Validated simulation of ODEs

Julien Alexandre dit Sandretto

Department U2IS

ENSTA Paris

SSC310-2020

ENSTA

IP PARIS

Initial Value Problem of Ordinary Differential Equations

Classical problem

Consider an IVP for ODE, over the time interval $[0, T]$

$$\dot{\mathbf{y}} = f(\mathbf{y}) \quad \text{with} \quad \mathbf{y}(0) = \mathbf{y}_0$$

This IVP has a unique solution $\mathbf{y}(t; \mathbf{y}_0)$ if $f : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz.

Interval IVP

$$\dot{\mathbf{y}} = f(\mathbf{y}, p) \quad \text{with} \quad \mathbf{y}(0) \in [\mathbf{y}_0] \quad \text{and} \quad p \in [p]$$

Numerical Integration

How compute $\mathbf{y}(t) = \mathbf{y}_0 + \int_0^t f(\mathbf{y}(s))ds$?

Goal of numerical integration

- Compute a sequence of time instants: $t_0 = 0 < t_1 < \cdots < t_n = T$
- Compute a sequence of values: $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_n$ such that

$$\forall i \in [0, n], \quad \mathbf{y}_i \approx \mathbf{y}(t_i; \mathbf{y}_0) \ .$$

Goal of validated numerical integration

- Compute a sequence of time instants: $t_0 = 0 < t_1 < \cdots < t_n = T$
- Compute a sequence of values: $[\mathbf{y}_0], [\mathbf{y}_1], \ldots, [\mathbf{y}_n]$ such that

$$\forall i \in [0, n], \quad [\mathbf{y}_i] \ni \mathbf{y}(t_i; \mathbf{y}_0) \ .$$

Problem of integral computation

Discrete system given by $\mathbf{y}_{n+1} = \mathbf{y}_n + \int_0^h f(\mathbf{y}(s))ds$

Bounding of $\int_0^h f(\mathbf{y}(s))ds$

If $\mathbf{y}(s)$ is bounded s.t. $\mathbf{y}(s) \in [\mathbf{x}], \quad \forall s \in [0, h]$, then

$$\int_0^h f([\mathbf{x}])ds \subset [0, h] \cdot [f]([\mathbf{x}])$$

How bound $\mathbf{y}(s)$?

Complex, it is what we are trying to compute !
We note by $[\tilde{\mathbf{y}}_n] \supset \{\mathbf{y}(s), s \in [t_n, t_{n+1}]\}$

Picard-Lindelöf (or Cauchy-Lipschitz)

Theorem (Banach fixed-point theorem)

Let (K, d) *a complete metric space and let* $g : K \to K$ *a contraction that is for all* x, y *in* K *there exists* $c \in]0, 1[$ *such that* $d\left(g(x), g(y)\right) \leqslant c \cdot d(x, y)$, *then* g *has a unique fixed-point in* K.

We consider the space of continuously differentiable functions $\mathcal{C}^0([t_j, t_{j+1}], \mathbb{R}^n)$ and the Picard-Lindelöf operator

$$\mathbf{p_f}(\mathbf{y}) = t \mapsto \mathbf{y}_j + \int_{t_j}^{t} \mathbf{f}(\mathbf{y}(s))ds \ , \quad \text{with} \quad \mathbf{y}_j = \mathbf{y}(t_j) \qquad (1)$$

If this operator is a contraction then its solution is unique and its solution is the solution of IVP.

Interval counterpart of Picard-Lindelöf

With a first order integration scheme that is for $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ a continuous function and $[\mathbf{a}] \subset \mathbb{IR}^n$, we have

$$\int_{\underline{a}}^{\overline{a}} f(s)ds \in (\underline{\mathbf{a}} - \overline{\mathbf{a}})f([\mathbf{a}]) = w([\mathbf{a}])\mathbf{f}([\mathbf{a}]) \ , \qquad (2)$$

we can define a simple enclosure function of Picard-Lindelöf such that

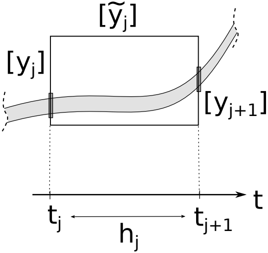
$$[\mathbf{p_f}]([\mathbf{r}]) \stackrel{\text{def}}{=} [\mathbf{y}_j] + [0, h] \cdot \mathbf{f}([\mathbf{r}]) \ , \qquad (3)$$

with $h = t_{j+1} - t_j$ the step-size. In consequence, if one can find $[\mathbf{r}]$ such that $[\mathbf{p_f}]([\mathbf{r}]) \subseteq [\mathbf{r}]$ then $[\tilde{\mathbf{y}}_j] \subseteq [\mathbf{r}]$ by the Banach fixed-point theorem.

Interval counterpart of Picard-Lindelöf

We can then build the Lohner 2-steps method:

1. Find $[\tilde{\mathbf{y}}_j]$ and h_j with Picard-Lindelöf operator and Banach's theorem
2. Compute $[\mathbf{y}_{j+1}]$ with a validated integration scheme: Taylor or Runge-Kutta



It is important to obtain $[\tilde{\mathbf{y}}_j]$ and $[\mathbf{y}_{j+1}]$ as tight as possible

Integration scheme at order higher than one: Taylor for example

Integration scheme

Two main approaches:

- **Taylor series** (Vnode, CAPD, etc.):
  $\mathbf{y}_{j+1} = \mathbf{y}_j + \sum_1^p h^i f^{[i]}(\mathbf{y}_j) + \mathcal{O}(h^{p+1})$ with $f^{[i]}$ the i^{th} term of serie expansion of f.

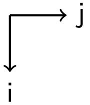
  $\mathcal{O}(h^{p+1})$ can be easily bounded by the Lagrange remainder of serie s.t. $\mathcal{O}(h^{p+1}) = f^{[p+1]}(\xi)$, with $\xi \in [\tilde{\mathbf{y}}_j]$, and then $\mathcal{O}(h^{p+1}) \in f^{[p+1]}(\tilde{\mathbf{y}}_j)$

- **Runge-Kutta methods** (DynIBEX):
  $\mathbf{y}_{j+1} = \Phi(\mathbf{y}_j, f, p) + LTE$, with Φ any RK method and LTE the local truncation error.

Runge-Kutta methods

s-stage Runge-Kutta methods are described by a Butcher tableau

$$
\begin{array}{c|cccc}
c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\
\vdots & \vdots & \vdots & & \vdots \\
c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\
\hline
 & b_1 & b_2 & \cdots & b_s
\end{array}
$$



Which induces the following recurrence:

$$
\mathbf{k}_i = f\left(t_j + c_i h_j, \quad \mathbf{y}_j + h \sum_{l=1}^{s} a_{il}\mathbf{k}_l\right) \quad \mathbf{y}_{j+1} = \mathbf{y}_j + h \sum_{i=1}^{s} b_i \mathbf{k}_i
$$

- **Explicit** method (ERK) if $a_{il} = 0$ is $i \leqslant l$
- **Diagonal Implicit** method (DIRK) if $a_{il} = 0$ is $i \leqslant l$ and at least one $a_{ii} \neq 0$
- **Implicit** method (IRK) otherwise

Explicit methods

Interval extensions

1. Computation of $\mathbf{k}_1 = f(\mathbf{y}_j)$, $\mathbf{k}_2 = f(\mathbf{y}_j + h \cdot a_{21} \cdot \mathbf{k}_1)$, ..., $\mathbf{k}_i = f(\mathbf{y}_j + h \sum_{\ell=1}^{i-1} a_{i\ell}\mathbf{k}_\ell)$, ..., $\mathbf{k}_s = f(\mathbf{y}_j + h \sum_{\ell=1}^{s-1} a_{s\ell}\mathbf{k}_\ell)$
2. Computation of $\mathbf{y}_{j+1} = \mathbf{y}_j + h \sum_{i=1}^{s} b_i \mathbf{k}_i + \text{LTE}$

$\Rightarrow$ with interval arithmetic (natural extension)

Example of HEUN

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}$$

$[\mathbf{k}_1] = [f]([\mathbf{y}_j])$, $\quad [\mathbf{k}_2] = [f]([\mathbf{y}_j] + h[\mathbf{k}_1])$,
$[\mathbf{y}_{j+1}] = [\mathbf{y}_j] + h([\mathbf{k}_1] + [\mathbf{k}_2])/2$

Implicit Schemes

Example of Radau IIA

$$\begin{array}{c|cc} 1/3 & 5/12 & -1/12 \\ 1 & 3/4 & 1/4 \\ \hline & 3/4 & 1/4 \end{array}$$

$$[\mathbf{k}_1] = [f]([\mathbf{y}_j] + h(5[\mathbf{k}_1]/12 - [\mathbf{k}_2]/12)),$$
$$[\mathbf{k}_2] = [f]([\mathbf{y}_j] + h(3[\mathbf{k}_1]/4 + [\mathbf{k}_2]/4)$$

We need to solve this system of implicit equations !

Solve implicit scheme with a contractor point of view

$\mathbf{k}_1$ is the approximate of $f(\mathbf{y}(t_j + h/3))$, but by construction $\mathbf{y}(t_j + h/3) \in [\tilde{\mathbf{y}}_j]$, then $[\mathbf{k}_1] \subset f([\tilde{\mathbf{y}}_j])$ (same for $\mathbf{k}_2$)

Algorithm based on contraction

Require: f, $[\tilde{\mathbf{y}}_j]$, $[\mathbf{y}_j]$, LTE

  $[\mathbf{k}_1] = [f]([\tilde{\mathbf{y}}_j])$ and $[\mathbf{k}_2] = [f]([\tilde{\mathbf{y}}_j])$

  while $[\mathbf{k}_1]$ or $[\mathbf{k}_2]$ improved **do**

    $[\mathbf{k}_1] = [\mathbf{k}_1] \cap [f]([\mathbf{y}_j] + h(5[k_1]/12 - [\mathbf{k}_2]/12))$

    $[\mathbf{k}_2] = [\mathbf{k}_2] \cap [f]([\mathbf{y}_j] + h(3[\mathbf{k}_1]/4 + [\mathbf{k}_2]/4)$

  end while

  $[\mathbf{y}_{j+1}] = [\mathbf{y}_j] + h(3[\mathbf{k}_1] + [\mathbf{k}_2])/4 + \text{LTE}$

  return $[\mathbf{y}_{j+1}]$

How to compute the LTE ?

$$\mathbf{y}(t_n; \mathbf{y}_{n-1}) - \mathbf{y}_n = C \cdot \left(h^{p+1}\right) \quad \text{with} \quad C \in \mathbb{R}.$$

Order condition

This condition states that a method of Runge-Kutta family is of order p **iff**

- the Taylor expansion of the exact solution
- and the Taylor expansion of the numerical methods

have the same $p+1$ first coefficients.

Consequence

The LTE is the **difference of Lagrange remainders of two Taylor expansions**

A quick view of Runge-Kutta order condition theory

Starting from $\mathbf{y}^{(q)} = (f(\mathbf{y}))^{(q-1)}$ and with the Chain rule, we have

High order derivatives of exact solution $\mathbf{y}$

$$\begin{aligned}
\dot{\mathbf{y}} &= f(\mathbf{y}) \\
\ddot{\mathbf{y}} &= f'(\mathbf{y})\dot{\mathbf{y}} && f'(\mathbf{y}) \text{ is a linear map} \\
\mathbf{y}^{(3)} &= f''(\mathbf{y})(\dot{\mathbf{y}},\dot{\mathbf{y}}) + f'(\mathbf{y})\ddot{\mathbf{y}} && f''(\mathbf{y}) \text{ is a bi-linear map} \\
\mathbf{y}^{(4)} &= f'''(\mathbf{y})(\dot{\mathbf{y}},\dot{\mathbf{y}},\dot{\mathbf{y}}) + 3f''(\mathbf{y})(\ddot{\mathbf{y}},\dot{\mathbf{y}}) + f'(\mathbf{y})\mathbf{y}^{(3)} && f'''(\mathbf{y}) \text{ is a tri-linear map} \\
\mathbf{y}^{(5)} &= f^{(4)}(\mathbf{y})(\dot{\mathbf{y}},\dot{\mathbf{y}},\dot{\mathbf{y}},\dot{\mathbf{y}}) + 6f'''(\mathbf{y})(\ddot{\mathbf{y}},\dot{\mathbf{y}},\dot{\mathbf{y}}) && \vdots \\
&\quad + 4f''(\mathbf{y})(\mathbf{y}^{(3)},\dot{\mathbf{y}}) + 3f''(\mathbf{y})(\ddot{\mathbf{y}},\ddot{\mathbf{y}}) + f'(\mathbf{y})\mathbf{y}^{(4)} \\
&\vdots
\end{aligned}$$

A quick view of Runge-Kutta order condition theory

Inserting the value of $\dot{\mathbf{y}}$, $\ddot{\mathbf{y}}$, ..., we have:

High order derivatives of exact solution $\mathbf{y}$

$$\begin{aligned}
\dot{\mathbf{y}} &= f \\
\ddot{\mathbf{y}} &= f'(f) \\
\mathbf{y}^{(3)} &= f''(f,f) + f'(f'(f)) \\
\mathbf{y}^{(4)} &= f'''(f,f,f) + 3f''(f'f,f) + f'(f''(f,f)) + f'(f'(f'(f))) \\
&\vdots
\end{aligned}$$

- `Elementary differentials` , such as $f''(f,f)$, are denoted by $F(\tau)$

Remark a tree structure is made apparent in these computations

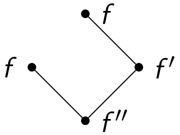
A quick view of Runge-Kutta order condition theory

Rooted trees

- f is a leaf
- f' is a tree with one branch, ..., $f^{(k)}$ is a tree with k branches

Example

$f''(f'f, f)$ is associated to



Remark: this tree is not unique e.g., symmetry

A quick view of Runge-Kutta order condition theory

Theorem 1 (Butcher, 1963)

The qth derivative of the **exact solution** is given by

$$\mathbf{y}^{(q)} = \sum_{r(\tau)=q} \alpha(\tau) F(\tau)(\mathbf{y}_0) \quad \text{with} \quad \begin{array}{l} r(\tau) \text{ the order of } \tau \text{ i.e., number of nodes} \\ \alpha(\tau) \text{ a positive integer} \end{array}$$

We can do the same for the numerical solution

Theorem 2 (Butcher, 1963)

The qth derivative of the **numerical solution** is given by

$$\mathbf{y}_1^{(q)} = \sum_{r(\tau)=q} \gamma(\tau)\phi(\tau)\alpha(\tau) F(\tau)(\mathbf{y}_0) \text{ with } \begin{array}{l} \gamma(\tau) \text{ a positive integer} \\ \phi(\tau) \text{ depending on a Butcher tableau} \end{array}$$

Theorem 3, order condition (Butcher, 1963)

A Runge-Kutta method has order p iff $\phi(\tau) = \frac{1}{\gamma(\tau)} \quad \forall \tau, r(\tau) \leqslant p$

LTE formula for **explicit and implicit** Runge-Kutta

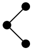
From Th. 1 and Th. 2, if a Runge-Kutta has order p then

$$\mathbf{y}(t_n; \mathbf{y}_{n-1}) - \mathbf{y}_n = \frac{h^{p+1}}{(p+1)!} \sum_{r(\tau)=p+1} \alpha(\tau) \left[1 - \gamma(\tau)\phi(\tau)\right] F(\tau)(\mathbf{y}(\xi))$$

$$\xi \in [t_{n-1}, t_n]$$

- $\alpha(\tau)$ and $\gamma(\tau)$ are positive integer (with some combinatorial meaning)
- $\phi(\tau)$ function of the coefficients of the RK method,

Example

$\phi\Big($  $\Big)$ is associated to $\displaystyle\sum_{i,j=1}^{s} b_i a_{ij} c_j$ with $\displaystyle c_j = \sum_{k=1}^{s} a_{jk}$

Note: $\mathbf{y}(\xi)$ may be over-approximated using Interval Picard-Lindelöf operator.

Implementation of LTE formula

Elementary differentials

$$F(\tau)(\mathbf{y}) = f^{(m)}(\mathbf{y})\left(F(\tau_1)(\mathbf{y}), \ldots, F(\tau_m)(\mathbf{y})\right) \quad \text{for} \quad \tau = [\tau_1, \ldots, \tau_m]$$

translate as a sum of partial derivatives of f associated to sub-trees

Notations

- n the state-space dimension
- p the order of a Rung-Kutta method

Two ways of computing $F(\tau)$

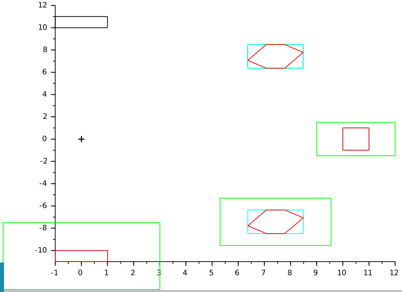
1. **Direct form**: complexity $\mathcal{O}(n^{p+1})$
2. **Factorized form**: complexity $\mathcal{O}(n(p+1)^{\frac{5}{2}})$ based on Automatic Differentiation

Wrapping effect

Consider the following IIVP: $\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} -y_2 \\ y_1 \end{pmatrix}$
with $y_1(0) \in [-1, 1]$, $y_2(0) \in [10, 11]$. Exact solution is

$$\mathbf{y}(t) = A(t)\mathbf{y}_0 \quad \text{with} \quad A(t) = \begin{pmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{pmatrix}$$

We compute periodically at $t = \frac{\pi}{4} n$ with $n = 1, \ldots, 4$



Wrapping effect comparison (black: initial, green: interval, blue: interval from QR, red: zonotope from affine)

Solution to wrapping effect

One solution is the centered form of Taylor series, coupled with QR

Taylor integration

$[\mathbf{y}_{j+1}] = [\mathbf{y}_j] + \sum_{i=1}^{N-1} h^i f^{[i-1]}([\mathbf{y}_j]) + h^N f^{[N-1]}([\tilde{\mathbf{y}}_j])$
Each $f^{[i-1]}([\mathbf{y}_j])$ evaluated in centered form:

$$f^{[i-1]}(\mathrm{m}([\mathbf{y}_j])) + J([\mathbf{y}_j])^T([\mathbf{y}_j] - \mathrm{m}([\mathbf{y}_j])) \ ,$$

and a QR-decomposition of J is used to reduce the wrapping effect...

Geometric sense

Consists on a rotation of the evaluation. But in $\mathcal{O}(n^3)$

Another solution: Affine arithmetic

A different arithmetic than interval

Represented by an *affine form* $\hat{x}$ (also called a *zonotope*):

$$\hat{x} = \alpha_0 + \sum_{i=1}^{n} \alpha_i \varepsilon_i$$

where α_i real numbers, α_0 the *center*, and ε_i are intervals $[-1, 1]$

Geometric sense

Represents a zonotope, a convex polytope with central symmetry (not affected by rotation !)

Affine arithmetic

An interval $a = [a_1, a_2]$ in affine form:
$\hat{x} = \alpha_0 + \alpha_1 \varepsilon$ with $\alpha_0 = (a_1 + a_2)/2$ and $\alpha_1 = (a_2 - a_1)/2$.
Usual operations: $\hat{x} = \alpha_0 + \sum_{i=1}^{n} \alpha_i \varepsilon_i$ and $\hat{y} = \beta_0 + \sum_{i=1}^{n} \beta_i \varepsilon_i$,
then with $a, b, c \in \mathbb{R}$

$$a\hat{x} + b\hat{y} + c = (a\alpha_0 + b\beta_0 + c) + \sum_{i=1}^{n} (a\alpha_i + b\beta_i)\varepsilon_i \quad .$$

Multiplication creates new noise symbols:

$$\hat{x} \times \hat{y} = \alpha_0 \alpha_1 + \sum_{i=1}^{n} (\alpha_i \beta_0 + \alpha_0 \beta_i)\varepsilon_i + \nu \varepsilon_{n+1} \quad ,$$

where $\nu = (\sum_{i=1}^{n} |\alpha_i|) \times (\sum_{i=1}^{n} |\beta_i|)$ over-approximates the error of linearization.
Other operations, like sin, exp, are evaluated using either the Min-Range method or a Chebychev approximation

Enclosure part of the algorithm

Compute $\tilde{\mathbf{y}}_1 = PL(\tilde{\mathbf{y}}_0)$
iter $= 1$
while $(\tilde{\mathbf{y}}_1 \not\subset \tilde{\mathbf{y}}_0)$ and (iter $<$ size($\mathbf{f}$) $+ 1$) **do**
    $\tilde{\mathbf{y}}_0 = \tilde{\mathbf{y}}_1$
    Compute $\tilde{\mathbf{y}}_1$ with $PL(\tilde{\mathbf{y}}_0)$
    iter $=$ iter $+ 1$
end while
if $(\tilde{\mathbf{y}}_1 \subset \tilde{\mathbf{y}}_0)$ **then**
    Compute lte $=$ LTE$(\tilde{\mathbf{y}}_1)$
    if lte $>$ tol **then**
        $h = h/2$, restart
    end if
else
    $h = h/2$, restart
end if

Stepsize h decreases but never increases: Zenon problem

Stepsize controller

If first step is achieved with success, multiply h by a factor function of method order and LTE:

$$fac = \left(\frac{\text{tol}}{\text{LTE}}\right)^{\frac{1}{p}}$$

Validated integration in a contractor formalism

Contractor for $[\tilde{\mathbf{y}}_j]$

After Picard-lindelöf contractance obtained :
$Ctc_{PL}([\tilde{\mathbf{y}}_j]) \triangleq [\tilde{\mathbf{y}}_j] \cap PL([\mathbf{y}_j], [\tilde{\mathbf{y}}_j])$ till a fixed point

Contractor for $[\mathbf{y}_{j+1}]$

$Ctc_{RK}([\mathbf{y}_{j+1}]) \triangleq [\mathbf{y}_{j+1}] \cap RK([\mathbf{y}_j]) + \text{LTE}([\tilde{\mathbf{y}}_j])$

Additive constraints

For constraint valid all the time

$$\forall t, g(\mathbf{y}(t)) = 0$$

Coming from mechanical constraints, energy conservation, etc.

A new contractor

Based on Fwd/Bwd contractor on g combined with previous Ctc:

- $Ctc_{FB}([\tilde{\mathbf{y}}_j]) \cap Ctc_{PL}([\tilde{\mathbf{y}}_j])$
- $Ctc_{FB}([\mathbf{y}_{j+1}]) \cap Ctc_{RK}([\mathbf{y}_{j+1}])$

Additive constraints

For constraint valid all the time

$$\forall t, g(\mathbf{y}(t)) = 0$$

Coming from mechanical constraints, energy conservation, etc.

A new contractor

Based on Fwd/Bwd contractor on g combined with previous Ctc:

- $Ctc_{FB}([\tilde{\mathbf{y}}_j]) \cap Ctc_{PL}([\tilde{\mathbf{y}}_j])$
- $Ctc_{FB}([\mathbf{y}_{j+1}]) \cap Ctc_{RK}([\mathbf{y}_{j+1}])$

But ? The second one is often a bad idea, lost of noise symbols !

Additive constraints

For constraint valid all the time

$$\forall t, g(\mathbf{y}(t)) = 0$$

Coming from mechanical constraints, energy conservation, etc.

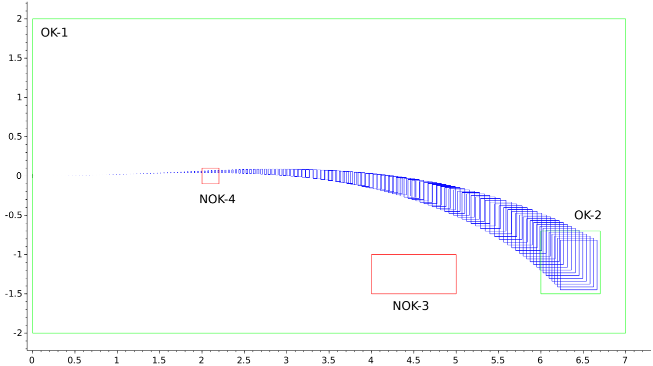
A new contractor

Based on Fwd/Bwd contractor on g combined with previous Ctc:

- $Ctc_{FB}([\tilde{\mathbf{y}}_j]) \cap Ctc_{PL}([\tilde{\mathbf{y}}_j])$
- $Ctc_{FB}([\mathbf{y}_{j+1}]) \cap Ctc_{RK}([\mathbf{y}_{j+1}])$

But ? The second one is often a bad idea, lost of noise symbols !
Because intersection of zonotopes is not a zonotope...

Temporal constraints



OK-1: safe zone, OK-2: Goal, NOK-3: obstacle, NOK-4: forbidden zone at a given time, ...

Temporal constraints

Constraint Satisfaction Differential Problem (CSDP)

With a tube $R(t)$, such that $y(t) \in R(t), \forall t$ (obtained with validated simulation:

Verbal property	CSDP translation
Stay in $\mathcal{A}$ (until τ)	$R(t) \subset Int(\mathcal{A}), \forall t \ (t < \tau)$
In $\mathcal{A}$ at τ	$R(\tau) \subset Int(\mathcal{A})$
Has crossed $\mathcal{A}$ (before τ)	$\exists t, R(t) \cap \Box\mathcal{A} \neq \emptyset \ (t < \tau)$
Go out $\mathcal{A}$ (before τ)	$\exists t, R(t) \cap \Box\mathcal{A} = \emptyset \ (t < \tau)$
Has reached $\mathcal{A}$	$R(T) \cap \Box\mathcal{A} \neq \emptyset$
Finish in $\mathcal{A}$	$R(T) \subset Int(\mathcal{A})$

Bibliography

- Solving Ordinary Differential Equations I: Nonstiff Problems - Hairer et al. - 2009 - Springer
- Coefficients for the study of Runge-Kutta integration processes - Butcher - 1963 - Journal of the Australian Mathematical Society
- Interval Analysis - Moore - 1966 - Prentice-hall
- Validated solutions of initial value problems for ordinary differential equations - Nedialkov et al. - 1999 - Appl. Math. and Comp.
- Validated Explicit and Implicit Runge-Kutta Methods - Sandretto et al. - 2016 - Reliable Computing
- DynIbex - `http://perso.ensta-paristech.fr/~chapoutot/dynibex/`

Do it yourself

Consider an IVP - Van der Pol oscillator

$$\dot{y} = \begin{pmatrix} y_1 \\ \mu(1 - y_0^2)y_1 - y_0 \end{pmatrix}$$

with $\mu = 1$ and $y(0) = (2;0)^T$

To Do

Compute the simulation of this ivp with DynIbex !

- Write a function, an IVP, launch simulation till t = 10s
- Export and plot the result (with vibes or matlab)
- Find the “best” method and precision to obtain a nice picture
- Play with μ (0.2, 2, etc.)
- What do you see after $\mu \geq 5$?
- What do you need to change ?