 2 propagations vs 3 simulations

Time : 2+2*1 = 4 sec. vs 2*3 = 6 sec.

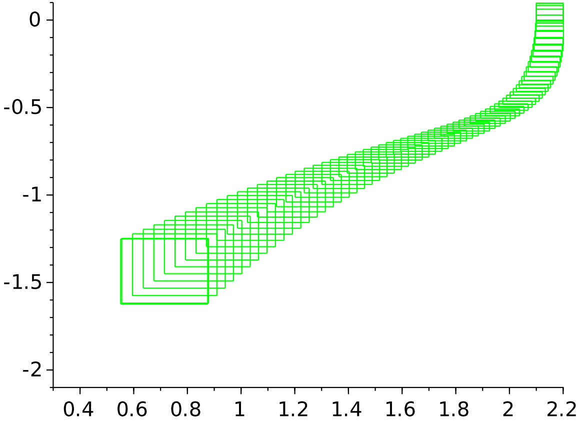


Bisection then propagation

$x(0) = \{x(0)_{left}, x(0)_{right}\}$

1 simulation then 2 propagations vs 3 simulations

Time : 2+2*1 = 4 sec. vs 2*3 = 6 sec.

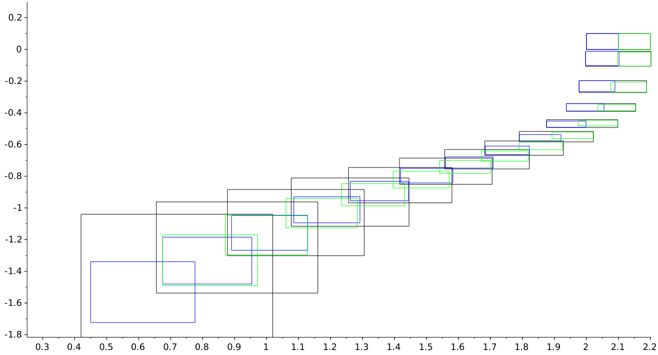


Bisection then propagation

$x(0) = \{x(0)_{left}, x(0)_{right}\}$

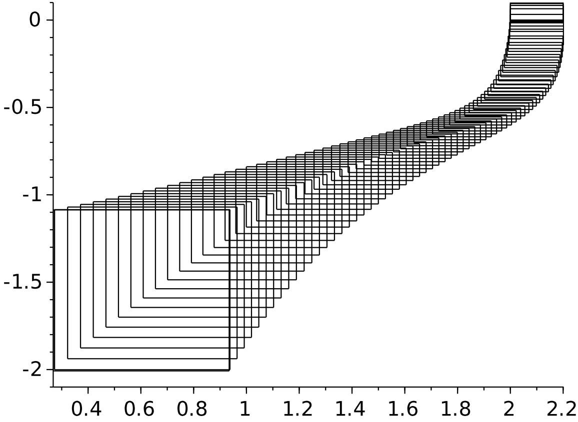
1 simulation then 2 propagations vs 3 simulations

Time : 2+2*1 = 4 sec. vs 2*3 = 6 sec.



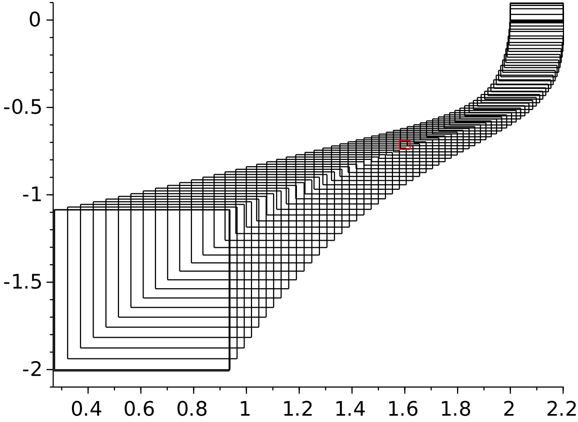
Contraction then propagation : Initial

$y(1.0) \in [1.58, 1.62]$ and $x(1.0) \in [-0.74, -0.69]$



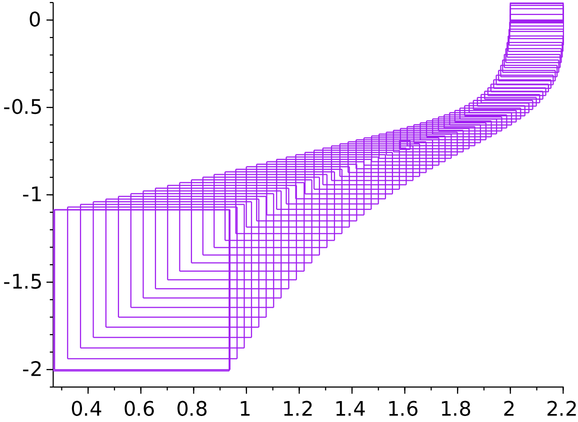
Contraction then propagation : With Measure

$y(1.0) \in [1.58, 1.62]$ and $x(1.0) \in [-0.74, -0.69]$



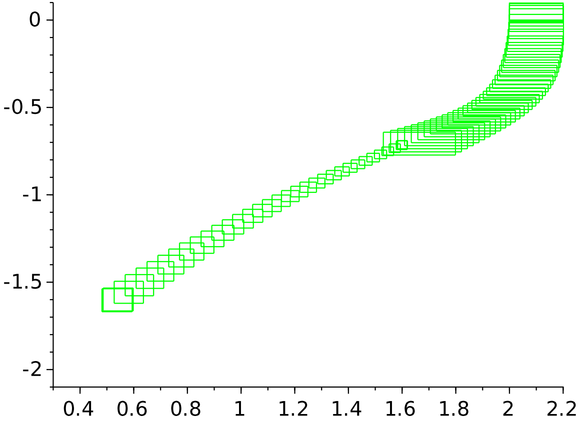
Contraction then propagation : Contraction

$y(1.0) \in [1.58, 1.62]$ and $x(1.0) \in [-0.74, -0.69]$



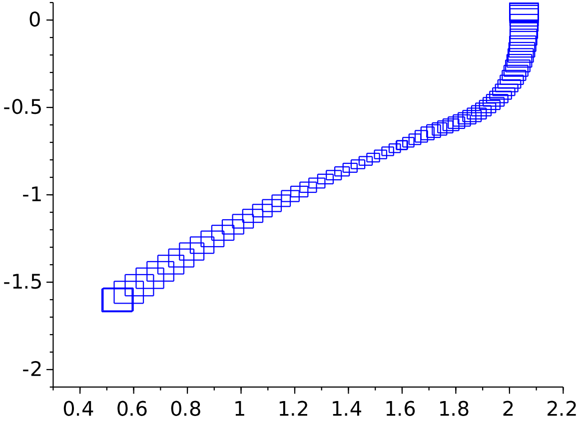
Contraction then propagation : Forward

$y(1.0) \in [1.58, 1.62]$ and $x(1.0) \in [-0.74, -0.69]$



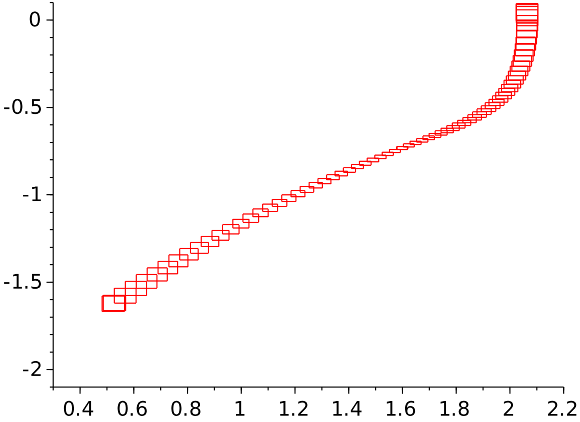
Contraction then propagation : Backward

$y(1.0) \in [1.58, 1.62]$ and $x(1.0) \in [-0.74, -0.69]$



Contraction then propagation : Fixed Point

$y(1.0) \in [1.58, 1.62]$ and $x(1.0) \in [-0.74, -0.69]$



Discussion

Remarks

- Propagation of a contraction on one state is close to a contraction on a tube [2]
- Easy to generalize to a contraction on parameters by changing $f(y, [p])$ by $f(y, Ctc([p]))$
- Easy to generalize to interval of time ($t = [\underline{t}, \overline{t}]$ same job for $\underline{t}$ and $\overline{t}$)
- Periodicity, limit cycle, etc.

Future work

Finalize implementation of CSDP in DynIBEX and applications !

[2] A. Bethencourt, and L. Jaulin, Solving Non-Linear Constraint Satisfaction Problems Involving Time-Dependant Functions, *Mathematics in Computer Science*, 2014.

Questions ?