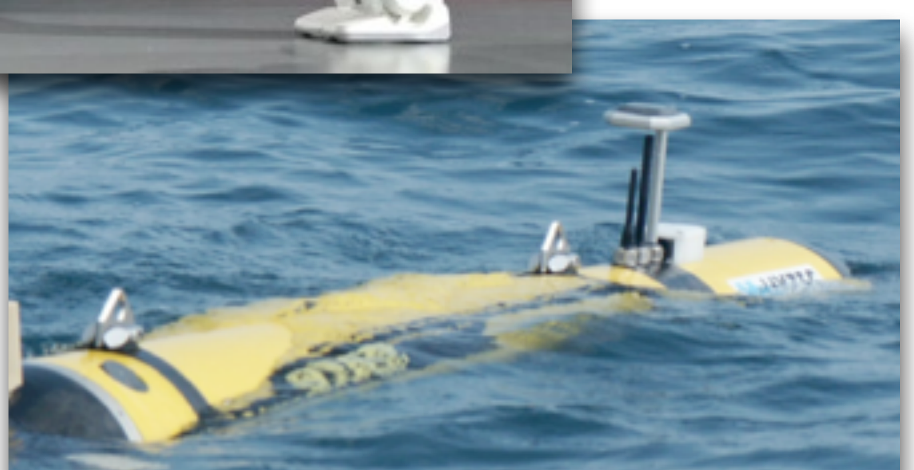
On the verification of nonlinear hybrid dynamical systems

Nacim RAMDANI,
Univ. Orléans, EA 4229 PRISME à Bourges.

GT MEA, 19 Mars 2015, Paris.

UNIVERSITE D'ORLEANS

Hybrid Cyber-Physical Systems



- Interaction discrete
  + continuous dynamics
- Safety-critical
  embedded systems
- Networked
  autonomous systems

UNIVERSITE D'ORLEANS

Hybrid Cyber-Physical Systems

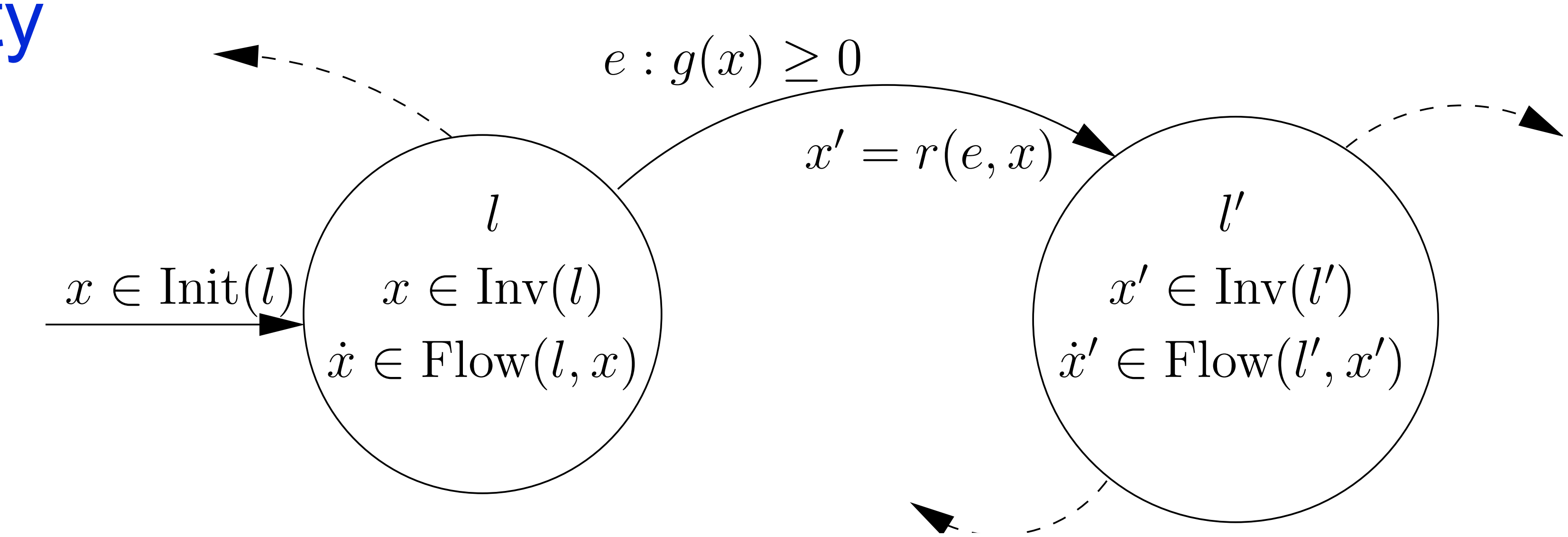
- **Verification**
  - **Numerical proof**
  - **Falsification via counter-example**

Hybrid Cyber-Physical Systems

UNIVERSITE D'ORLEANS

Modelling → **hybrid automaton** (Alur, et al. 1995)

- Non-linear continuous dynamics
- Bounded uncertainty



$$H = (\mathcal{Q}, \mathcal{D}, \mathcal{P}, \Sigma, \mathcal{A}, \mathsf{Inv}, \mathcal{F}),$$

Continuous dynamics

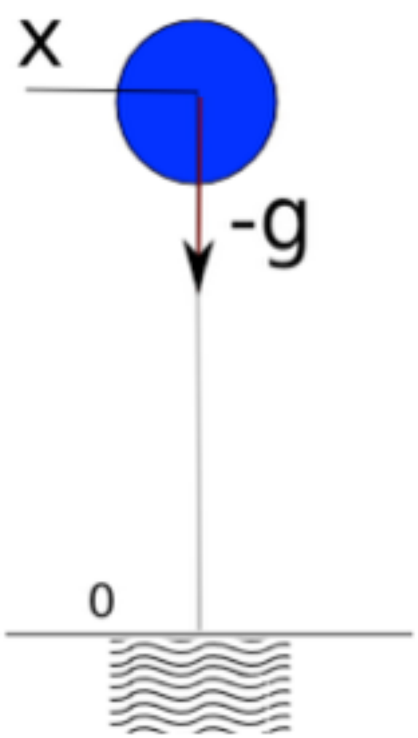
$$\begin{aligned} \mathsf{flow}(q) :&\quad \dot{\mathbf{x}}(t) = f_q(\mathbf{x}, \mathbf{p}, t), \\ \mathsf{Inv}(q) :&\quad \nu_q(\mathbf{x}(t), \mathbf{p}, t) < 0, \end{aligned}$$

Discrete dynamics

$$\begin{aligned} \mathcal{A} \ni e :&\quad (q \to q') = (q, \mathsf{guard}, \sigma, \rho, q'), \\ \mathsf{guard}(e) :&\quad \gamma_e(\mathbf{x}(t), \mathbf{p}, t) = 0, \end{aligned}$$

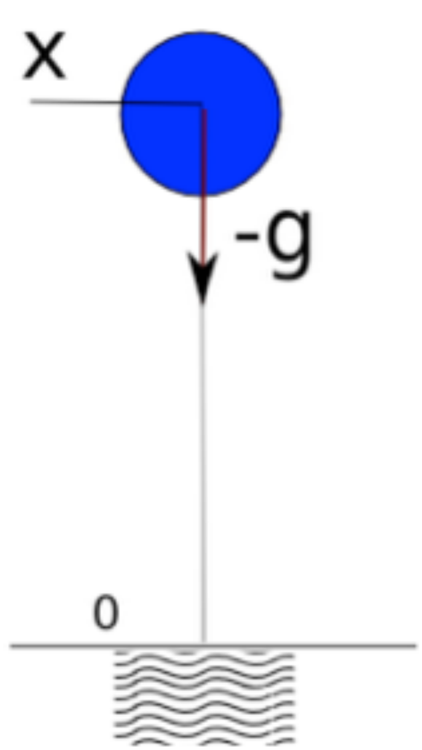
$$t_0 \leq t \leq t_N, \quad \mathbf{x}(t_0) \in \mathbb{X}_0 \subseteq \mathbb{R}^n, \quad \mathbf{p} \in \mathbb{P}$$

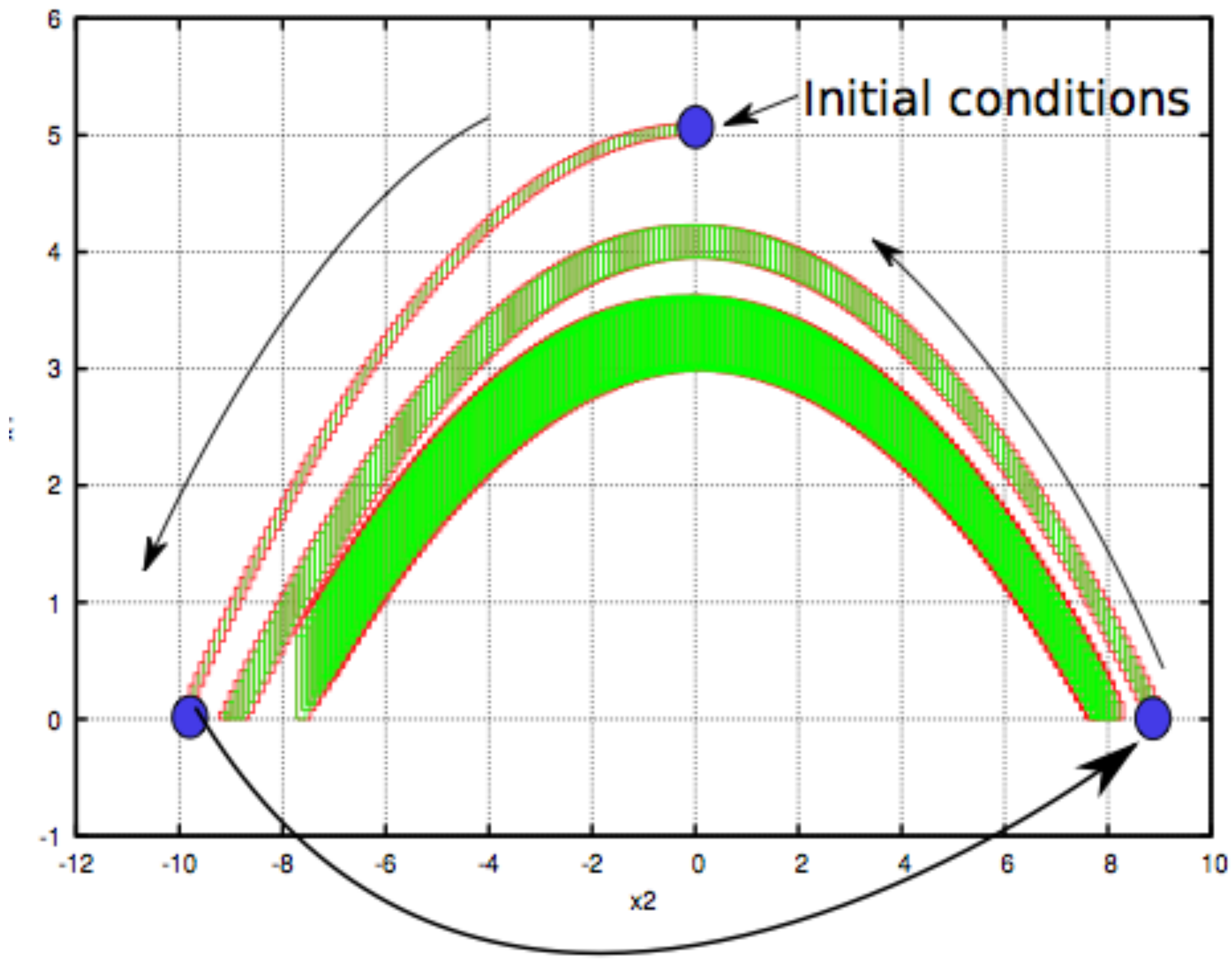
- **Example :** bouncing ball



Hybrid Cyber-Physical Systems

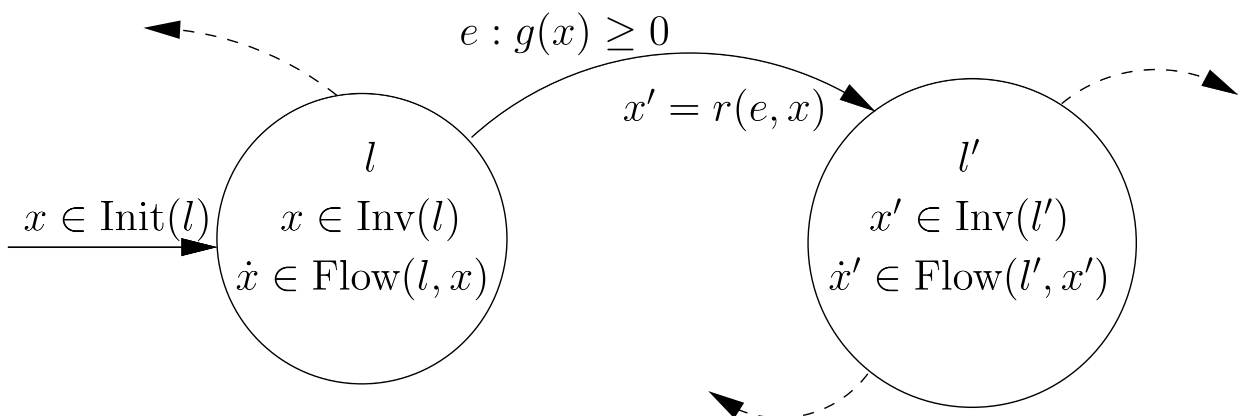
- **Example :** bouncing ball

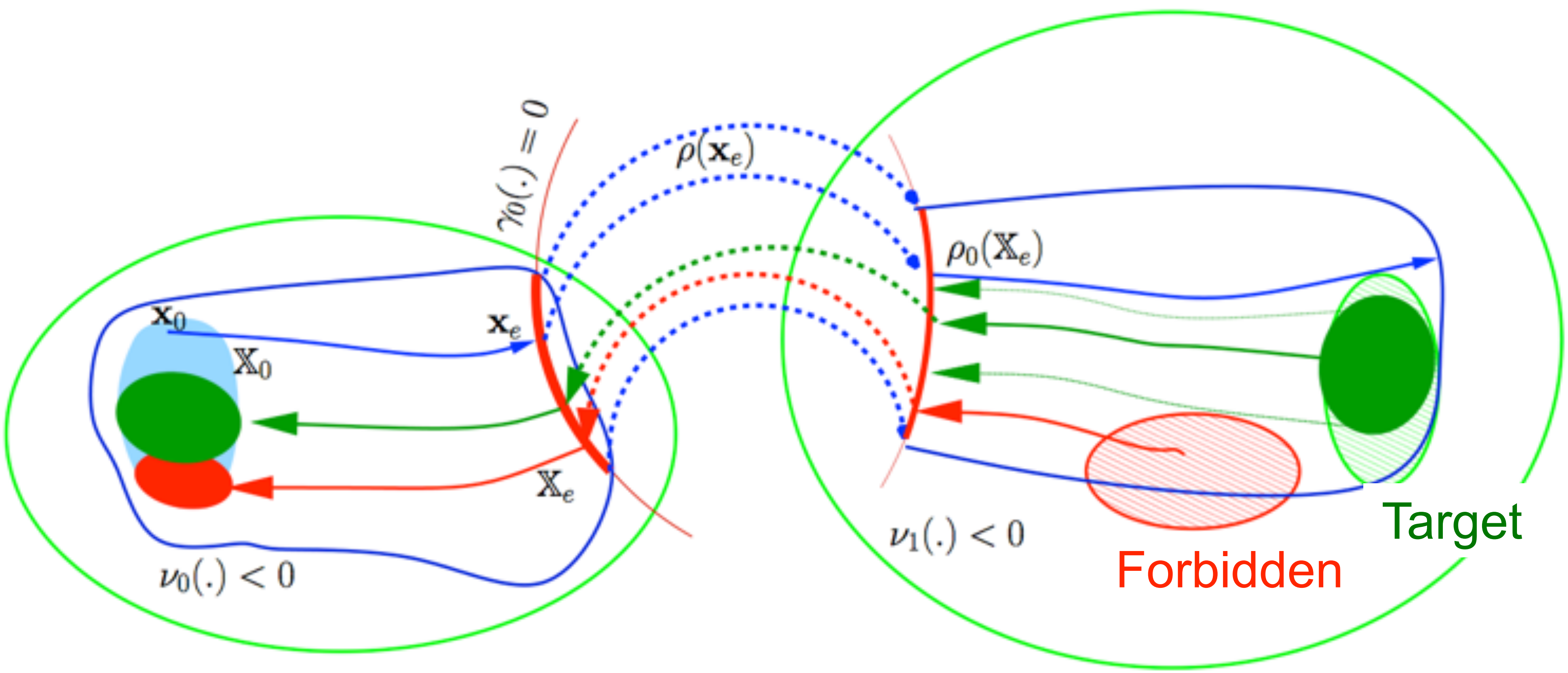




Hybrid Cyber-Physical Systems

- **Verification**
  - ***Modelling :***
  - ***Property specification*** **:**
  - **Verification algorithm :**
    - **Hybrid / Continuous reachability**





UNIVERSITE D'ORLEANS

Outline

- Safety Critical Systems
- Nonlinear Continuous Reachability
- Nonlinear Hybrid Reachability
- Satisfiability mod ODE

Hybrid Cyber-Physical Systems

Continuous reachability

$$\mathbb{R}([t_0,\, t]\,;\mathbb{X}_0) = \left\{ \begin{array}{l} \mathbf{x}(\tau),\;\; t_0 \leq \tau \leq t \;| \\ \dot{\mathbf{x}}(\tau) = f(\mathbf{x}, \mathbf{p}, \tau) \;\wedge\; \mathbf{x}(t_0) \in \mathbb{X}_0 \;\wedge\; \mathbf{p} \in \mathbb{P} \end{array} \right\}$$

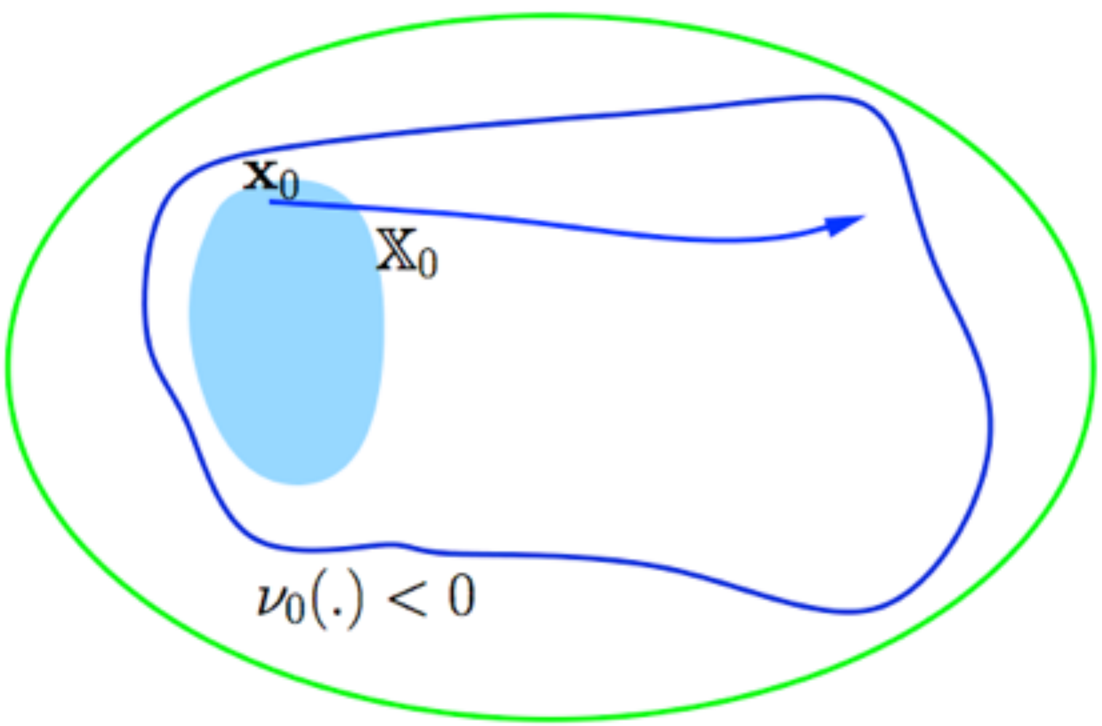
- Set integration
  - Interval Taylor methods
  - Bracketing enclosures

Hybrid Cyber-Physical Systems

Continuous reachability

$$\mathbb{R}([t_0,\, t]\,;\mathbb{X}_0) = \left\{ \begin{array}{l} \mathbf{x}(\tau),\;\; t_0 \le \tau \le t \mid \\ \dot{\mathbf{x}}(\tau) = f(\mathbf{x}, \mathbf{p}, \tau) \;\wedge\; \mathbf{x}(t_0) \in \mathbb{X}_0 \;\wedge\; \mathbf{p} \in \mathbb{P} \end{array} \right\}$$

- Set integration
  - Interval Taylor methods
  - Bracketing enclosures



Nonlinear Set Integration

- **Guaranteed set integration with Taylor methods**
  - (Moore,66) (Lohner,88) (Rihm,94) (Berz,98) (Nedialkov,99)

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, \mathbf{p}, t), \quad t_0 \leq t \leq t_N,\ \mathbf{x}(t_0) \in [\mathbf{x}_0],\ \mathbf{p} \in [\mathbf{p}]$$

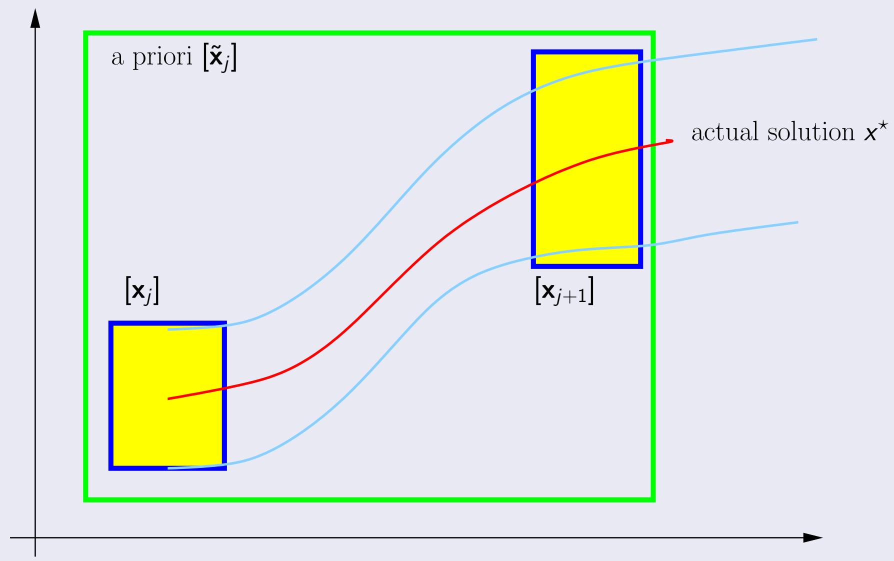
Nonlinear Set Integration

Guaranteed set integration with Taylor methods

- (Moore,66) (Lohner,88) (Rihm,94) (Berz,98) (Nedialkov,99)

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, \mathbf{p}, t), \quad t_0 \leq t \leq t_N,\ \mathbf{x}(t_0) \in [\mathbf{x}_0],\ \mathbf{p} \in [\mathbf{p}]$$

Time grid $\rightarrow \quad t_0 < t_1 < t_2 < \cdots < t_N$



- Proof of existence
- Yield *a priori* solution $[\tilde{\mathbf{x}}_j]$: $\forall \tau \in [t_j,\ t_{j+1}] \quad x(\tau) \in [\tilde{\mathbf{x}}_j]$

Nonlinear Set Integration

- **Guaranteed set integration with Taylor methods**
  - (Moore,66) (Lohner,88) (Rihm,94) (Berz,98) (Nedialkov,99)

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, \mathbf{p}, t), \quad t_0 \leq t \leq t_N, \ \mathbf{x}(t_0) \in [\mathbf{x}_0], \ \mathbf{p} \in [\mathbf{p}]$$

$$[\mathbf{x}_j] + [0, h]\mathbf{f}([\tilde{\mathbf{x}}_j]) \subseteq [\tilde{\mathbf{x}}_j]$$

Nonlinear Set Integration

Guaranteed set integration with Taylor methods

- (Moore,66) (Lohner,88) (Rihm,94) (Berz,98) (Nedialkov,99)

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, \mathbf{p}, t), \quad t_0 \leq t \leq t_N, \ \mathbf{x}(t_0) \in [\mathbf{x}_0], \ \mathbf{p} \in [\mathbf{p}]$$

a priori enclosure (*entrée* : $[\mathbf{x}_j]$, h, α ; sortie : $[\tilde{\mathbf{x}}_j]$)

1. Initialisation : $[\tilde{\mathbf{x}}_j] := [\mathbf{x}_j] + [0, h]\,\mathbf{f}([\mathbf{x}_j])$;
2. tant que $([\mathbf{x}_j] + [0, h]\,\mathbf{f}([\tilde{\mathbf{x}}_j]) \not\subset [\tilde{\mathbf{x}}_j])$

$$\begin{aligned} [\tilde{\mathbf{x}}_j] &:= [\tilde{\mathbf{x}}_j] + [-\alpha, \alpha]\,|[\tilde{\mathbf{x}}_j]| \\ h &:= h/2 \end{aligned}$$

fin

An Effective High-Order Interval Method for Validating Existence and Uniqueness of the Solution of an IVP for an ODE, Nedialko S. Nedialkov, Kenneth R. Jackson, and John D. Pryce, Reliable Computing 7(6) :449 - 465, 2001.

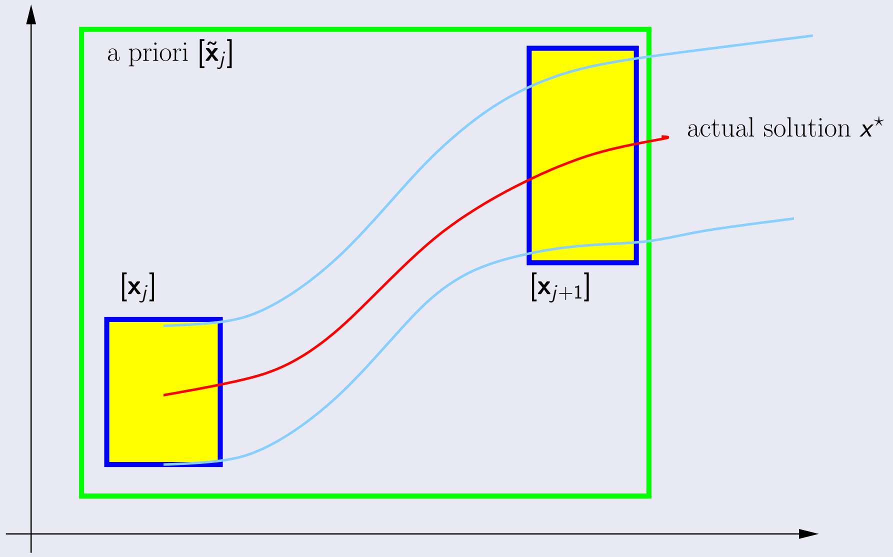
Nonlinear Set Integration

Guaranteed set integration with Taylor methods

- (Moore,66) (Lohner,88) (Rihm,94) (Berz,98) (Nedialkov,99)

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, \mathbf{p}, t), \quad t_0 \leq t \leq t_N, \ \mathbf{x}(t_0) \in [\mathbf{x}_0], \ \mathbf{p} \in [\mathbf{p}]$$

Time grid $\rightarrow \quad t_0 < t_1 < t_2 < \cdots < t_N$



- Compute tight enclosure $[\mathbf{x}_{j+1}] \ni \mathbf{x}(t_{j+1})$

$$[\mathbf{x}_{j+1}] = [\mathbf{x}_j] + \sum_{i=1}^{k-1} (t_{j+1} - t_j)^i \mathbf{f}^{[i]}([\mathbf{x}_j], [\mathbf{p}]) + (t_{j+1} - t_j)^k \mathbf{f}^{[k]}([\tilde{\mathbf{x}}_j], [\mathbf{p}])$$

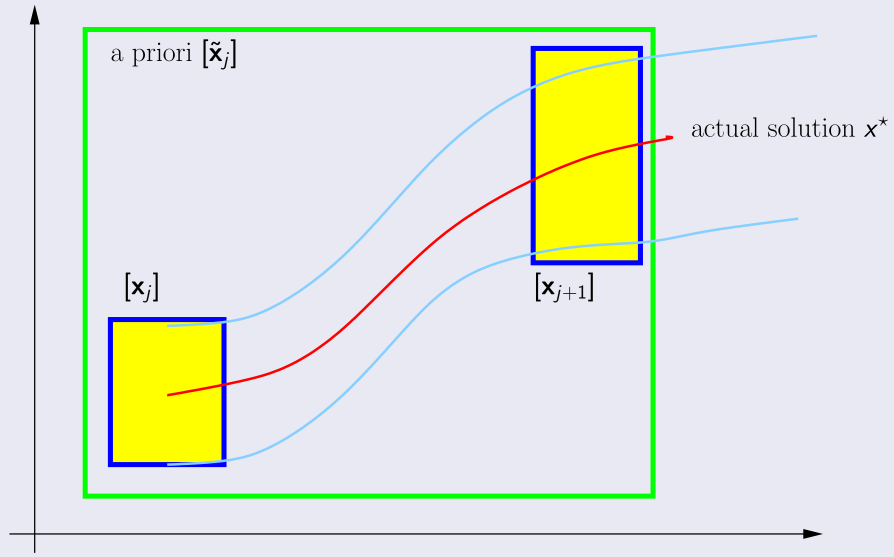
Nonlinear Set Integration

Guaranteed set integration with Taylor methods

- (Moore,66) (Lohner,88) (Rihm,94) (Berz,98) (Nedialkov,99)

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, \mathbf{p}, t), \quad t_0 \leq t \leq t_N, \ \mathbf{x}(t_0) \in [\mathbf{x}_0], \ \mathbf{p} \in [\mathbf{p}]$$

Time grid $\rightarrow \quad t_0 < t_1 < t_2 < \cdots < t_N$



- Analytical solution for $[\mathbf{x}](t), \ t \in [t_j, t_{j+1}]$

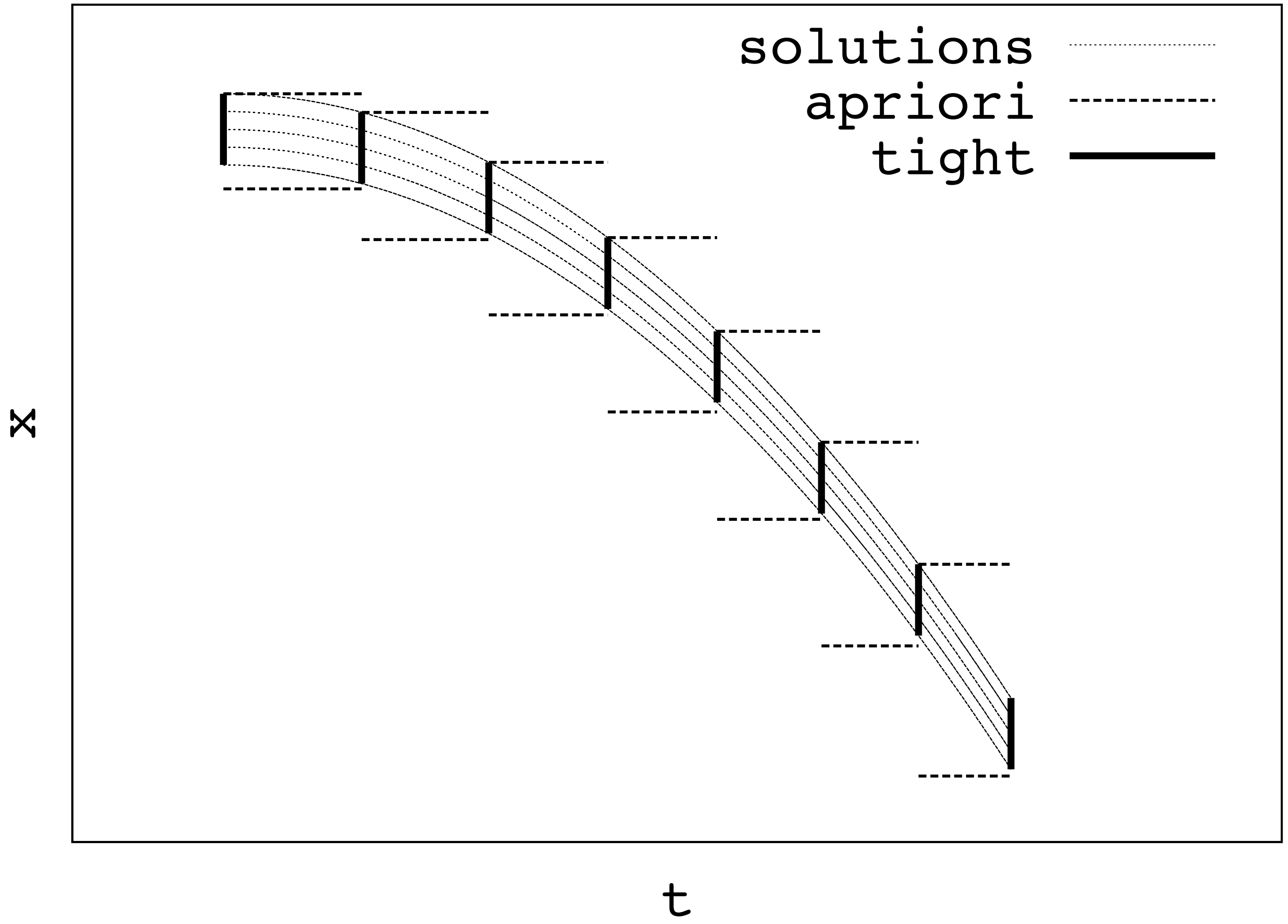
$$[\mathbf{x}](t) = [\mathbf{x}_j] + \sum_{i=1}^{k-1} (t - t_j)^i \mathbf{f}^{[i]}([\mathbf{x}_j], [\mathbf{p}]) + (t - t_j)^k \mathbf{f}^{[k]}([\tilde{\mathbf{x}}_j], [\mathbf{p}])$$

Nonlinear Set Integration

Guaranteed set integration with Taylor methods

- (Moore,66) (Lohner,88) (Rihm,94) (Berz,98) (Nedialkov,99)

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, \mathbf{p}, t), \quad t_0 \leq t \leq t_N, \ \mathbf{x}(t_0) \in [\mathbf{x}_0], \ \mathbf{p} \in [\mathbf{p}]$$



Nonlinear Set Integration

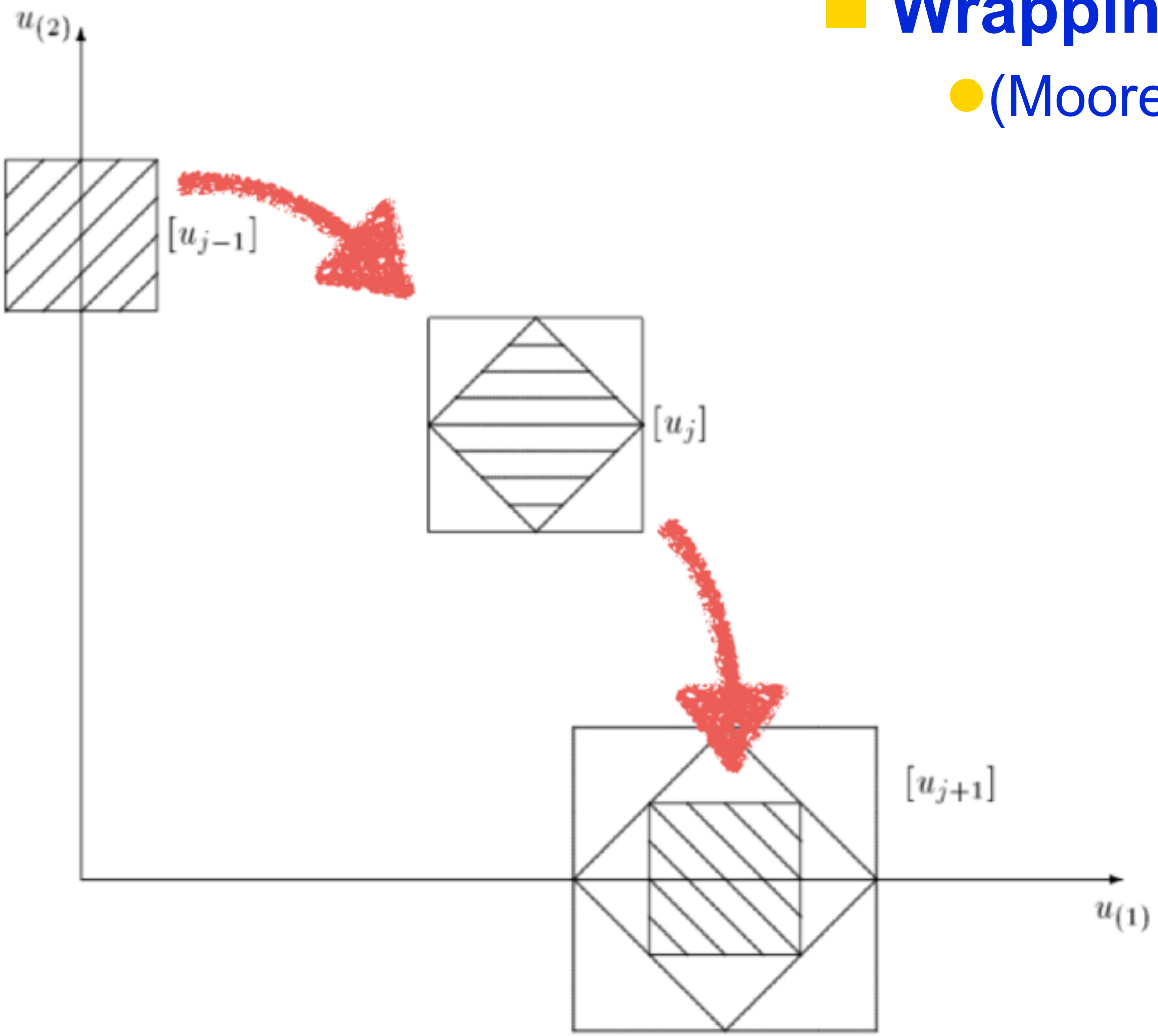
- **Guaranteed set integration with Taylor methods**
  - (Moore,66) (Lohner,88) (Rihm,94) (Berz,98) (Nedialkov,99)

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, \mathbf{p}, t), \quad t_0 \leq t \leq t_N, \ \mathbf{x}(t_0) \in [\mathbf{x}_0], \ \mathbf{p} \in [\mathbf{p}]$$

$$\begin{aligned}
\mathbf{f}^{[1]} &= \mathbf{x}^{(1)} = \mathbf{f} \\
\mathbf{f}^{[2]} &= \tfrac{1}{2}\mathbf{x}^{(2)} = \tfrac{1}{2}\tfrac{\mathrm{d}\mathbf{f}}{\mathrm{d}\mathbf{x}}\mathbf{f} \\
\mathbf{f}^{[i]} &= \tfrac{1}{i!}\mathbf{x}^{(i)} = \tfrac{1}{i}\tfrac{\mathrm{d}\mathbf{f}^{[i-1]}}{\mathrm{d}\mathbf{x}}\mathbf{f}, \ \ i \geqslant 2
\end{aligned}$$

Nonlinear Set Integration

- **Wrapping effect**
  - (Moore,66)



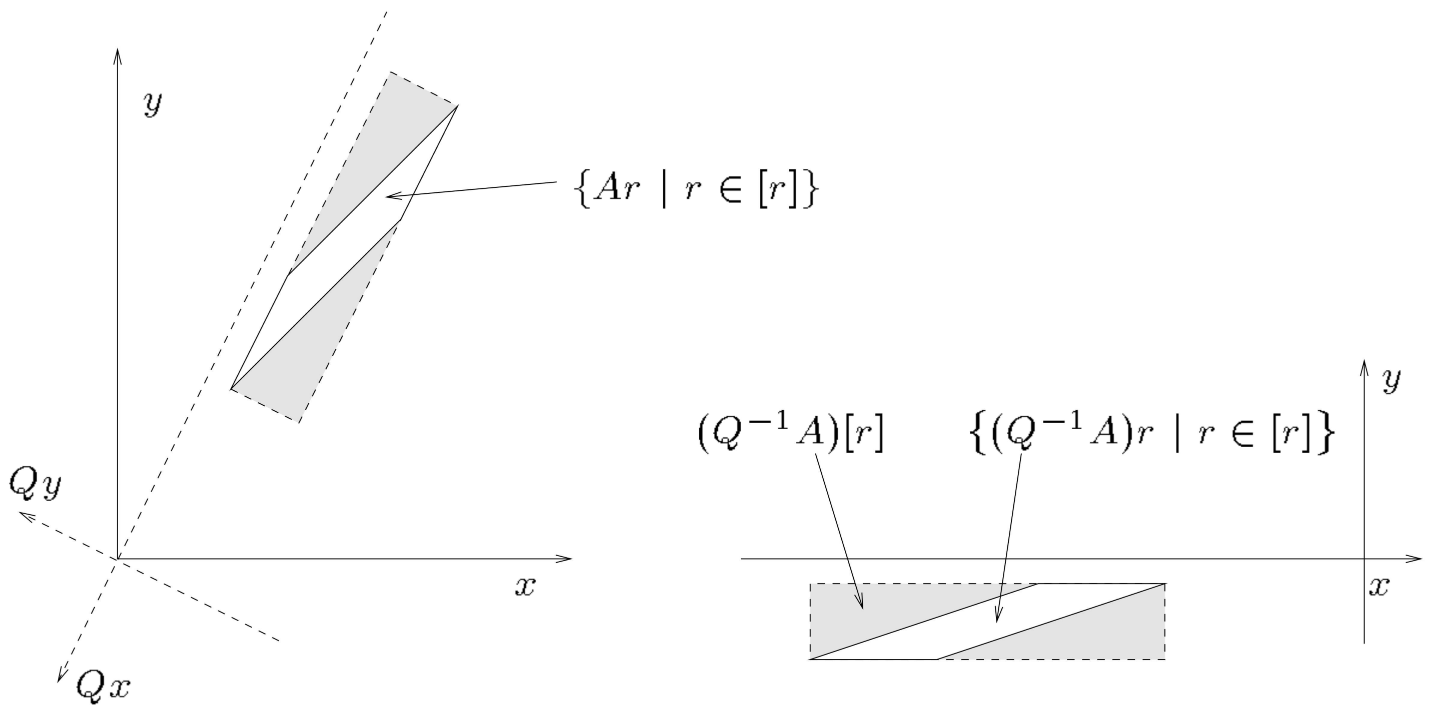
Nonlinear Set Integration

- **Guaranteed set integration with Taylor methods**
  - (Moore,66) (Lohner,88) (Rihm,94) (Berz,98) (Nedialkov,99)

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, \mathbf{p}, t), \quad t_0 \leq t \leq t_N, \ \mathbf{x}(t_0) \in [\mathbf{x}_0], \ \mathbf{p} \in [\mathbf{p}]$$

Mean-value approach

$$[\mathbf{x}](t) \in \left\{ \mathbf{v}(t) + \mathbf{A}(t)\mathbf{r}(t) \quad | \quad \mathbf{v}(t) \in [\mathbf{v}](t), \ \mathbf{r}(t) \in [\mathbf{r}](t) \right\}.$$



Nonlinear Set Integration

- **Guaranteed set integration with Taylor methods**
  - (Moore,66) (Lohner,88) (Rihm,94) (Berz,98) (Nedialkov,99)

- **Complexity**
  - *Work per step is of polynomial complexity*
    - Computing Taylor coefficients → o($\boldsymbol{k}^2$)
    - Linear algebra → o($\boldsymbol{n}^3$)

- **In practice :** Obtaining Taylor coefficients ...
  - **FADBAD++** (www.fadbad.com)

    Flexible Automatic differentiation using templates and operator overloading in C++

VNODE-LP

An Interval Solver for Initial Value Problems in Ordinary Differential Equations

Ned Nedialkov
nedialk@mcmaster.ca

VNODE-LP is a C++ package for computing bounds on solutions in IVPs for ODEs. In contrast to traditional ODE solvers, which compute approximate solutions, this solver tries to prove that a unique solution to a problem exists and then computes bounds that contain this solution. Such bounds can be used to help prove a theoretical result, check if a solution satisfies a condition in a safety-critical calculation, or simply to verify the results produced by a traditional ODE solver.

This package is a successor of the VNODE package of N. Nedialkov. A distinctive feature of the present solver is that it is developed entirely using Literate Programming. As a result, the correctness of VNODE-LP's implementation can be examined easier than the correctness of VNODE: the theory, documentation, and source code are produced from the same CWEB files.

download

Nonlinear Set Integration

Comparison theorems for differential inequalities

- Müller's existence theorem (1936)

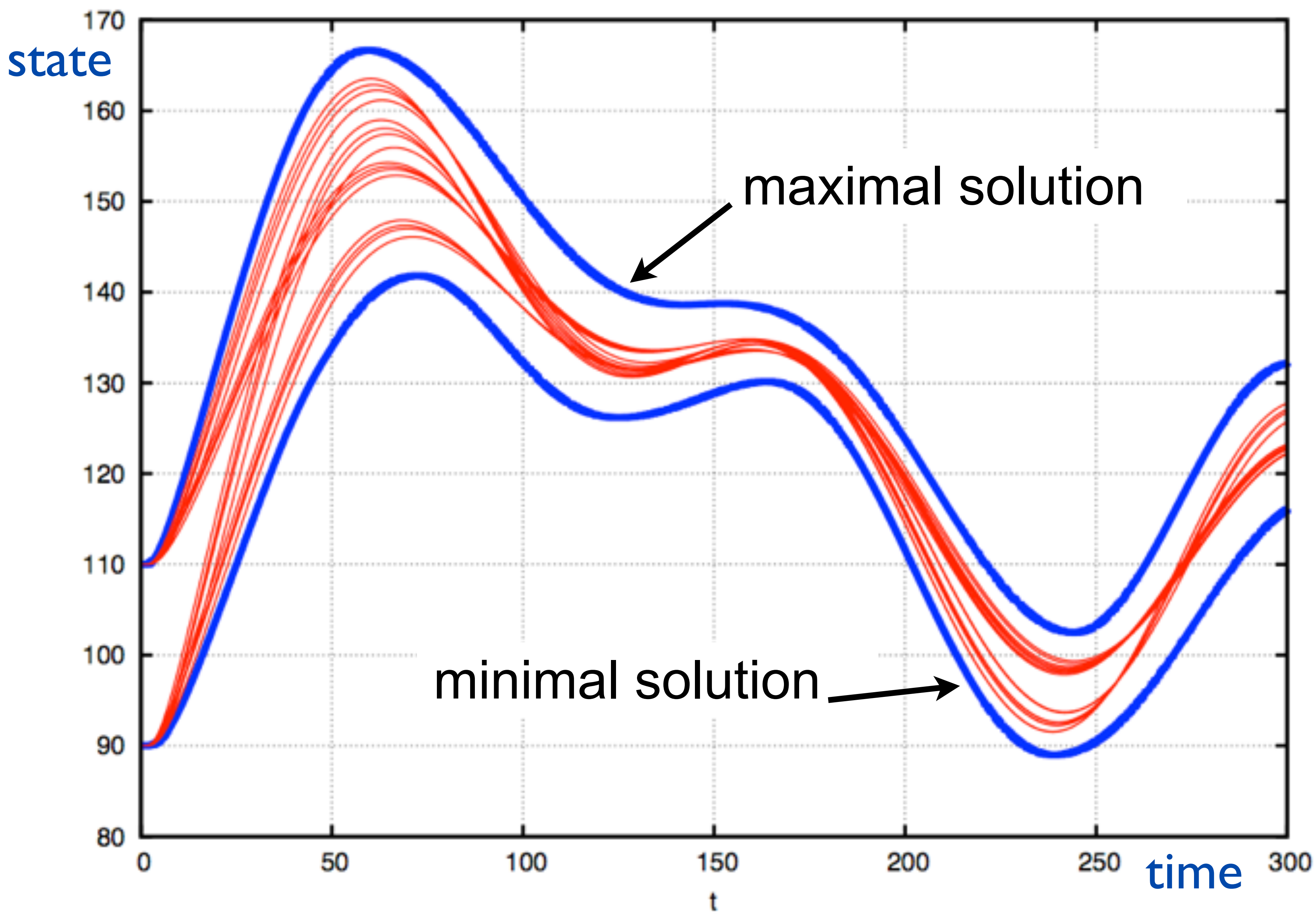
$$\text{If} \left\{ \begin{array}{c} \forall t \in [t_0,\ t_N],\ \forall \mathbf{x} \in \mathbb{D},\ \forall \mathbf{p} \in \mathbb{P} \\ \forall i\ \min_{x_i=\omega_i} f_i(\mathbf{x},\mathbf{p},t) \geq D^{\pm}\omega_i(t) \\ \forall i\ \max_{x_i=\Omega_i} f_i(\mathbf{x},\mathbf{p},t) \leq D^{\pm}\Omega_i(t) \\ \omega(t_0) \leq \mathbf{x}(t_0) \leq \Omega(t_0) \end{array} \right. \Rightarrow \left\{ \begin{array}{c} \text{solution exists} \\ \\ \forall t \in [t_0,\ t_N], \\ \omega(t) \leq \mathbf{x}(t) \leq \Omega(t) \end{array} \right.$$

- **Bracketing systems**
  - (Ramdani, et al., IEEE Trans. Automatic Control 2009)

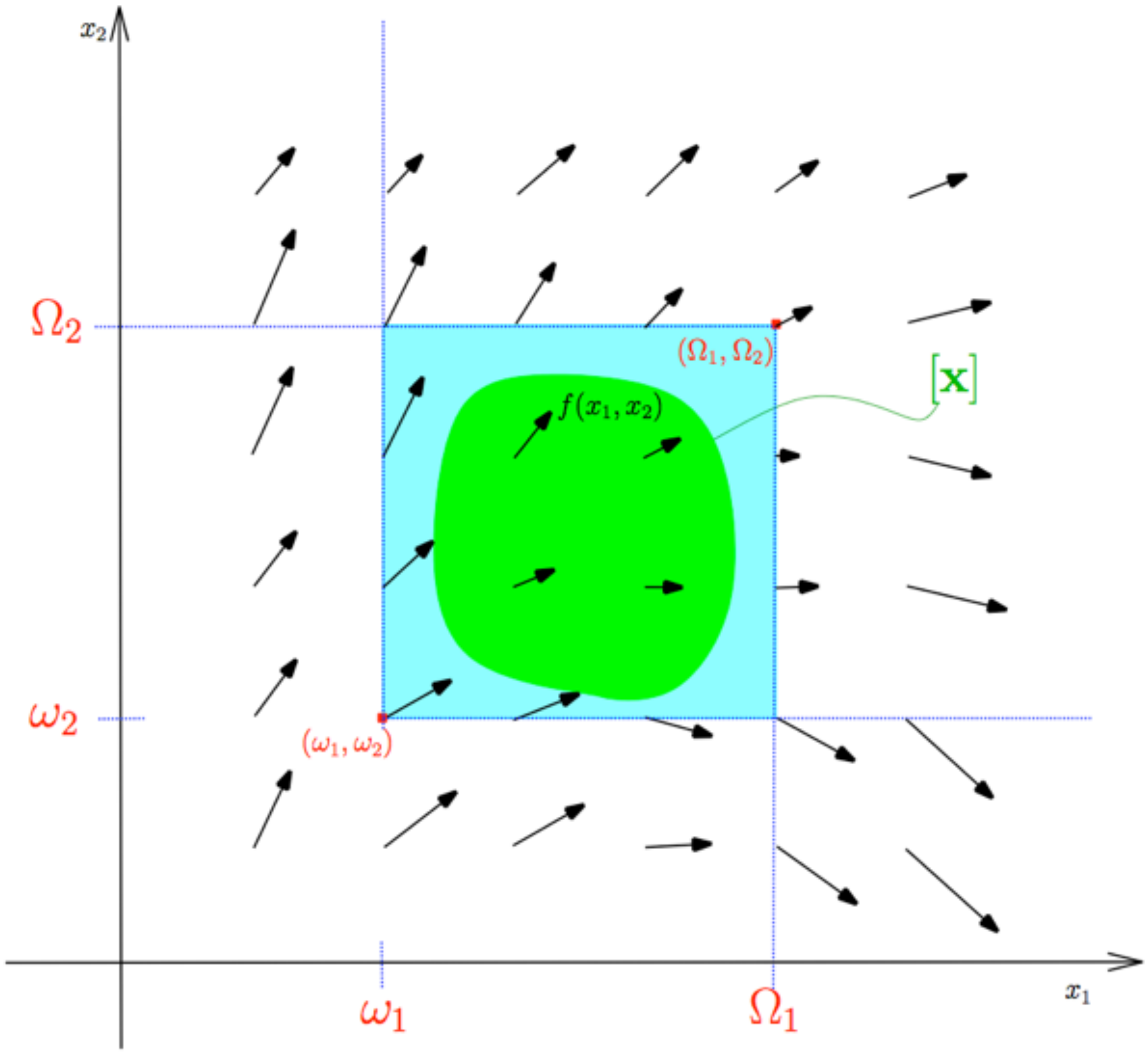
Nonlinear Set Integration

Comparison theorems for differential inequalities

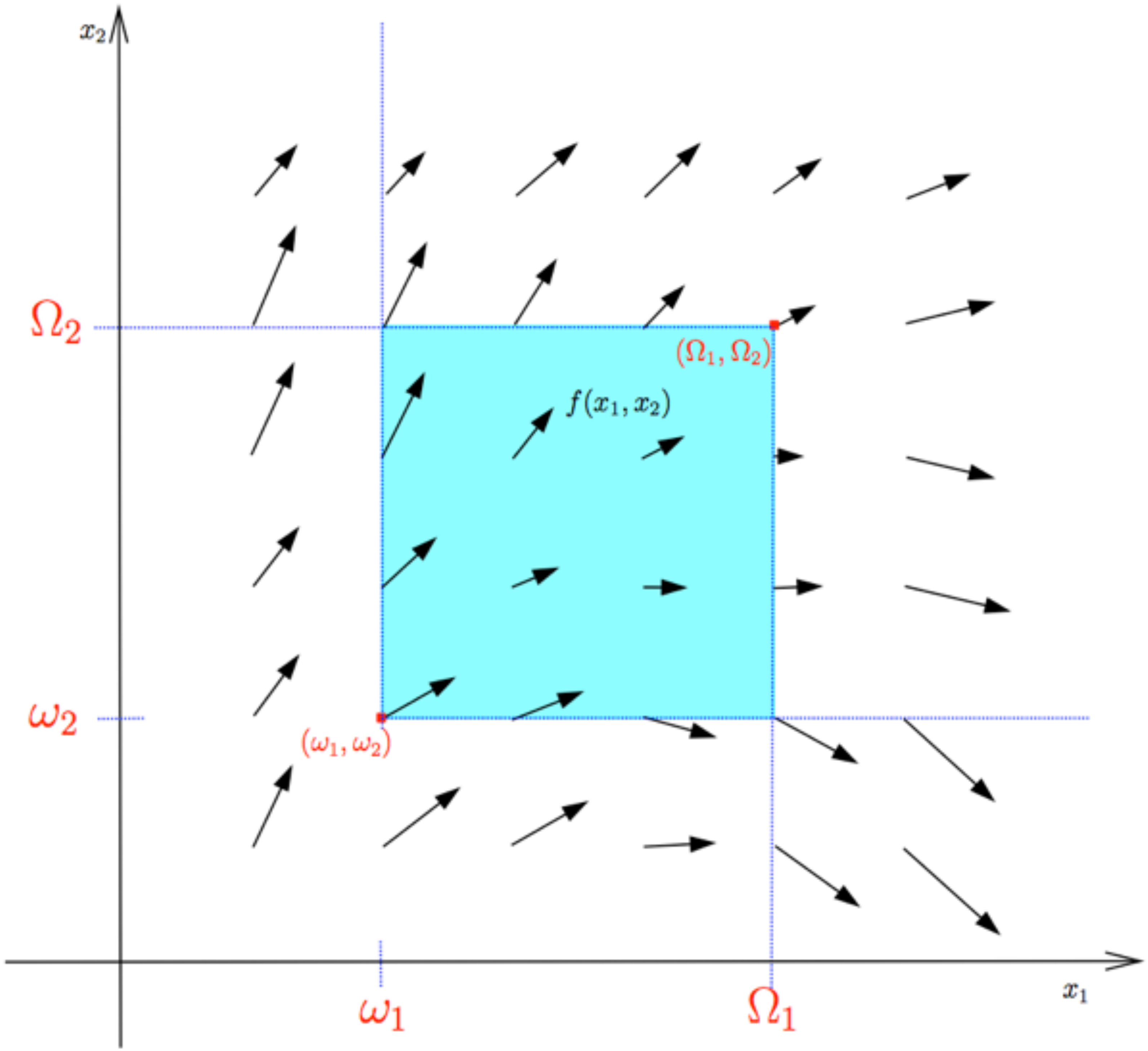
Bracketing systems



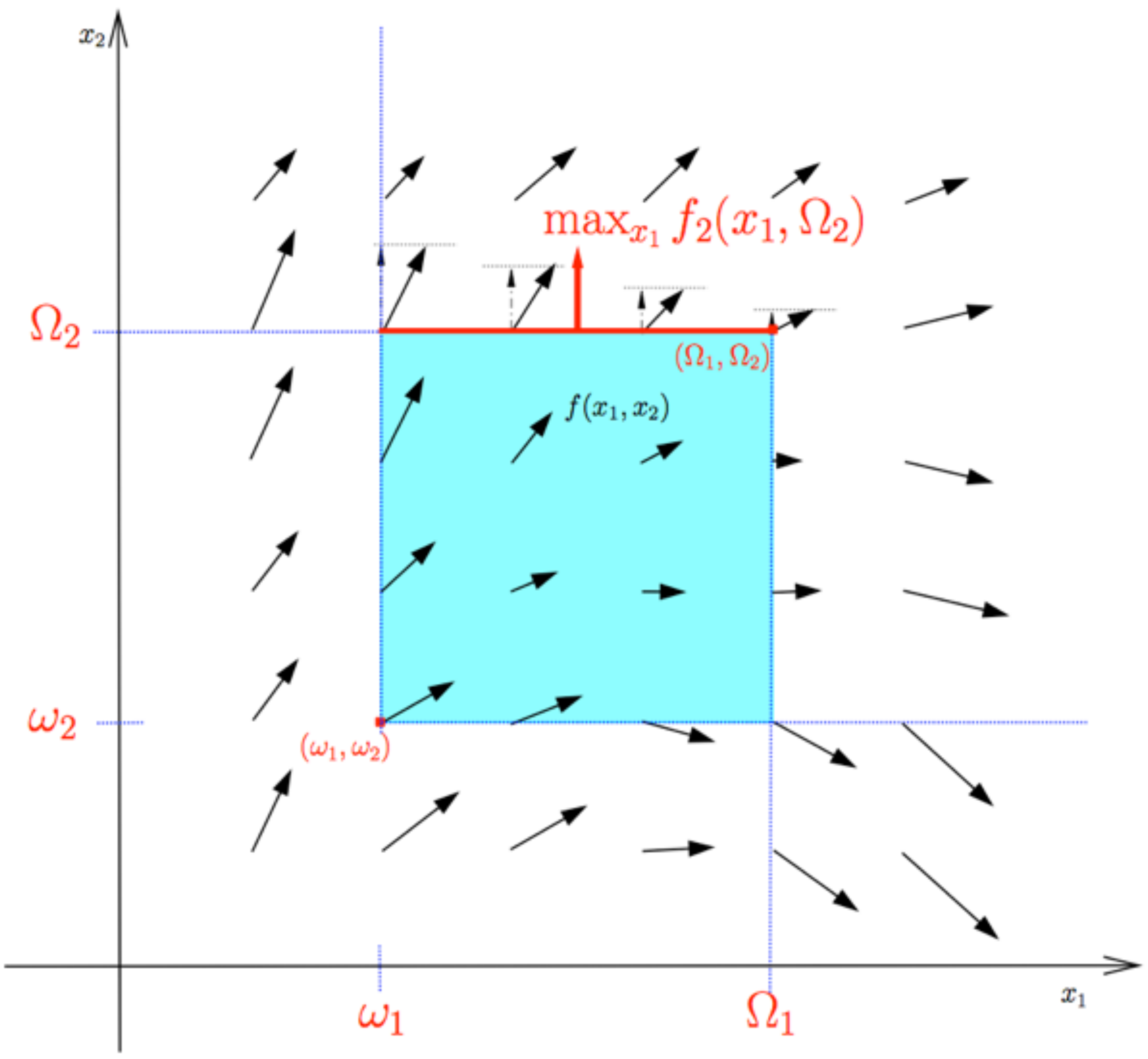
Müller's theorem



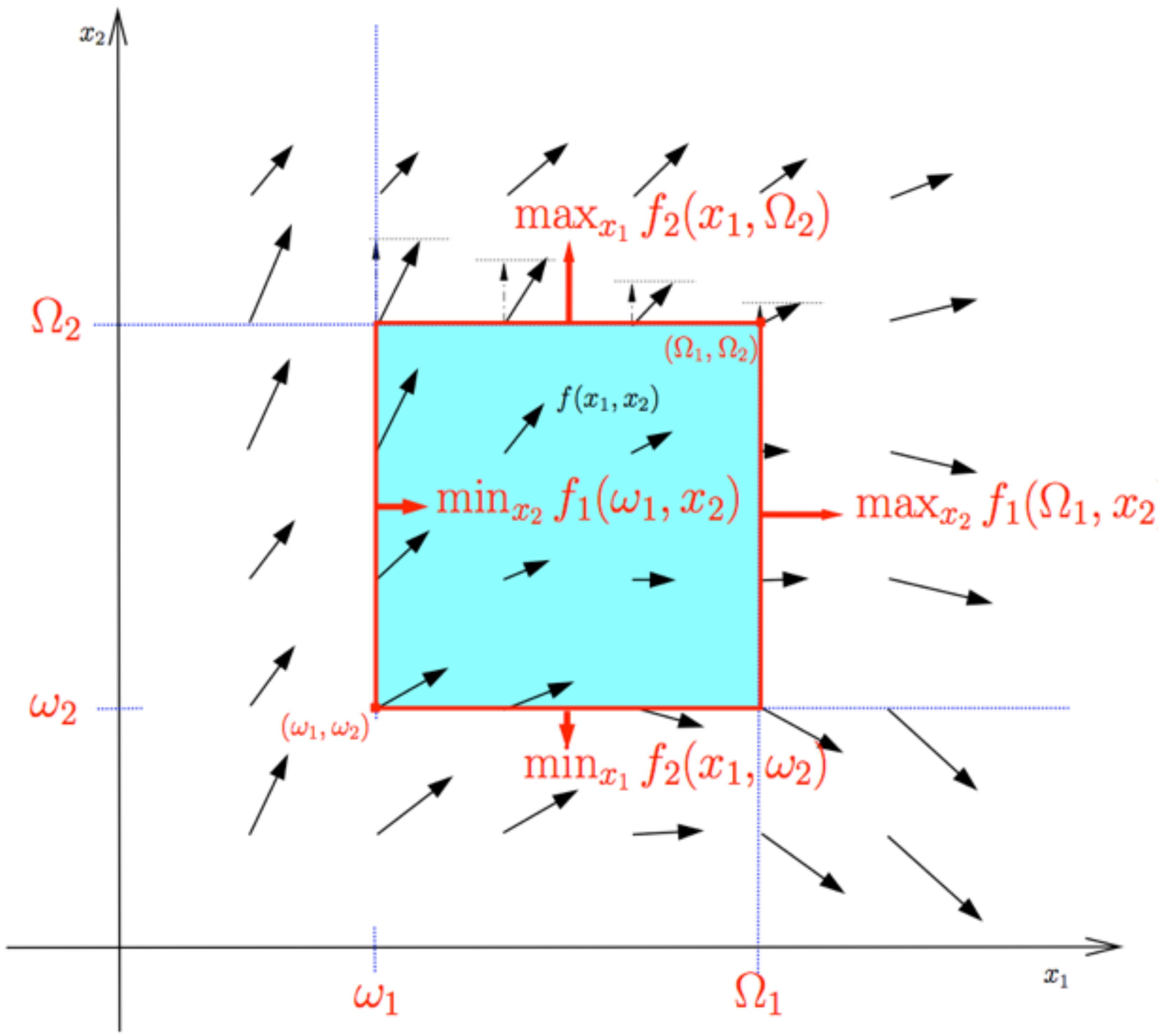
Müller's theorem



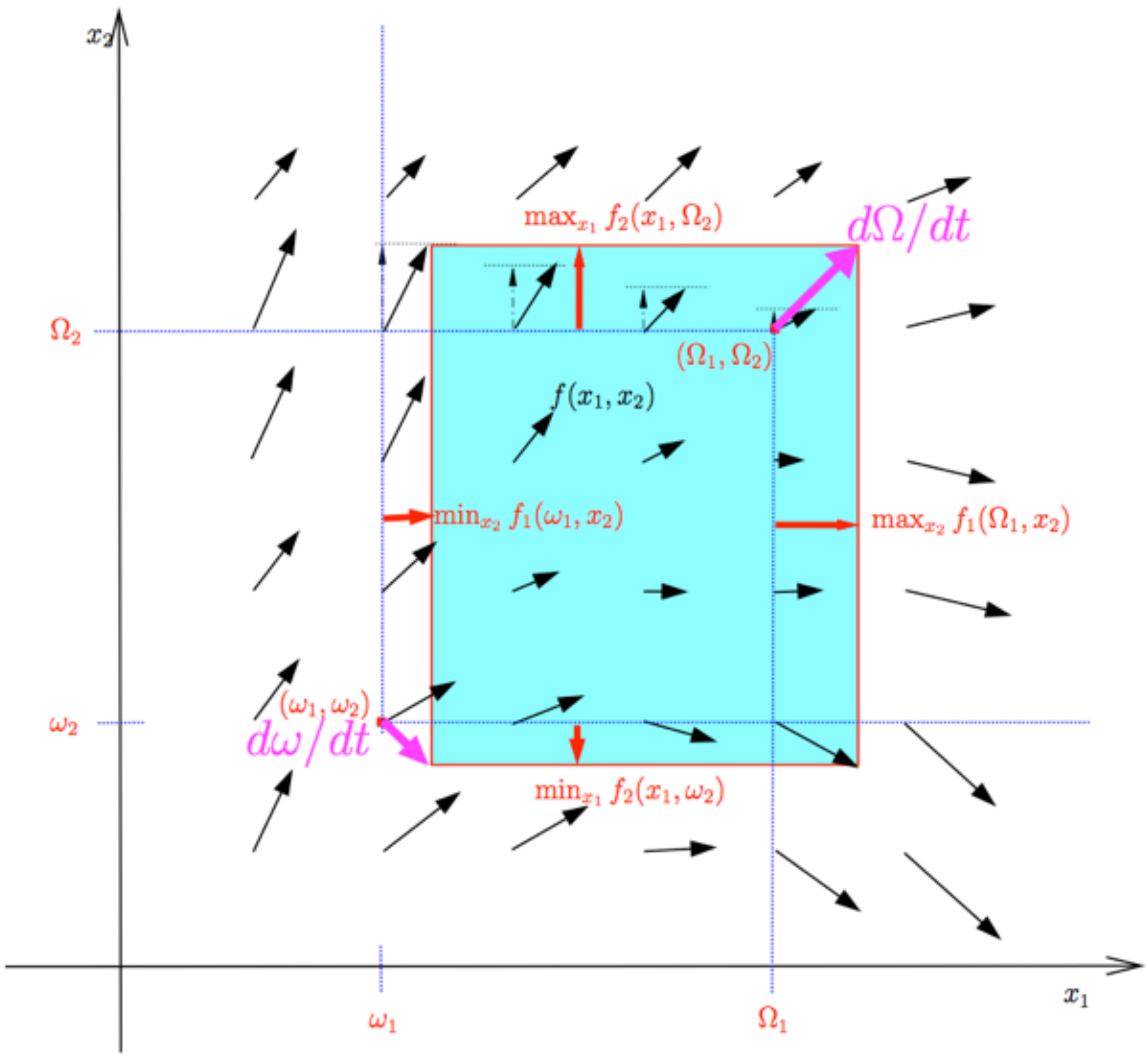
Müller's theorem



Müller's theorem



Müller's theorem



Nonlinear Set Integration

Bracketing systems

- Dynamics of ...

$$\begin{cases} \dot{x}_1 = f_1(x_1, x_2, p, t), & x_1(t_0) \in [\underline{x}_{1,0}, \overline{x}_{1,0}] \subset \mathbb{R}, \quad p \in [\underline{p}, \overline{p}] \quad t \geq t_0 \\ \dot{x}_2 = f_2(x_1, x_2, p, t), & x_2(t_0) \in [\underline{x}_{2,0}, \overline{x}_{2,0}] \subset \mathbb{R}, \end{cases}$$

If $\forall t \geq t_0,\ \forall \mathbf{x}(t) \in [\omega(t), \Omega(t)] \subset \mathbb{R}^2,\ \forall p \in [\underline{p}, \overline{p}]$,

$$\frac{\partial f_1}{\partial x_2} > 0 \ \wedge\ \frac{\partial f_1}{\partial p} > 0$$

then $\quad f_1(\omega_1, \omega_2, \underline{p}) \leq f_1(\omega_1, x_2, p, t) \quad$ and $\quad f_1(\Omega_1, x_2, p, t) \leq f_1(\Omega_1, \Omega_2, \overline{p})$

$$\dot{\omega}_1(t) \equiv f_1(\omega_1, \omega_2, \underline{p}) \quad \text{and} \quad f_1(\Omega_1, \Omega_2, \overline{p}) \equiv \dot{\Omega}_1(t)$$

Nonlinear Set Integration

Bracketing systems

- Dynamics of ...

$$\begin{cases} \dot{x}_1 = f_1(x_1, x_2, p, t), & x_1(t_0) \in [\underline{x}_{1,0}, \overline{x}_{1,0}] \subset \mathbb{R}, \quad p \in [\underline{p}, \overline{p}] \quad t \geq t_0 \\ \dot{x}_2 = f_2(x_1, x_2, p, t), & x_2(t_0) \in [\underline{x}_{2,0}, \overline{x}_{2,0}] \subset \mathbb{R}, \end{cases}$$

If $\forall t \geq t_0$, $\forall \mathbf{x}(t) \in [\omega(t), \Omega(t)] \subset \mathbb{R}^2$, $\forall p \in [\underline{p}, \overline{p}]$,

$$\frac{\partial f_1}{\partial x_2} > 0 \wedge \frac{\partial f_1}{\partial p} > 0$$

then $\quad f_1(\omega_1, \omega_2, \underline{p}) \leq f_1(\omega_1, x_2, p, t) \quad$ and $\quad f_1(\Omega_1, x_2, p, t) \leq f_1(\Omega_1, \Omega_2, \overline{p})$

$$\dot{\omega}_1(t) \equiv f_1(\omega_1, \omega_2, \underline{p}) \quad \text{and} \quad f_1(\Omega_1, \Omega_2, \overline{p}) \equiv \dot{\Omega}_1(t)$$

Nonlinear Set Integration

UNIVERSITE D'ORLEANS

Comparison theorems for differential inequalities

- Müller's existence theorem (1936)

$$\text{If } \begin{cases} \forall t \in [t_0,\ t_N], \forall \mathbf{x} \in \mathbb{D}, \forall \mathbf{p} \in \mathbb{P} \\ \forall i \ \min_{x_i=\omega_i} f_i(\mathbf{x},\mathbf{p},t) \geq D^{\pm}\omega_i(t) \\ \forall i \ \max_{x_i=\Omega_i} f_i(\mathbf{x},\mathbf{p},t) \leq D^{\pm}\Omega_i(t) \\ \omega(t_0) \leq \mathbf{x}(t_0) \leq \Omega(t_0) \end{cases} \Rightarrow \begin{cases} \text{solution exists} \\ \\ \forall t \in [t_0,\ t_N], \\ \omega(t) \leq \mathbf{x}(t) \leq \Omega(t) \end{cases}$$

- **Bracketing systems : coupled EDOs**

$$\Rightarrow \begin{cases} \dot{\omega}(t) = \underline{f}(\omega, \mathbf{\Omega}, \underline{\mathbf{p}}, \overline{\mathbf{p}}, t), & \omega(t_0) = \underline{\mathbf{x}}_0 \\ \dot{\mathbf{\Omega}}(t) = \overline{f}(\omega, \mathbf{\Omega}, \underline{\mathbf{p}}, \overline{\mathbf{p}}, t), & \mathbf{\Omega}(t_0) = \overline{\mathbf{x}}_0 \end{cases}$$

Nonlinear Set Integration

- **Bracketing systems**
  - Example : Mitogen- Activated Protein Kinase (Sontag, 2005)

Nonlinear Set Integration

Bracketing systems

- Example : Mitogen- Activated Protein Kinase (Sontag, 2005)

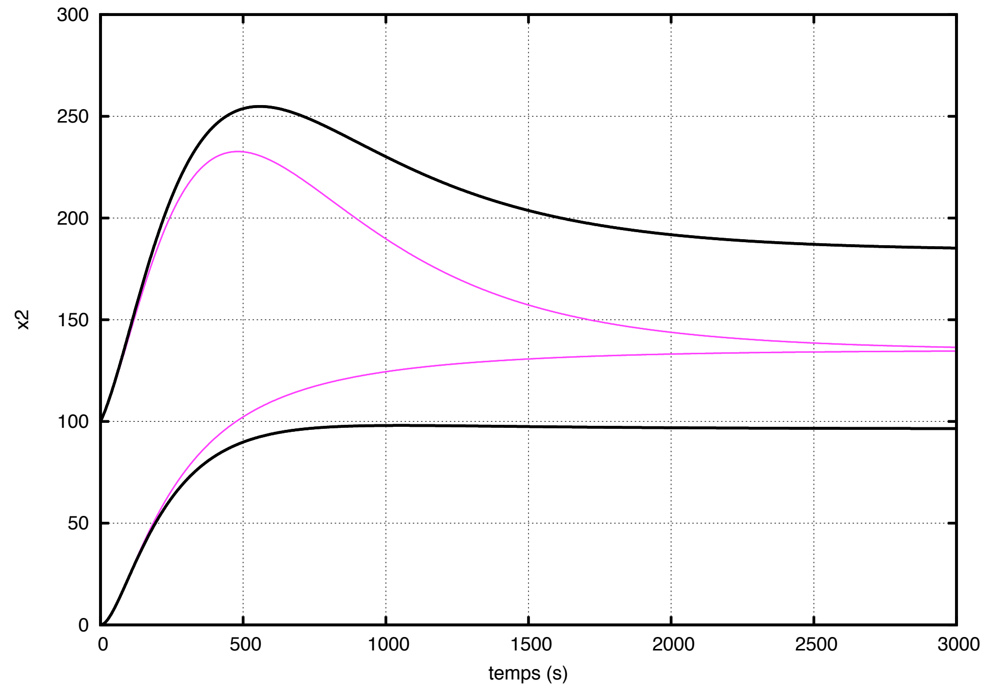
$$
\begin{cases}
\dot{x}_1 = -\frac{v_2 x_1}{k_2 + x_1} + v_0 u + v_1 \\
\dot{x}_2 = \frac{v_6 (y_{tot} - x_2 - x_3)}{k_6 + (y_{tot} - x_2 - x_3)} - \frac{v_3 x_1 x_2}{k_3 + x_2} \\
\dot{x}_3 = \frac{v_4 x_1 (y_{tot} - x_2 - x_3)}{k_4 + (y_{tot} - x_2 - x_3)} - \frac{v_5 x_3}{k_5 + x_3} \\
\dot{x}_4 = \frac{v_{10} (z_{tot} - x_4 - x_5)}{k_{10} + (z_{tot} - x_4 - x_5)} - \frac{v_7 x_3 x_4}{k_7 + x_4} \\
\dot{x}_5 = \frac{v_8 x_3 (z_{tot} - x_4 - x_5)}{k_8 + (z_{tot} - x_4 - x_5)} - \frac{v_9 x_5}{k_9 + x_5} \\
u = g x_5
\end{cases}
$$

Nonlinear Set Integration

Bracketing systems

- Example : Mitogen- Activated Protein Kinase (Sontag, 2005)

$$
\left\{
\begin{array}{lcl}
\dot{\underline{x}}_1 & = & -\dfrac{\overline{v}_2 \underline{x}_1}{\underline{k}_2 + \underline{x}_1} + \underline{v}_0 \underline{u} + \underline{v}_1 \\
\dot{\underline{x}}_2 & = & \dfrac{\underline{v}_6 (\underline{y}_{tot} - \underline{x}_2 - \overline{x}_3)}{\overline{k}_6 + (\underline{y}_{tot} - \underline{x}_2 - \overline{x}_3)} - \dfrac{\overline{v}_3 \overline{x}_1 \underline{x}_2}{\underline{k}_3 + \underline{x}_2} \\
\dot{\underline{x}}_3 & = & \dfrac{\underline{v}_4 \underline{x}_1 (\underline{y}_{tot} - \overline{x}_2 - \underline{x}_3)}{\overline{k}_4 + (\underline{y}_{tot} - \overline{x}_2 - \underline{x}_3)} - \dfrac{\overline{v}_5 \underline{x}_3}{\underline{k}_5 + \underline{x}_3} \\
\dot{\underline{x}}_4 & = & \dfrac{\underline{v}_{10} (\underline{z}_{tot} - \underline{x}_4 - \overline{x}_5)}{\overline{k}_{10} + (\underline{z}_{tot} - \underline{x}_4 - \overline{x}_5)} - \dfrac{\overline{v}_7 \overline{x}_3 \underline{x}_4}{\underline{k}_7 + \underline{x}_4} \\
\dot{\underline{x}}_5 & = & \dfrac{\underline{v}_8 \underline{x}_3 (\underline{z}_{tot} - \overline{x}_4 - \underline{x}_5)}{\overline{k}_8 + (\underline{z}_{tot} - \overline{x}_4 - \underline{x}_5)} - \dfrac{\overline{v}_9 \underline{x}_5}{\underline{k}_9 + \underline{x}_5} \\
\dot{\overline{x}}_1 & = & -\dfrac{\underline{v}_2 \overline{x}_1}{\overline{k}_2 + \overline{x}_1} + \overline{v}_0 \overline{u} + \overline{v}_1 \\
\dot{\overline{x}}_2 & = & \dfrac{\overline{v}_6 (\overline{y}_{tot} - \overline{x}_2 - \underline{x}_3)}{\underline{k}_6 + (\overline{y}_{tot} - \overline{x}_2 - \underline{x}_3)} - \dfrac{\underline{v}_3 \underline{x}_1 \overline{x}_2}{\overline{k}_3 + \overline{x}_2} \\
\dot{\overline{x}}_3 & = & \dfrac{\overline{v}_4 \overline{x}_1 (\overline{y}_{tot} - \underline{x}_2 - \overline{x}_3)}{\underline{k}_4 + (\overline{y}_{tot} - \underline{x}_2 - \overline{x}_3)} - \dfrac{\underline{v}_5 \overline{x}_3}{\overline{k}_5 + \overline{x}_3} \\
\dot{\overline{x}}_4 & = & \dfrac{\overline{v}_{10} (\overline{z}_{tot} - \overline{x}_4 - \underline{x}_5)}{\underline{k}_{10} + (\overline{z}_{tot} - \overline{x}_4 - \underline{x}_5)} - \dfrac{\underline{v}_7 \underline{x}_3 \overline{x}_4}{\overline{k}_7 + \overline{x}_4} \\
\dot{\overline{x}}_5 & = & \dfrac{\overline{v}_8 \overline{x}_3 (\overline{z}_{tot} - \underline{x}_4 - \overline{x}_5)}{\underline{k}_8 + (\overline{z}_{tot} - \underline{x}_4 - \overline{x}_5)} - \dfrac{\underline{v}_9 \overline{x}_5}{\overline{k}_9 + \overline{x}_5} \\
\underline{u} & = & g \underline{x}_5 \\
\overline{u} & = & g \overline{x}_5
\end{array}
\right.
$$



Nonlinear Set Integration

- **Monotone order-preserving systems**
  - Müller, Kamke, Krasnoselskii, Hirsch, Smith, Angeli and Sontag.
  - Preserve ordering on initial conditions.

$$\mathbf{x}(t_0) \prec \mathbf{y}(t_0) \Rightarrow \forall \mathbf{t} \geqslant \mathbf{t_0} \quad \mathbf{x}(t) \prec \mathbf{y}(t) \qquad \prec \in \{<, \leq, \geq, >\}$$

Nonlinear Set Integration

Monotone order-preserving systems

- **Graphical test : monotone wrt orthant cones (Kunze & Siegel, 1999)**

$$\text{if } \exists\, \mathbf{D} = diag[(-1)^{\varepsilon_1}, ..., [(-1)^{\varepsilon_n}], \varepsilon_i \in \{0, 1\}$$

s.t $\mathbf{x}(t, \mathbf{x}_0, t_0)$ and $\mathbf{y}(t, \mathbf{y}_0, t_0)$ satisfy

$$\mathbf{Dy}_0 \geq \mathbf{Dx}_0 \Rightarrow \mathbf{Dy}(t, \mathbf{y}_0, t_0) \geq \mathbf{Dx}(t, \mathbf{x}_0, t_0)\ \forall t \geq t_0.$$

Nonlinear Set Integration

Monotone order-preserving systems

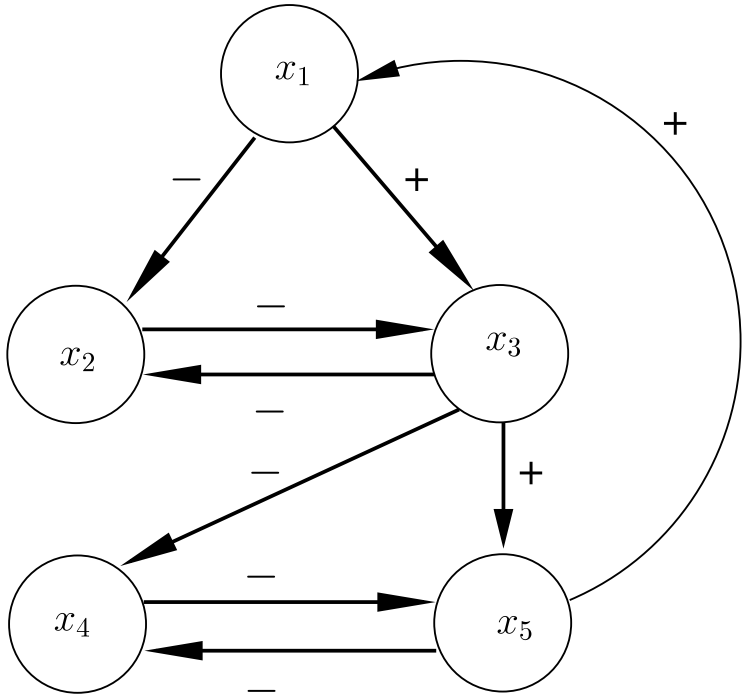
- **Graphical test : monotone wrt orthant cones (Kunze & Siegel, 1999)**

$$\text{if } \exists\, \mathbf{D} = diag[(-1)^{\varepsilon_1}, \ldots, [(-1)^{\varepsilon_n}], \varepsilon_i \in \{0,1\}$$

s.t $\mathbf{x}(t, \mathbf{x}_0, t_0)$ and $\mathbf{y}(t, \mathbf{y}_0, t_0)$ satisfy

$$\mathbf{D}\mathbf{y}_0 \geq \mathbf{D}\mathbf{x}_0 \Rightarrow \mathbf{D}\mathbf{y}(t, \mathbf{y}_0, t_0) \geq \mathbf{D}\mathbf{x}(t, \mathbf{x}_0, t_0)\ \forall t \geq t_0.$$

$$\begin{cases} \dot{x}_1 = -(v_2 x_1)/(k_2 + x_1) + v_0 g x_5 + v_1 \\ \dot{x}_2 = (v_6(y_{tot} - x_2 - x_3))/(k_6 + (y_{tot} - x_2 - x_3)) - (v_3 x_1 x_2)/(k_3 + x_2) \\ \dot{x}_3 = (v_4 x_1 (y_{tot} - x_2 - x_3))/(k_4 + (y_{tot} - x_2 - x_3)) - (v_5 x_3)/(k_5 + x_3) \\ \dot{x}_4 = (v_{10}(z_{tot} - x_4 - x_5))/(k_{10} + (z_{tot} - x_4 - x_5)) - (v_7 x_3 x_4)/(k_7 + x_4) \\ \dot{x}_5 = (v_8 x_3 (z_{tot} - x_4 - x_5))/(k_8 + (z_{tot} - x_4 - x_5)) - (v_9 x_5)/(k_9 + x_5) \end{cases}$$



Nonlinear Set Integration

IOLAVABE: iSAT-ODE Layer Around VNODE-LP and Bracketing Enclosures

About

The IOLAVABE library encapsulates the part of the iSAT-ODE tool that handles the generation of ODE enclosures using VNODE-LP and bracketing systems.

IOLAVABE is made available here solely for scientific research.

Detailed licensing information can be found in the LICENSE file inside the source code archive. IOLAVABE depends on and the archive file contains modified versions of VNODE-LP (itself including a copy of FADBAD++) and of filib++. The unmodified versions can be found in the bundled archive as well. Please note the licensing information shipped with these and all indirectly or directly used libraries as well (you will find pointers to the respective terms of use in the INSTALL or LICENSE file or in your system's package management system).

Installation instructions are to be found in the INSTALL file, and a list of changes with respect to earlier releases can be found in the changelog file.

Contact the author: Andreas Eggers

https://seshome.informatik.uni-oldenburg.de/eggers/iolavabe.php

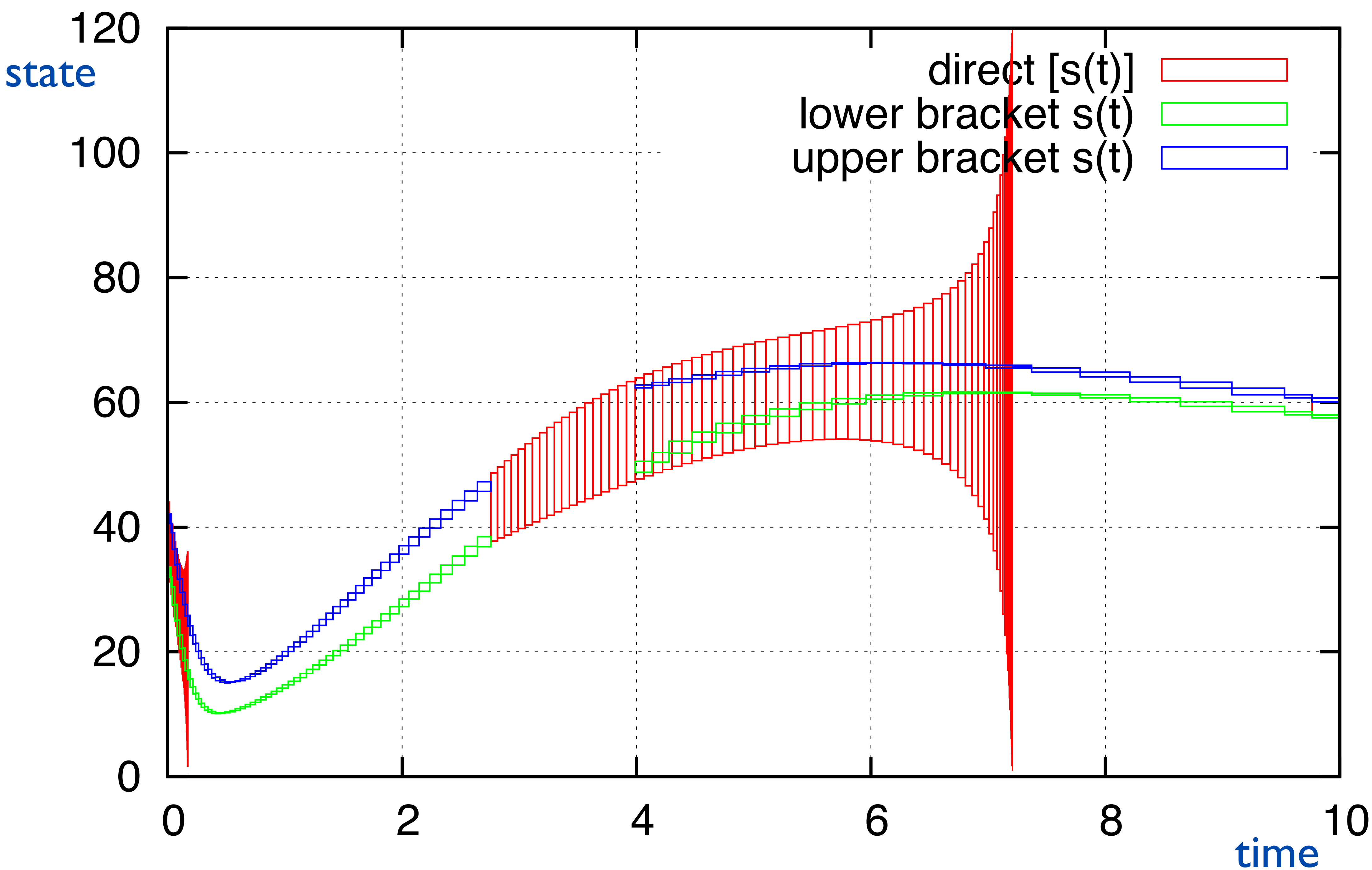
iSAT-ODE Layer

- generates **on-the-fly** hybrid bracketing systems, i.e. tries to re-start bracketing system when monotonicity changes

- uses subordinate local optimization to compute signs of partial derivatives on subranges to improve bracketing

iSAT-ODE Layer

- **Typical results:** Taylor methods vs Bracketing systems

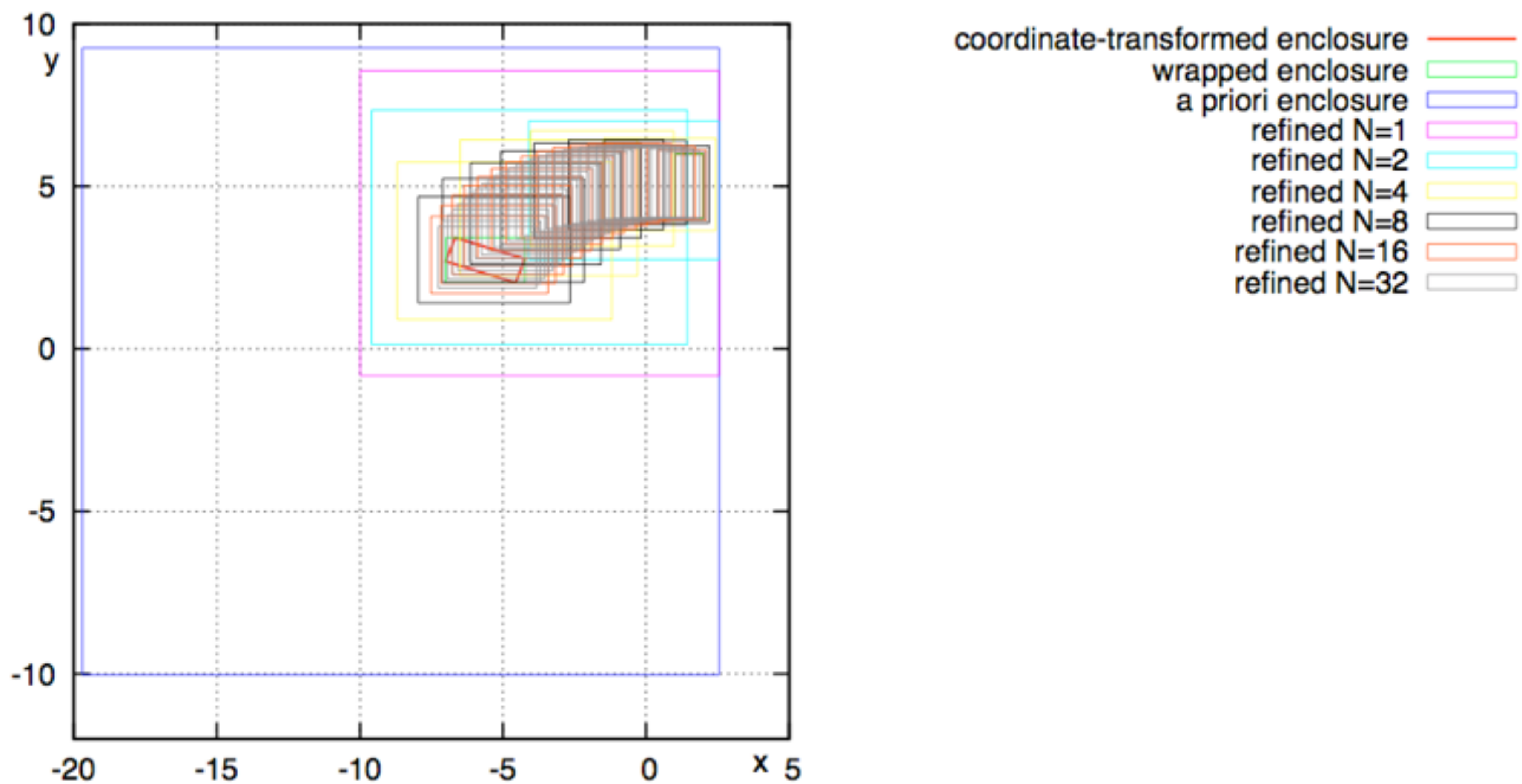


iSAT-ODE Layer

- harmonizes bracketing and direct enclosures, i.e. synchronizes time step,
- often intersects enclosures and reinitializes methods

iSAT-ODE Layer

- stores Taylor coefficients to recompute «refined» enclosures at intermediate steps.



$$\dot{x} = -y, \quad \dot{y} = 0.6 \cdot x, \quad x_0 \in [1,2], \quad y_0 \in [4,6], \quad t_1 = 1.6$$

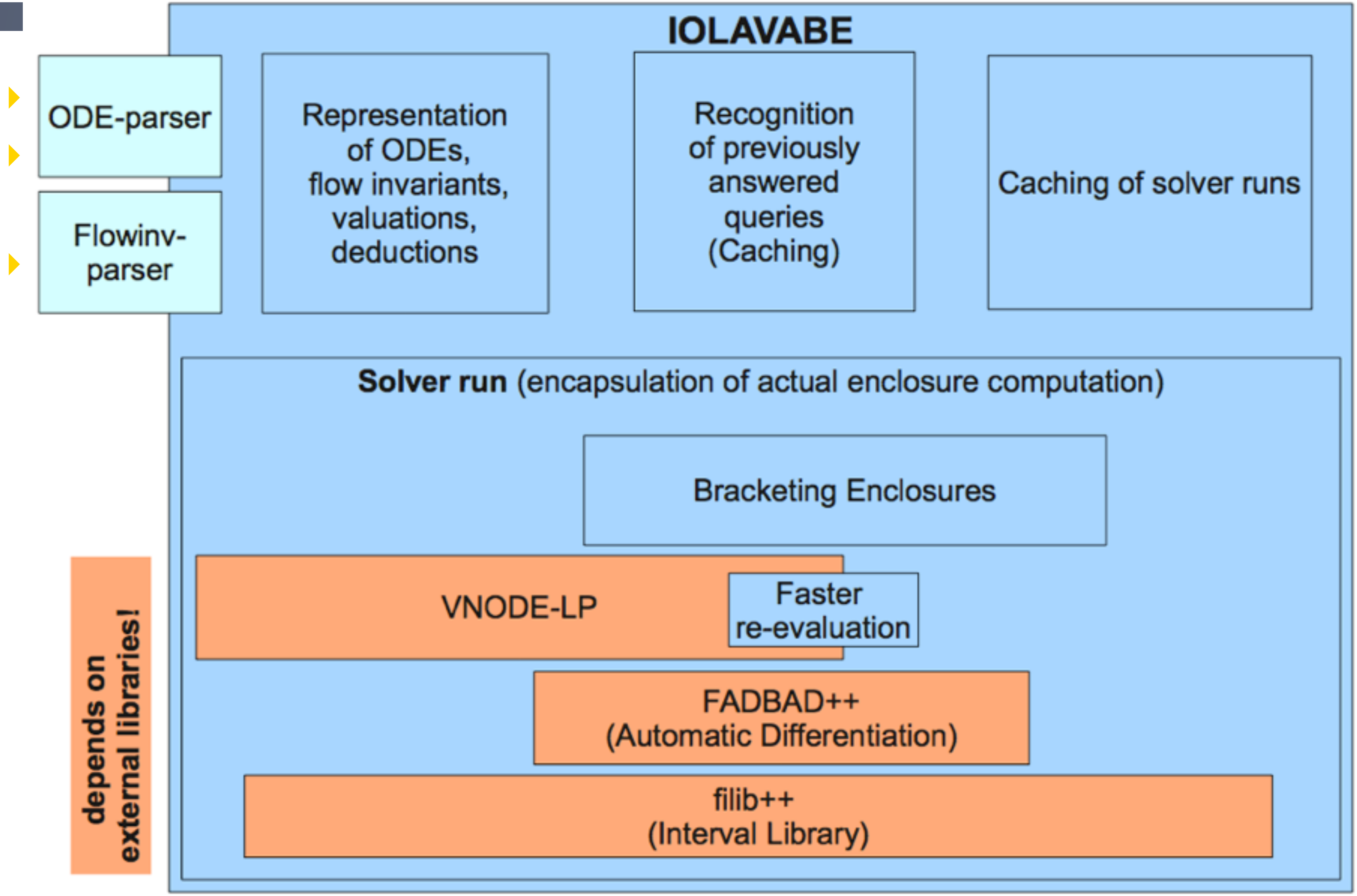
iSAT-ODE Layer

- detects independent group of ODEs
- detects when flow invariants are being left

iSAT-ODE Layer

- algorithm's parameters are exposed to the outside
- parsers for ODEs and flow invariants
  offer string interface

IOLAVABE Architecture Sketch



IOLAVABE

- IOLAVABE :
  the iSAT-ODE layer around VNODE-LP and bracketing enclosures
- gives a high-level interface for generating enclosures of ODE constraints
- Source code available for not-for-profit civilian scientific research : **try it !**

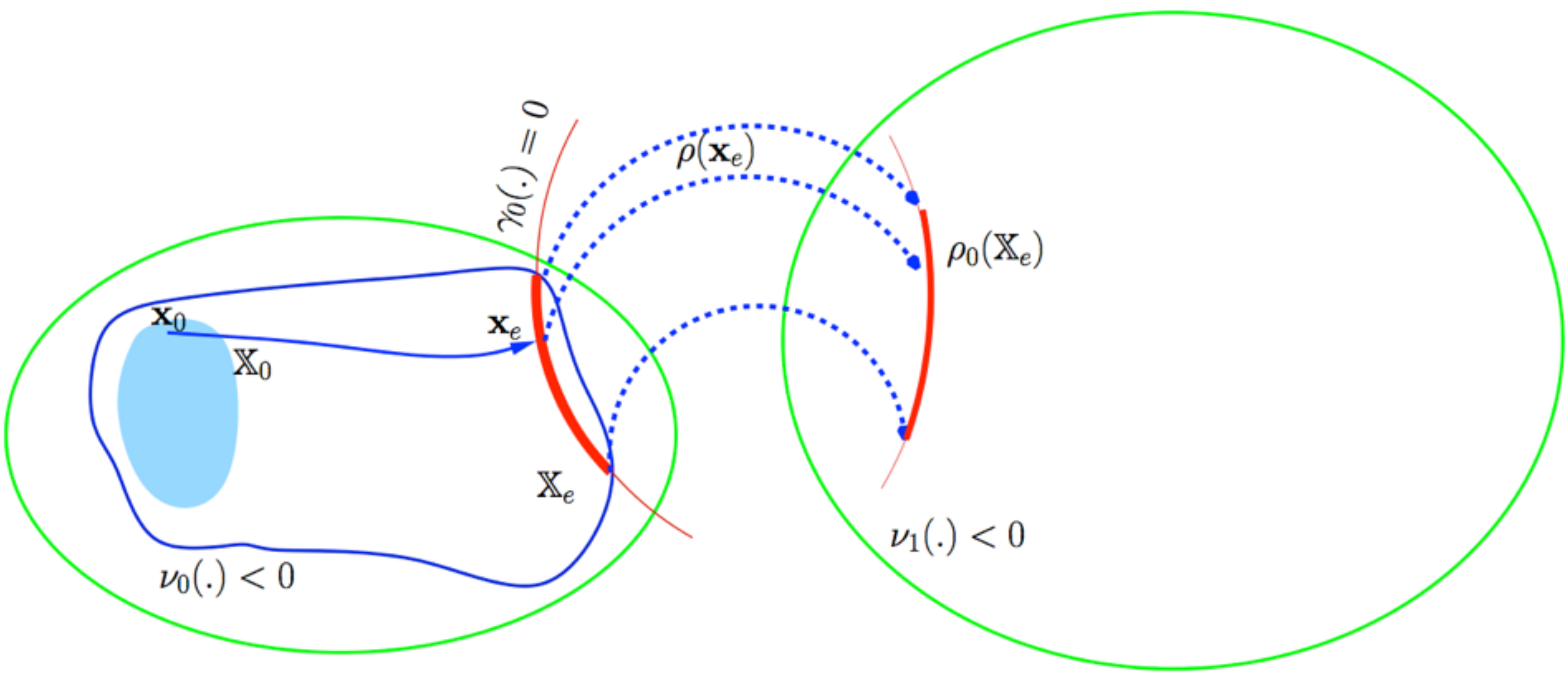
Outline

- Safety Critical Systems
- Nonlinear Continuous Reachability
- Nonlinear Hybrid Reachability
- Satisfiability mod ODE

Hybrid Cyber-Physical Systems

Hybrid reachability

- Continuous reachability
- Guard conditions, jumps & resets



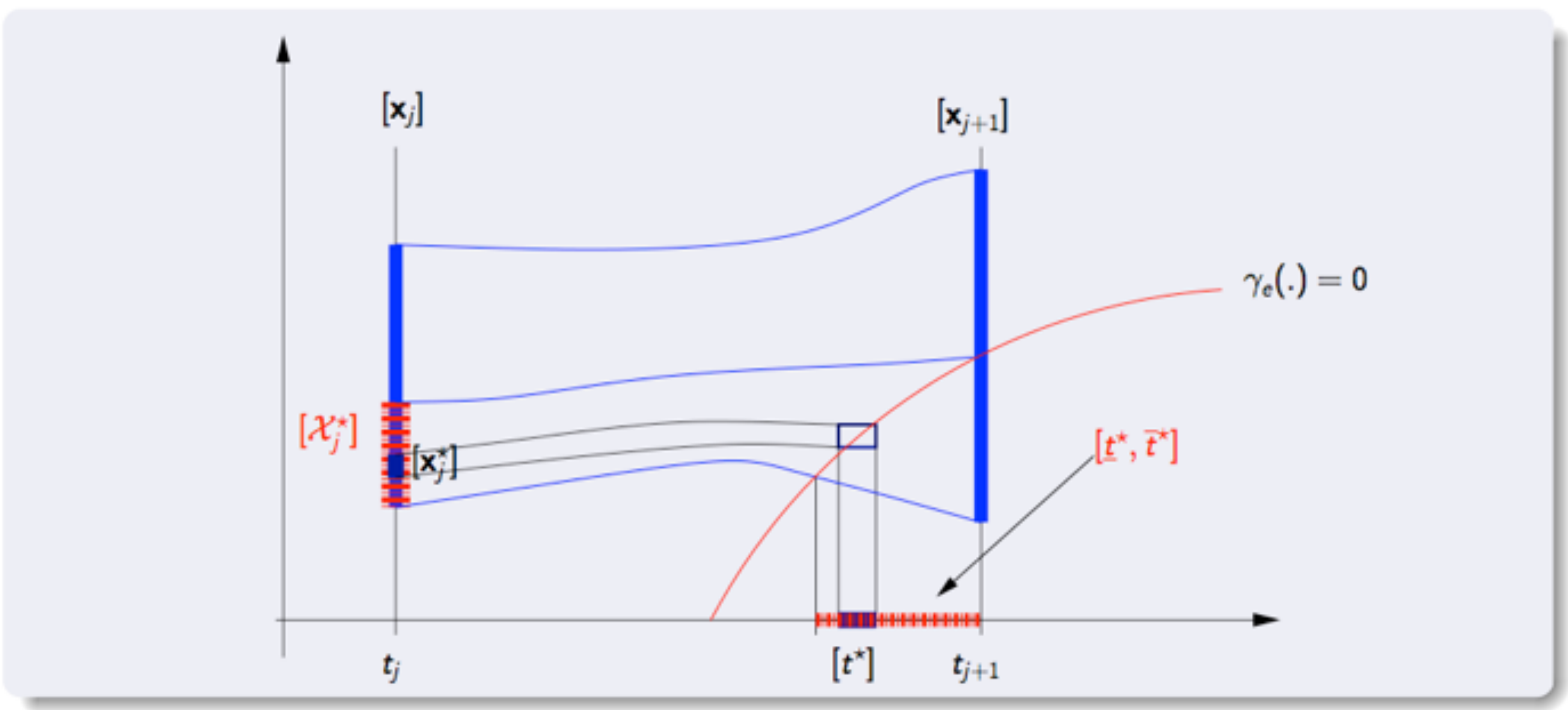
Hybrid Reachability Analysis

UNIVERSITE D'ORLEANS

Guaranteed event detection & localization

- **An interval constraint propagation approach**
  - (Ramdani, et al., Nonlinear Analysis Hybrid Systems 2011)

Time grid → $t_0 < t_1 < t_2 < \cdots < t_N$



Compute $[\underline{t}^\star, \overline{t}^\star] \times [\mathcal{X}_j^\star]$

Hybrid Reachability Analysis

Guaranteed event detection & localization

- **An interval constraint propagation approach**
  - (Ramdani, et al., Nonlinear Analysis Hybrid Systems 2011)

Time grid → $t_0 < t_1 < t_2 < \cdots < t_N$

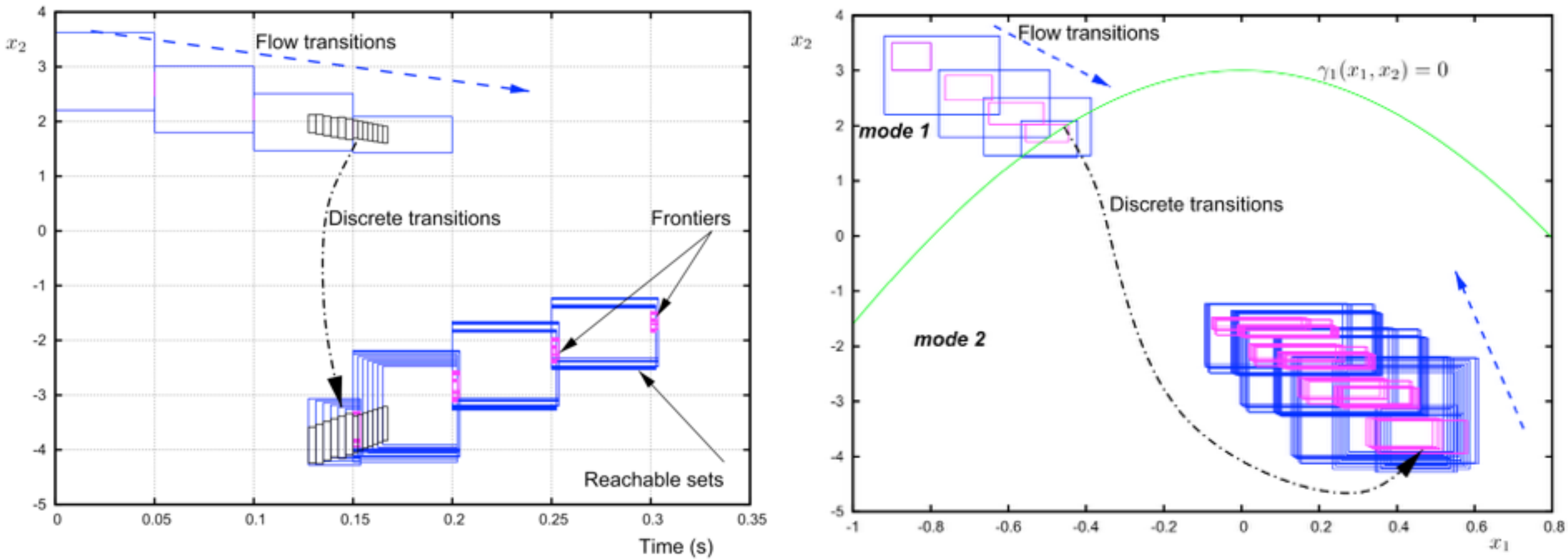
- $[\mathbf{x}](t) = \text{Interval Taylor Series (ITS)}(t, [\mathbf{x}_j], [\tilde{\mathbf{x}}_j])$
- $\gamma([\mathbf{x}](t)) = 0$

$\Rightarrow \gamma \circ \text{ITS}(t, \mathbf{x}_j, [\tilde{\mathbf{x}}_j]) \to \psi(t, \mathbf{x}_j)$

Solve CSP $([t_j, t_{j+1}] \times [\mathbf{x}_j], \psi(.,.) \ni 0)$

Hybrid Reachability Analysis

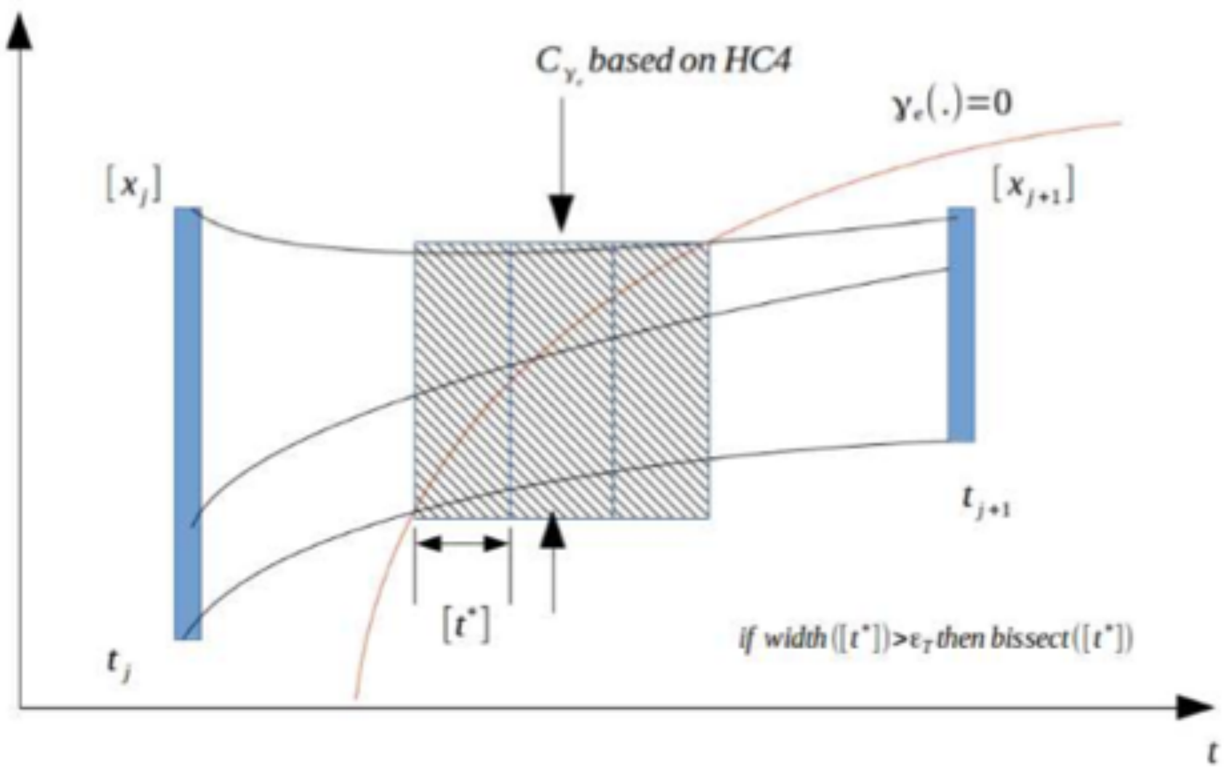
- **Guaranteed event detection & localization**
  - **An interval constraint propagation approach**
    - (Ramdani, et al., Nonlinear Analysis Hybrid Systems 2011)



Hybrid Reachability Analysis

Detecting and localizing events

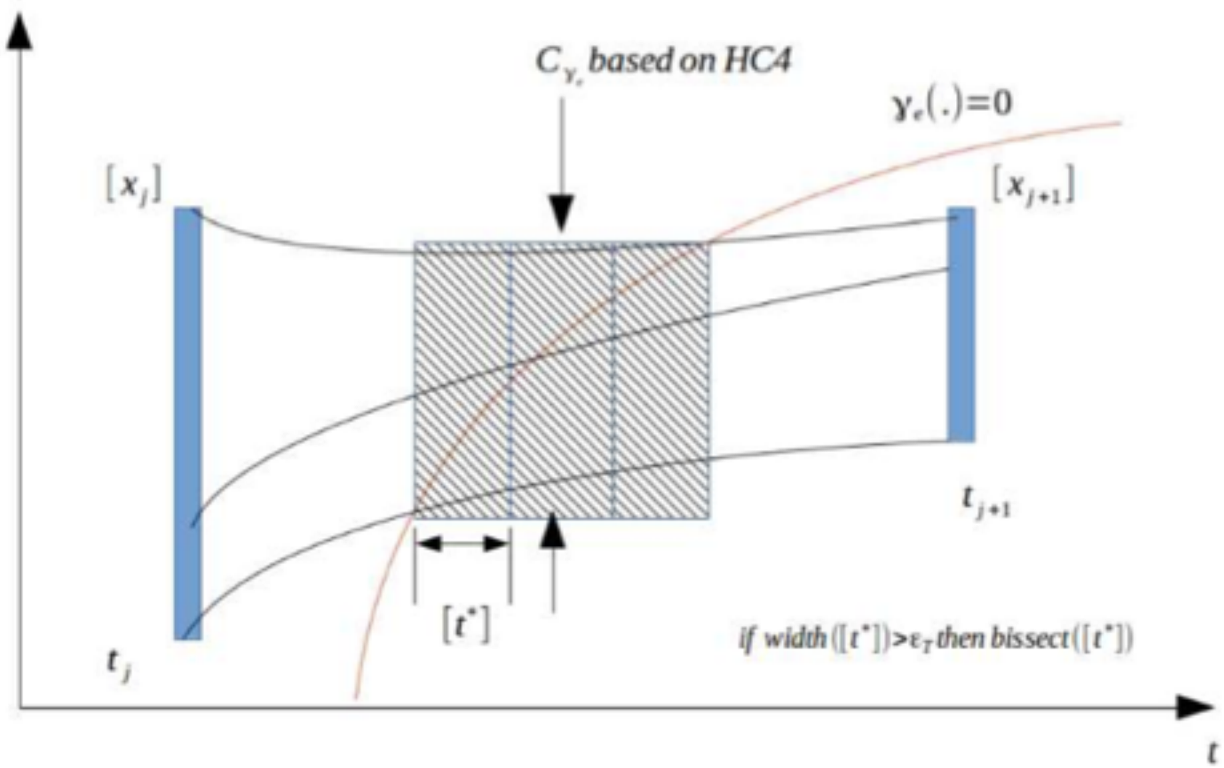
- **Improved and enhanced version**
  - (Maïga, et al., IEEE CDC 2013, ECC 2014)

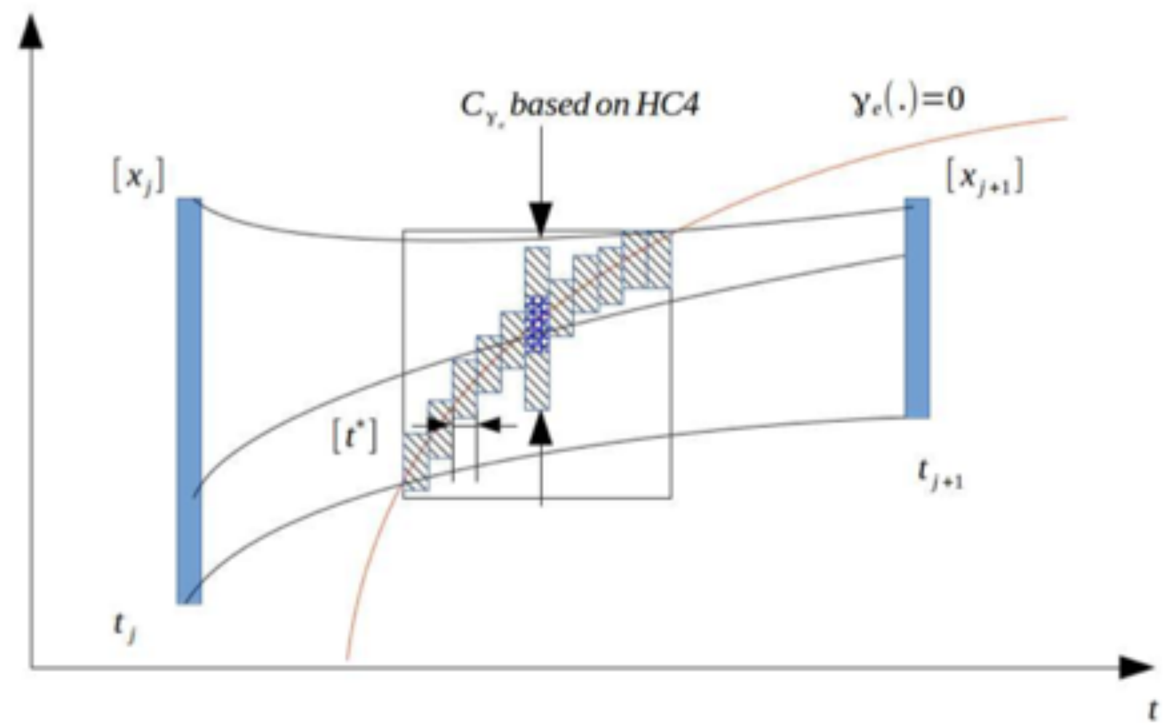


Hybrid Reachability Analysis

Detecting and localizing events

- **Improved and enhanced version**
  - (Maïga, et al., IEEE CDC 2013, ECC 2014)

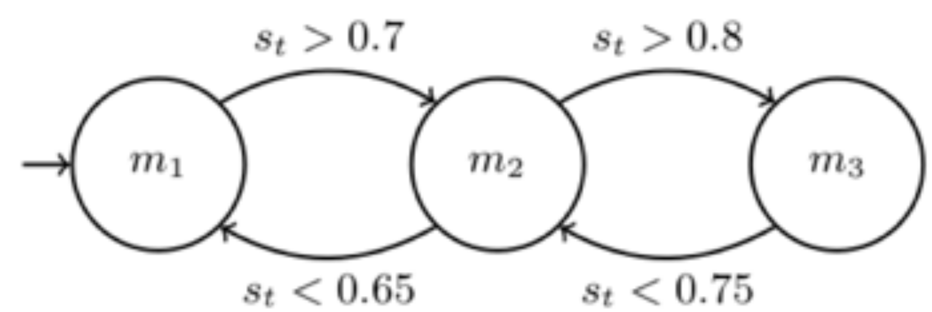




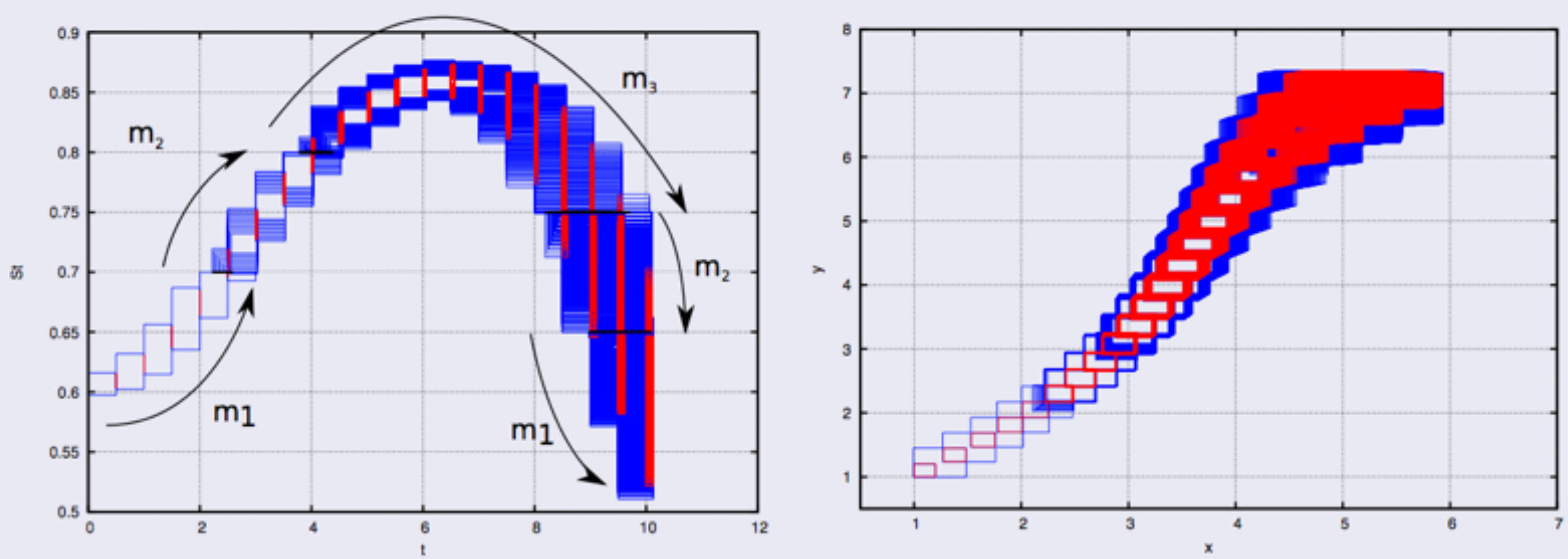
Hybrid Reachability Analysis

Detecting and localizing events

- **Improved and enhanced version**
  - (Maïga, et al., IEEE CDC 2013, ECC 2014)



$\sigma = [0, 0.01]$ and h=0.5



(e) St×t (f) Y×X space

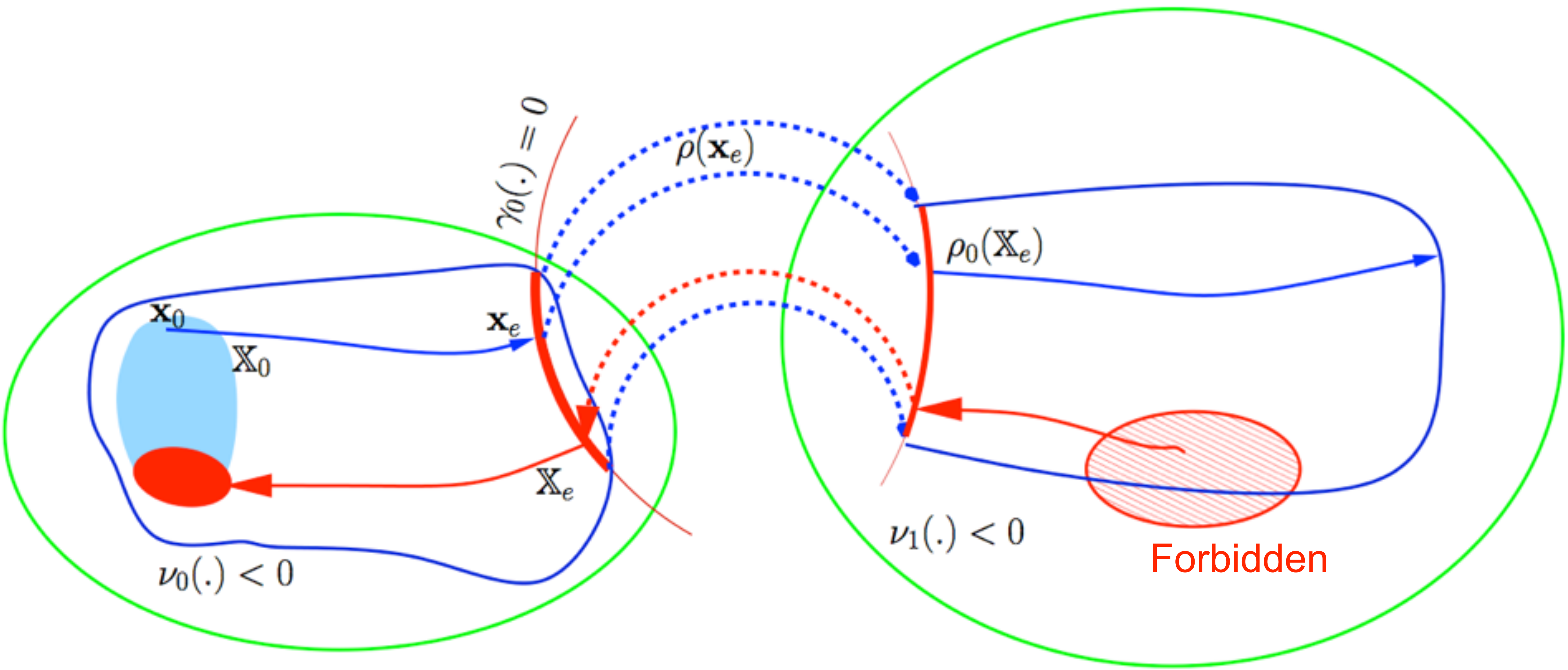
CPU times=87s with HC4 contractor
CPU times> 1h without HC4 contractor

Outline

- Safety Critical Systems
- Nonlinear Continuous Reachability
- Nonlinear Hybrid Reachability
- Satisfiability mod ODE

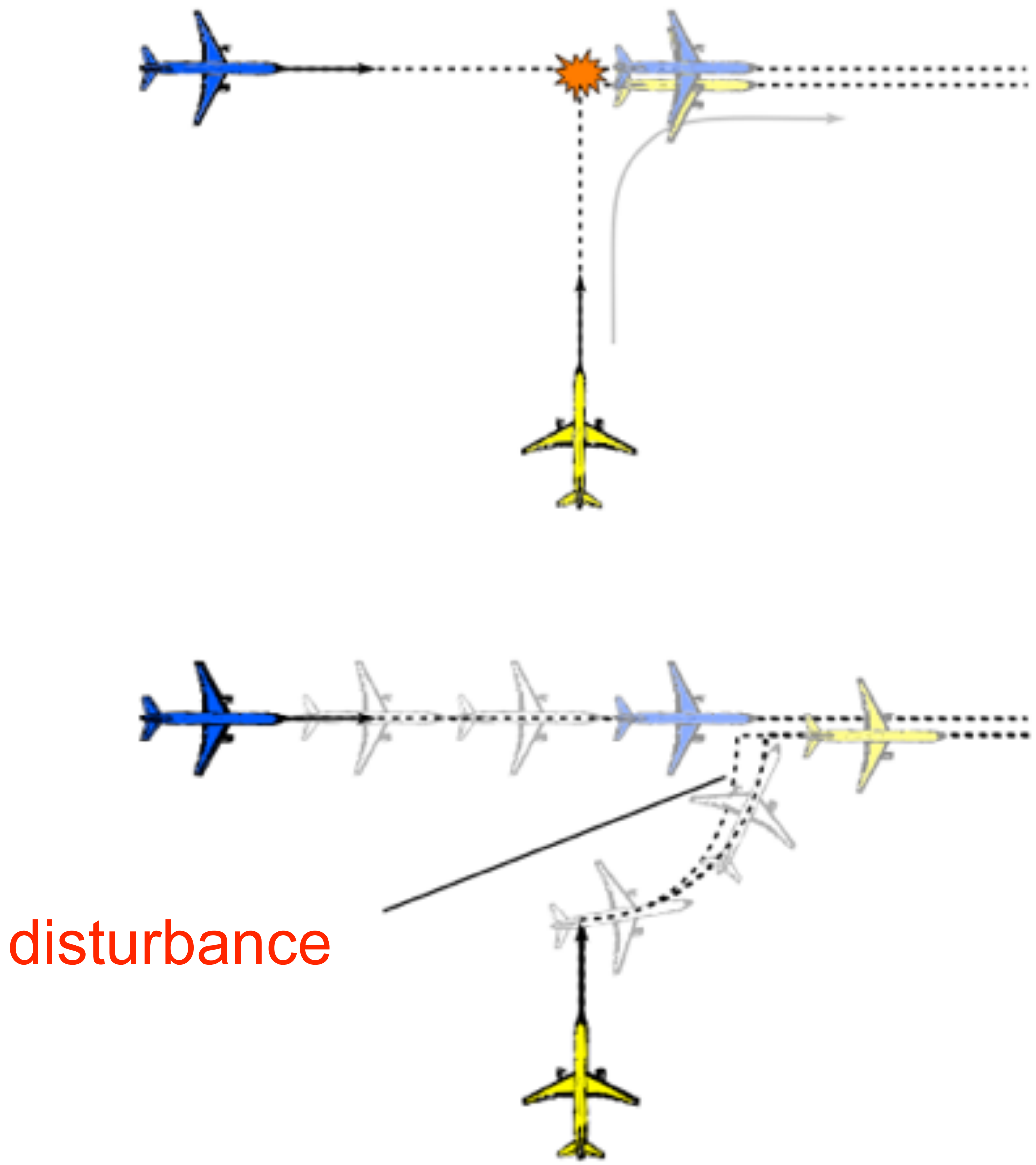
Verification of Hybrid Systems

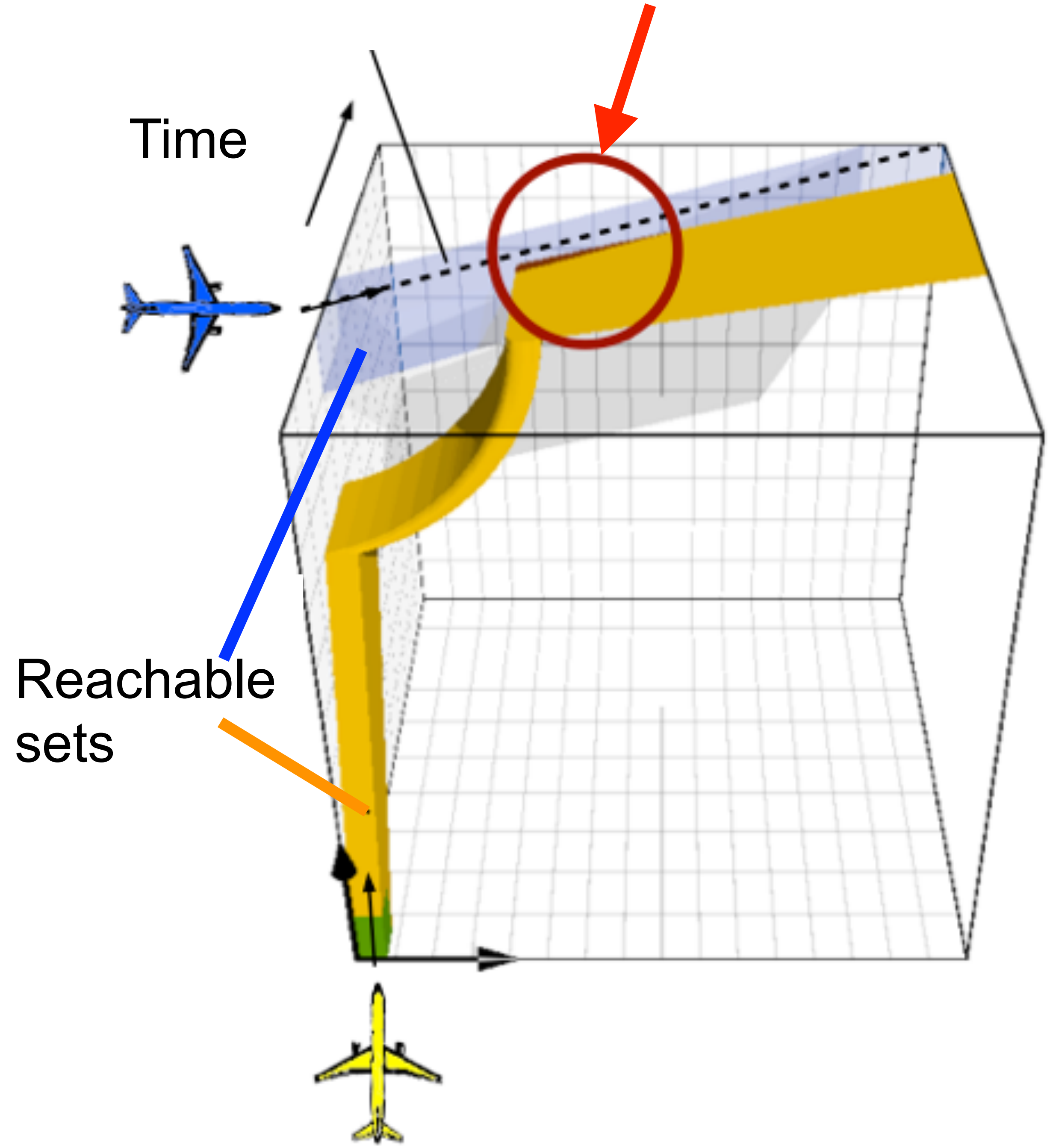
- **Verification :**
  - **Reachability of a forbidden area**



Verification of Hybrid Systems

- **Aircraft trafic control** **[Tomlin, et al.]** **Collision possible!**



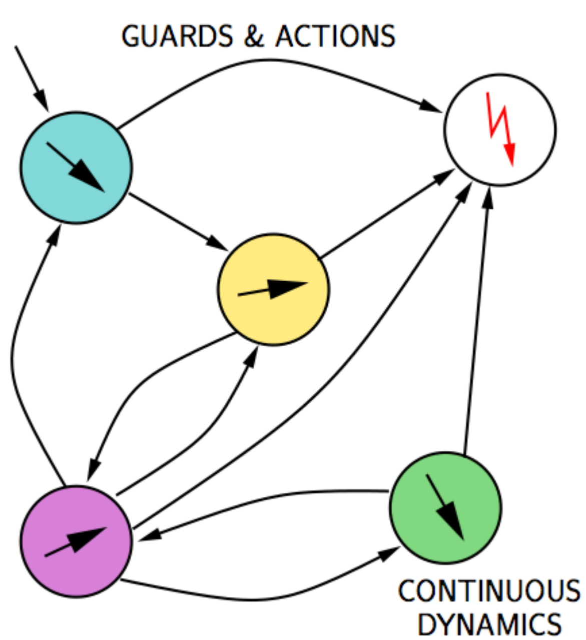


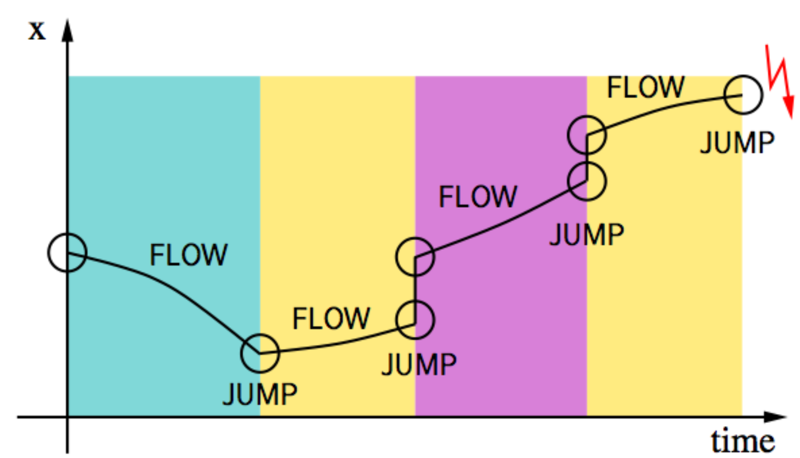
Verification

Bounded Model Checking

- Can the system reach an **unsafe** state within k (discrete or continuous) transition steps ?
- Check **satisfiability** of a **SAT Mod ODE** formula

$$\Phi_k := init[0] \wedge trans[0,1] \wedge \cdots \wedge trans[k-1,k] \wedge target[k]$$

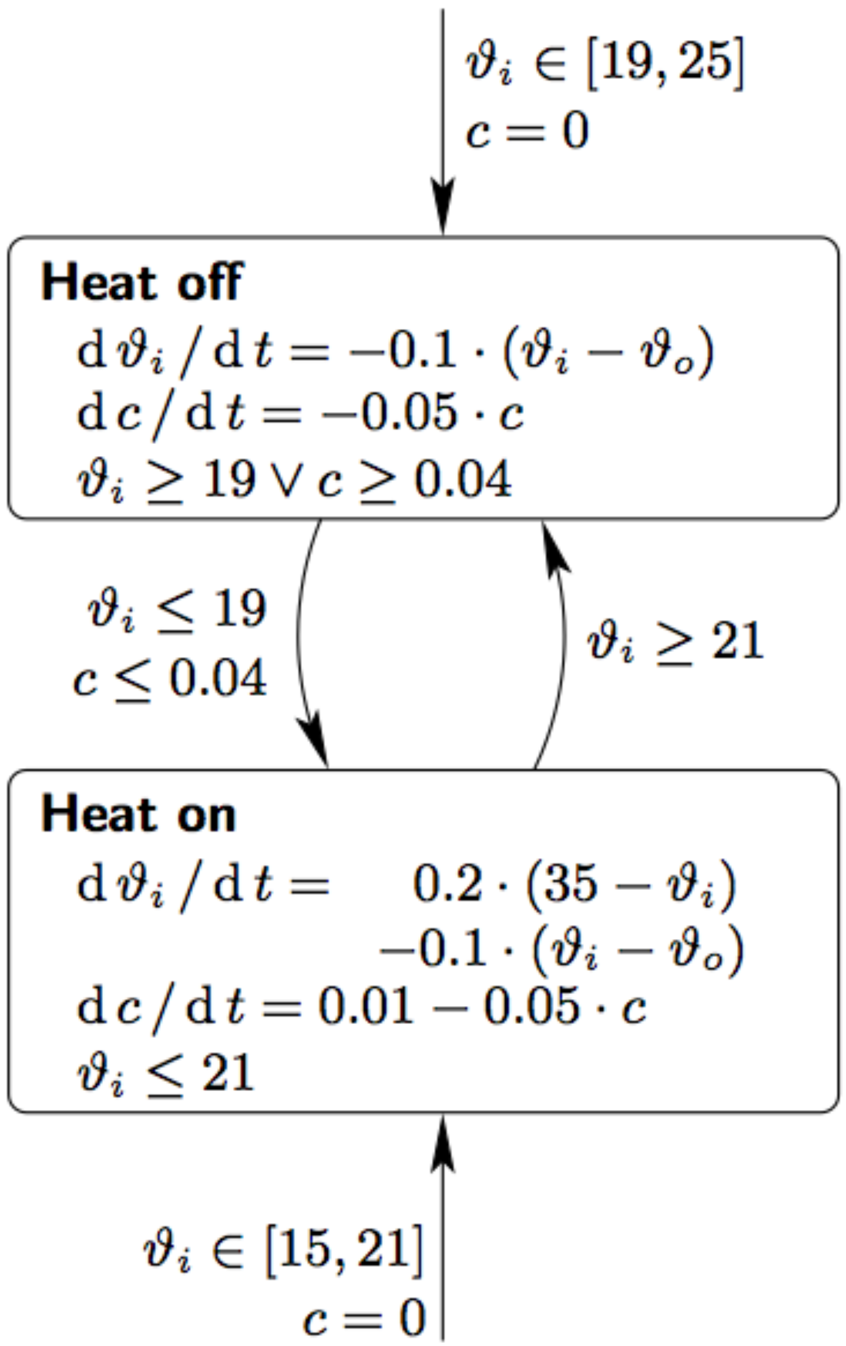




error trace

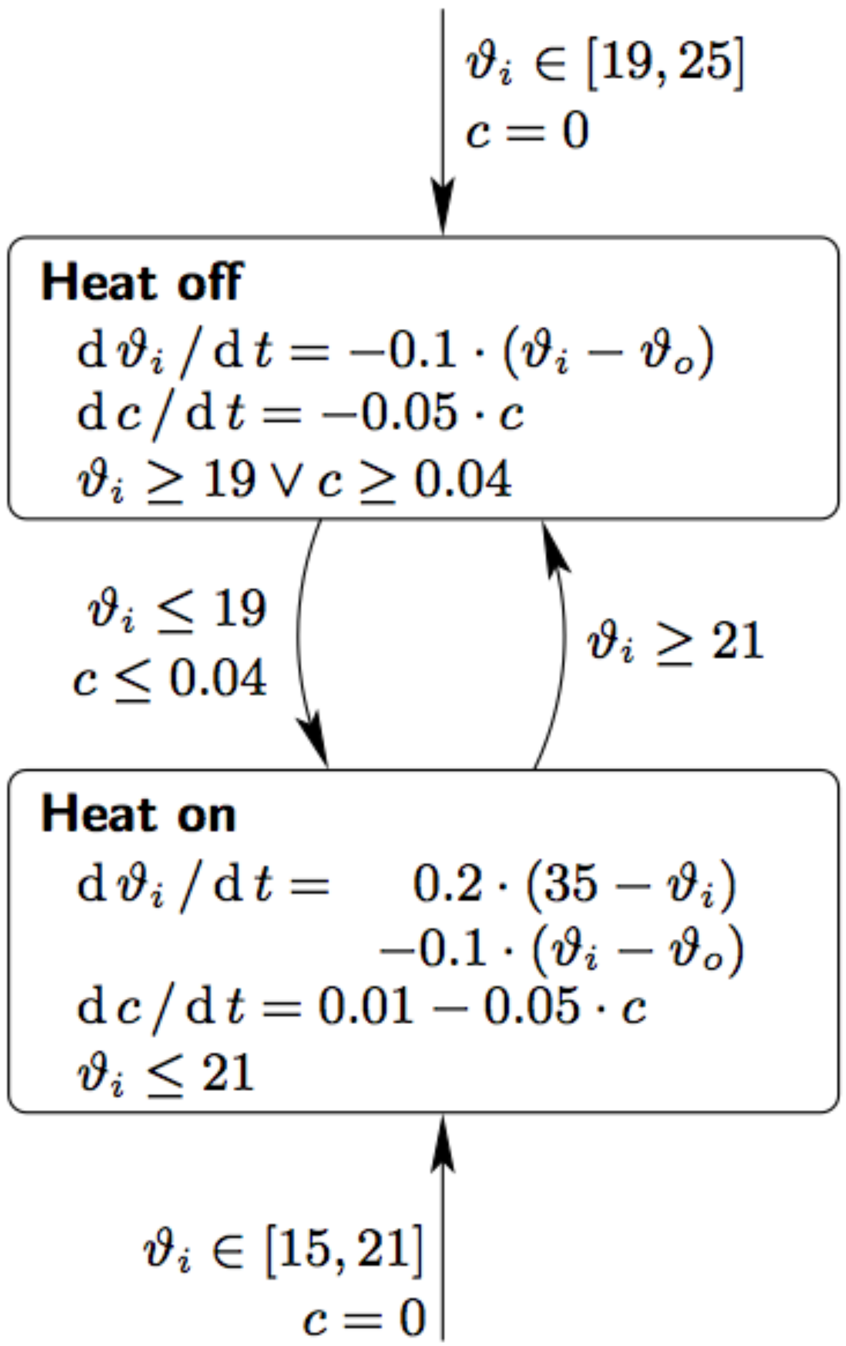
Bounded Model Checking

Bounded Model Checking



$$
\begin{aligned}
&\mathit{init} = \\
&\qquad -10 \leq \vartheta_o \leq 20 \wedge c = 0 \\
&\qquad \wedge \left(\begin{array}{l} \quad 19 \leq \vartheta_i \leq 25 \wedge \neg on \\ \vee \;\; 15 \leq \vartheta_i \leq 21 \wedge on \end{array} \right) \\
&\mathit{trans} = \\
&\qquad\quad (\;\; \neg on \wedge on' \wedge \vartheta_i \leq 19 \wedge c \leq 0.04 \\
&\qquad\quad \wedge \;\; \vartheta_i' = \vartheta_i \wedge \vartheta_o' = \vartheta_o \wedge c' = c) \\
&\qquad \vee \; (\;\; on \wedge \neg on' \wedge \vartheta_i \geq 21 \\
&\qquad\quad \wedge \;\; \vartheta_i' = \vartheta_i \wedge \vartheta_o' = \vartheta_o \wedge c' = c) \\
&\qquad \vee \; (\;\; \neg on \wedge \neg on' \\
&\qquad\quad \wedge \;\; \tfrac{\mathrm{d}\vartheta_i}{\mathrm{d}t} = -0.1(\vartheta_i - \vartheta_o) \\
&\qquad\quad \wedge \;\; \tfrac{\mathrm{d}c}{\mathrm{d}t} = -0.05c \\
&\qquad\quad \wedge \;\; (\vartheta_i' \geq 19 \vee c' \geq 0.04) \wedge \vartheta_o' = \vartheta_o) \\
&\qquad \vee \; (\;\; on \wedge on' \\
&\qquad\quad \wedge \;\; \tfrac{\mathrm{d}\vartheta_i}{\mathrm{d}t} = 0.2 \cdot 35 - 0.3\vartheta_i + 0.1\vartheta_o \\
&\qquad\quad \wedge \;\; \tfrac{\mathrm{d}c}{\mathrm{d}t} = 0.01 - 0.05c \\
&\qquad\quad \wedge \;\; \vartheta_i' \leq 21 \wedge \vartheta_o' = \vartheta_o) \\
&\mathit{target} = \\
&\qquad (c > 0.1)
\end{aligned}
$$

Bounded Model Checking



$$
\begin{aligned}
&\mathit{init} = \\
&\qquad -10 \leq \vartheta_o \leq 20 \wedge c = 0 \\
&\qquad \wedge \left(\begin{array}{l} \quad 19 \leq \vartheta_i \leq 25 \wedge \neg on \\ \vee \ 15 \leq \vartheta_i \leq 21 \wedge on \end{array} \right)
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{trans} = \\
&\qquad (\ \neg on \wedge on' \wedge \vartheta_i \leq 19 \wedge c \leq 0.04 \\
&\qquad \wedge\ \vartheta_i' = \vartheta_i \wedge \vartheta_o' = \vartheta_o \wedge c' = c) \\
&\quad \vee\ (\ on \wedge \neg on' \wedge \vartheta_i \geq 21 \\
&\qquad \wedge\ \vartheta_i' = \vartheta_i \wedge \vartheta_o' = \vartheta_o \wedge c' = c) \\
&\quad \vee\ (\ \neg on \wedge \neg on' \\
&\qquad \wedge\ \tfrac{\mathrm{d}\vartheta_i}{\mathrm{d}t} = -0.1(\vartheta_i - \vartheta_o) \\
&\qquad \wedge\ \tfrac{\mathrm{d}c}{\mathrm{d}t} = -0.05c \\
&\qquad \wedge\ (\vartheta_i' \geq 19 \vee c' \geq 0.04) \wedge \vartheta_o' = \vartheta_o) \\
&\quad \vee\ (\ on \wedge on' \\
&\qquad \wedge\ \tfrac{\mathrm{d}\vartheta_i}{\mathrm{d}t} = 0.2 \cdot 35 - 0.3\vartheta_i + 0.1\vartheta_o \\
&\qquad \wedge\ \tfrac{\mathrm{d}c}{\mathrm{d}t} = 0.01 - 0.05c \\
&\qquad \wedge\ \vartheta_i' \leq 21 \wedge \vartheta_o' = \vartheta_o)
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{target} = \\
&\qquad (c > 0.1)
\end{aligned}
$$

$$\Phi_k := \mathit{init}[0] \wedge \mathit{trans}[0,1] \wedge \cdots \wedge \mathit{trans}[k-1,k] \wedge \mathit{target}[k]$$

Bounded Model Checking

SAT mod ODE

- *Model*: init → definition of variables.
  trans[k,k+1] → transition dynamics.
- *Property*: prop
- *SAT solvers check the following formulas*:

  init ∧ ¬prop

  init ∧ trans[0,1] ∧ ¬prop

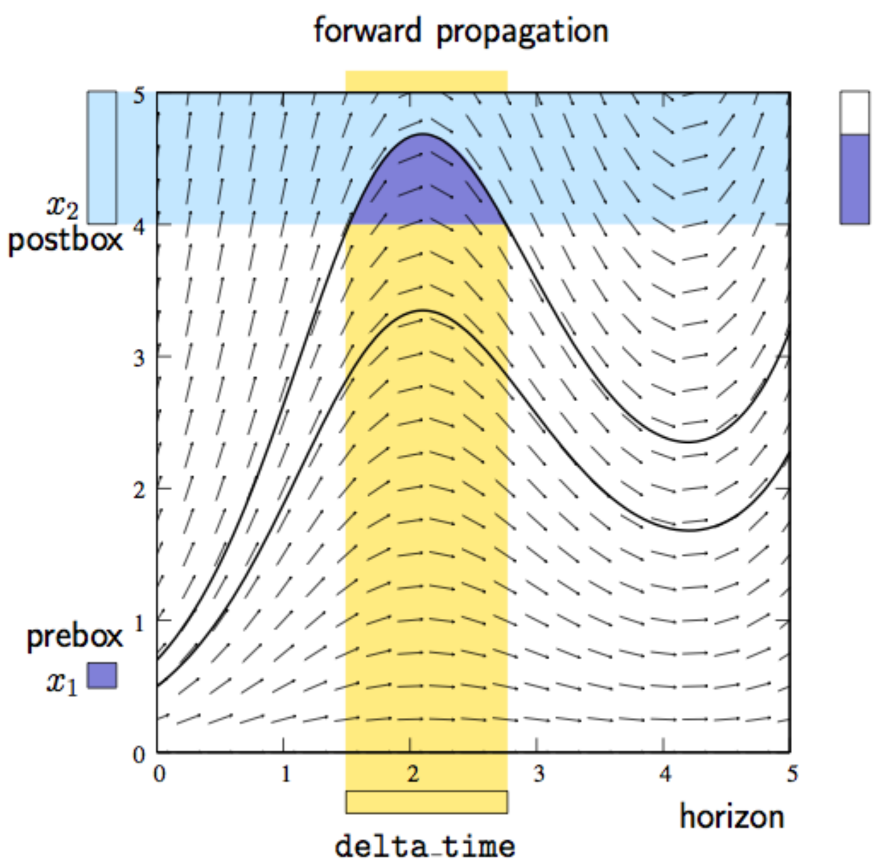
  init ∧ trans[0,1] ∧ trans[1,2] ∧ ¬prop

  init ∧ trans[0,1] ∧ trans[1,2] ∧ trans[2,3] ∧ ¬prop ...

- **If one formula is satisfiable → Property is violated !**

A SAT mod ODE approach

- **iSAT** [Fränzle, et al. 2007]
  - Interval constraint propagation. Branching...
  - Conflict-driven learning ...
- **iSAT-ODE** [Eggers, et al. 2008, 2014]
  - ODE allowed in transition dynamics
  - Enclosure of ODE solutions ...
    - **VNODE-LP**
      - **(Nedialkov, 2008)**
    - **Bounding systems**
      - **(Ramdani, et al., 2009)**



The core iSAT algorithm

[Fränzle, Herde, Ratschan, Schubert, and Teige, 2007]

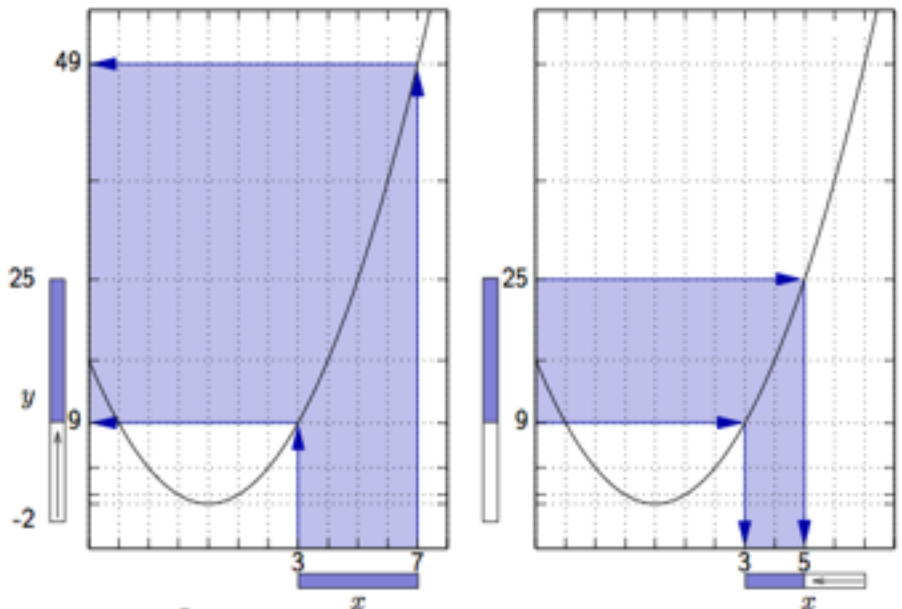
Generalization of DPLL/CDCL
solving manipulating **interval bounds**

$$x \in [3, 7], \quad y \in [-2, 25]$$

Deductions:

prune off definite non-solutions

- Unit propagation:
  $\cdots \wedge (x > 8 \vee \underline{y = x^2}) \wedge \cdots$
- Interval constraint propagation:



$$y = x^2 \wedge x \geq 3 \quad \Rightarrow y \geq 9$$
$$y = x^2 \wedge y \leq 25 \Rightarrow x \leq 5$$

Decisions:

Split interval (e.g. at its midpoint), propagate resulting bound

Conflict-driven Learning:

- Deduction can yield empty box
- Learn reasons from implication graph (conflict clause)
- Jump back undoing decisions

Termination:

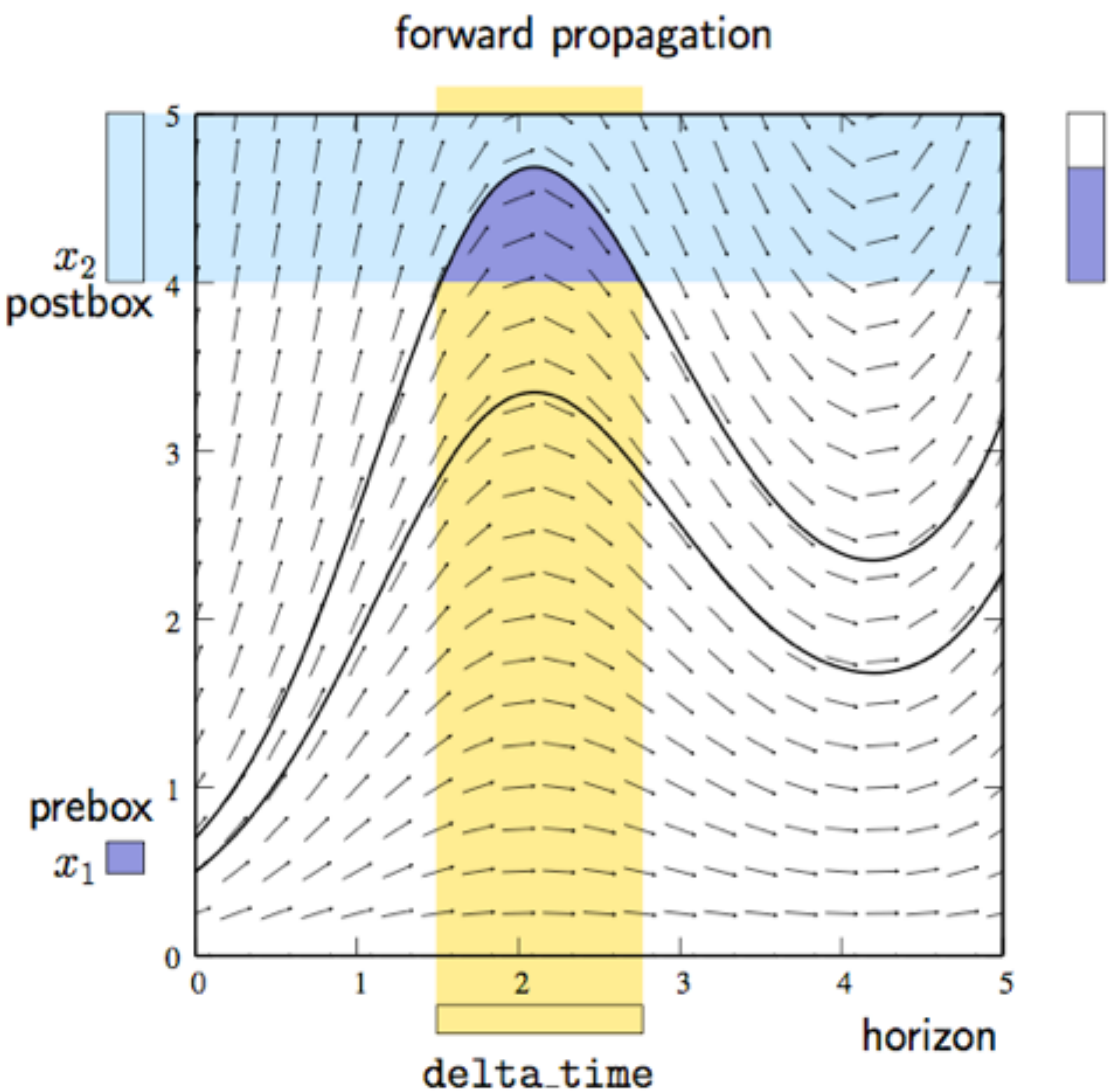
Stop search when

- unresolvable conflict is found or
- reasonably small conflict-free box found

Use **optimizations from propositional SAT** (backjumps, two-watched literal scheme, isomorphy inference, restarts, . . .)

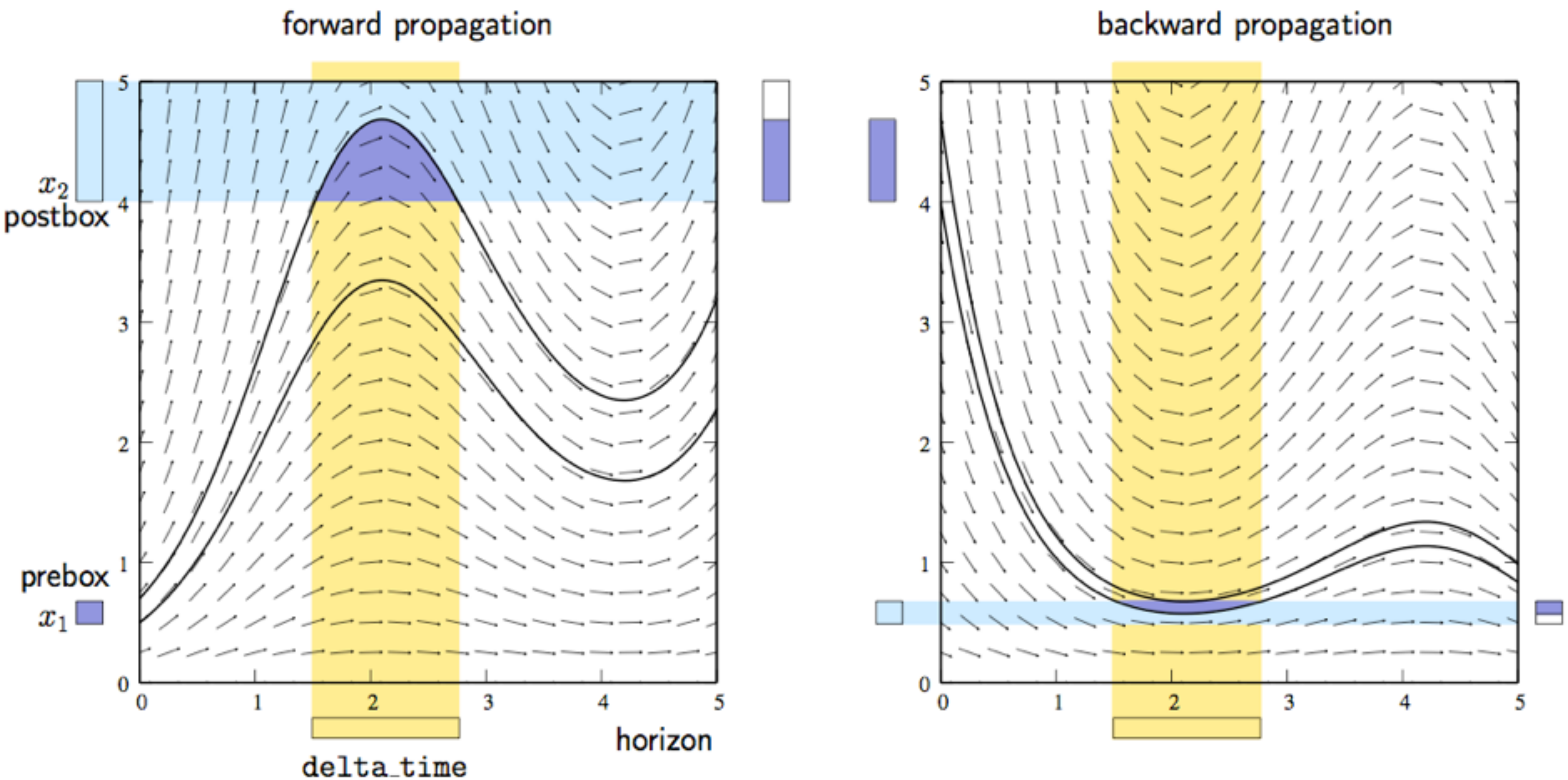
iSAT-ODE

- iSAT + ODE enclosures as propagators: *contract pre- & post-box using* forward and backward deductions



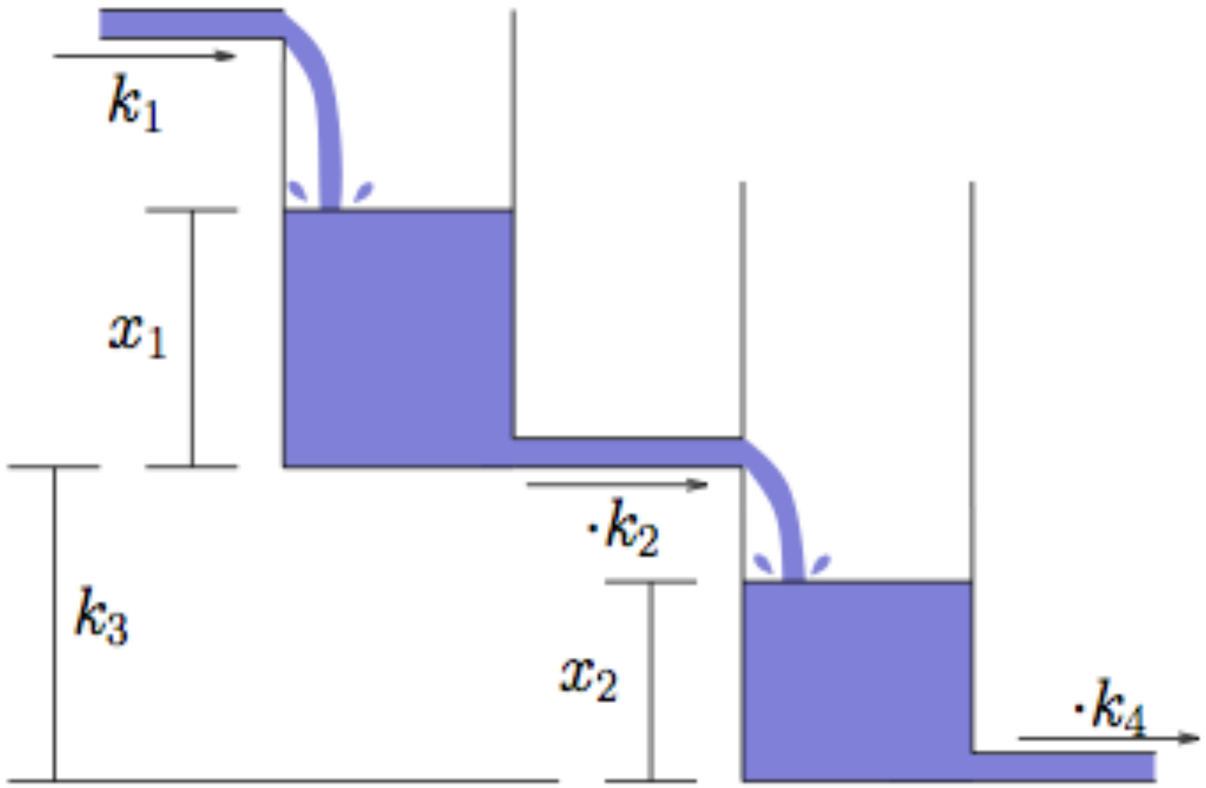
iSAT-ODE

- iSAT + ODE enclosures as propagators: *contract pre- & post-box using* forward and backward deductions



Verification

- **Example : 2-tanks system**



For $x_2 > k_3$:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} k_1 - k_2\sqrt{x_1 - x_2 + k_3} \\ k_2\sqrt{x_1 - x_2 + k_3} - k_4\sqrt{x_2} \end{pmatrix}$$

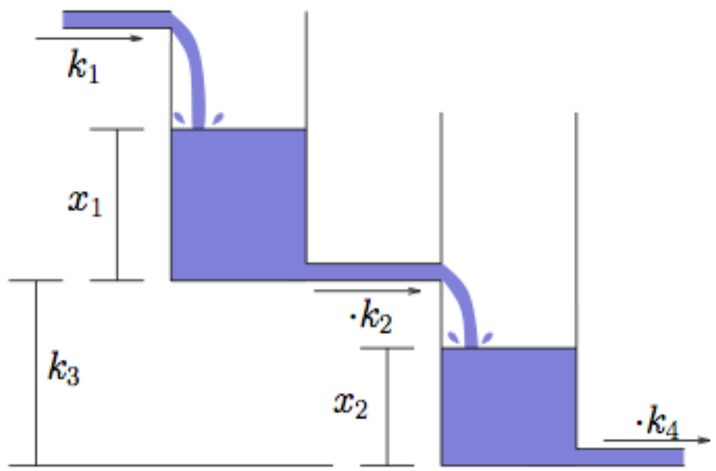
For $x_2 \leq k_3$:

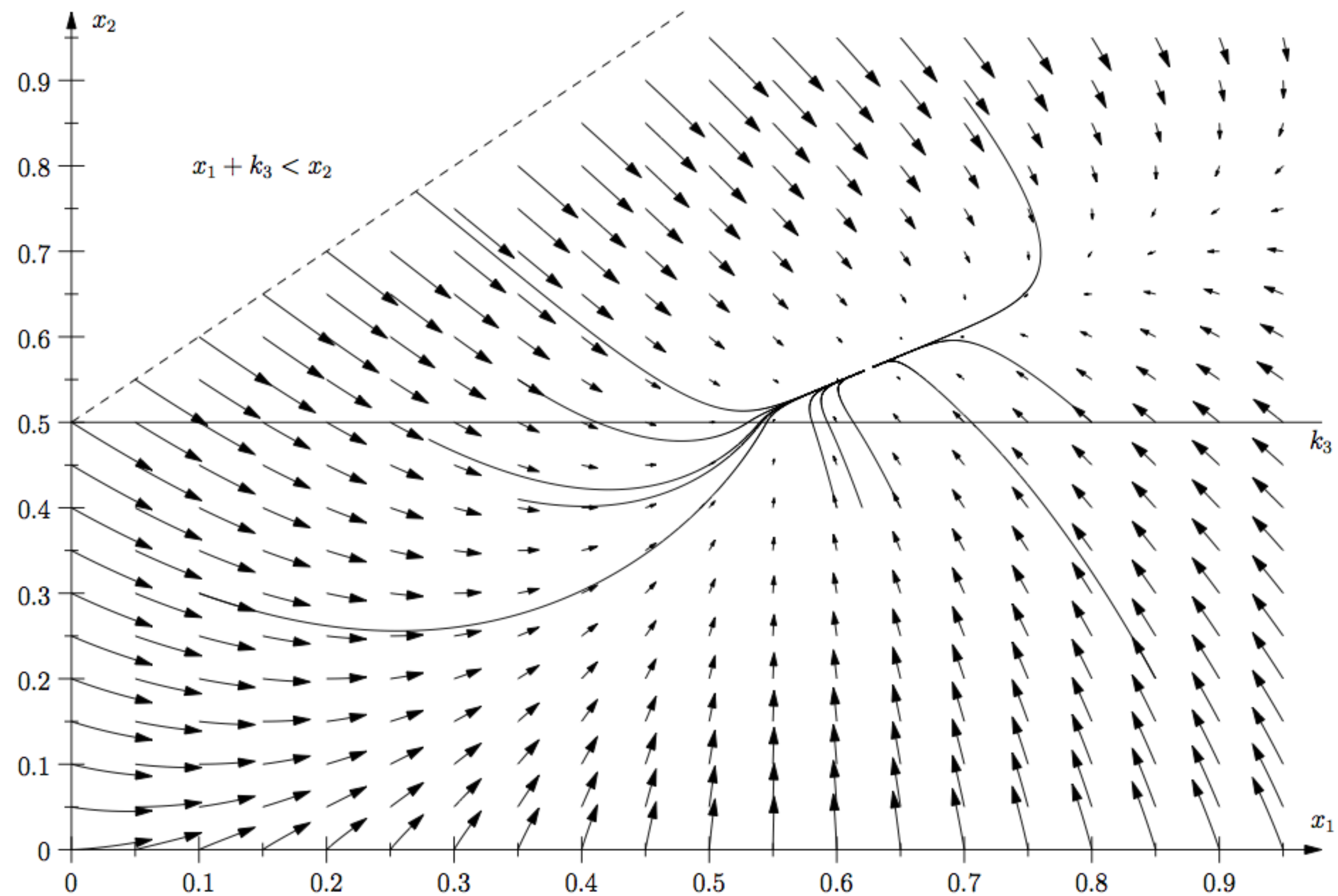
$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} k_1 - k_2\sqrt{x_1} \\ k_2\sqrt{x_1} - k_4\sqrt{x_2} \end{pmatrix}$$

$$k_1 = 0.75,\ k_2 = 1,\ k_3 = 0.5,\ k_4 = 1$$

Verification

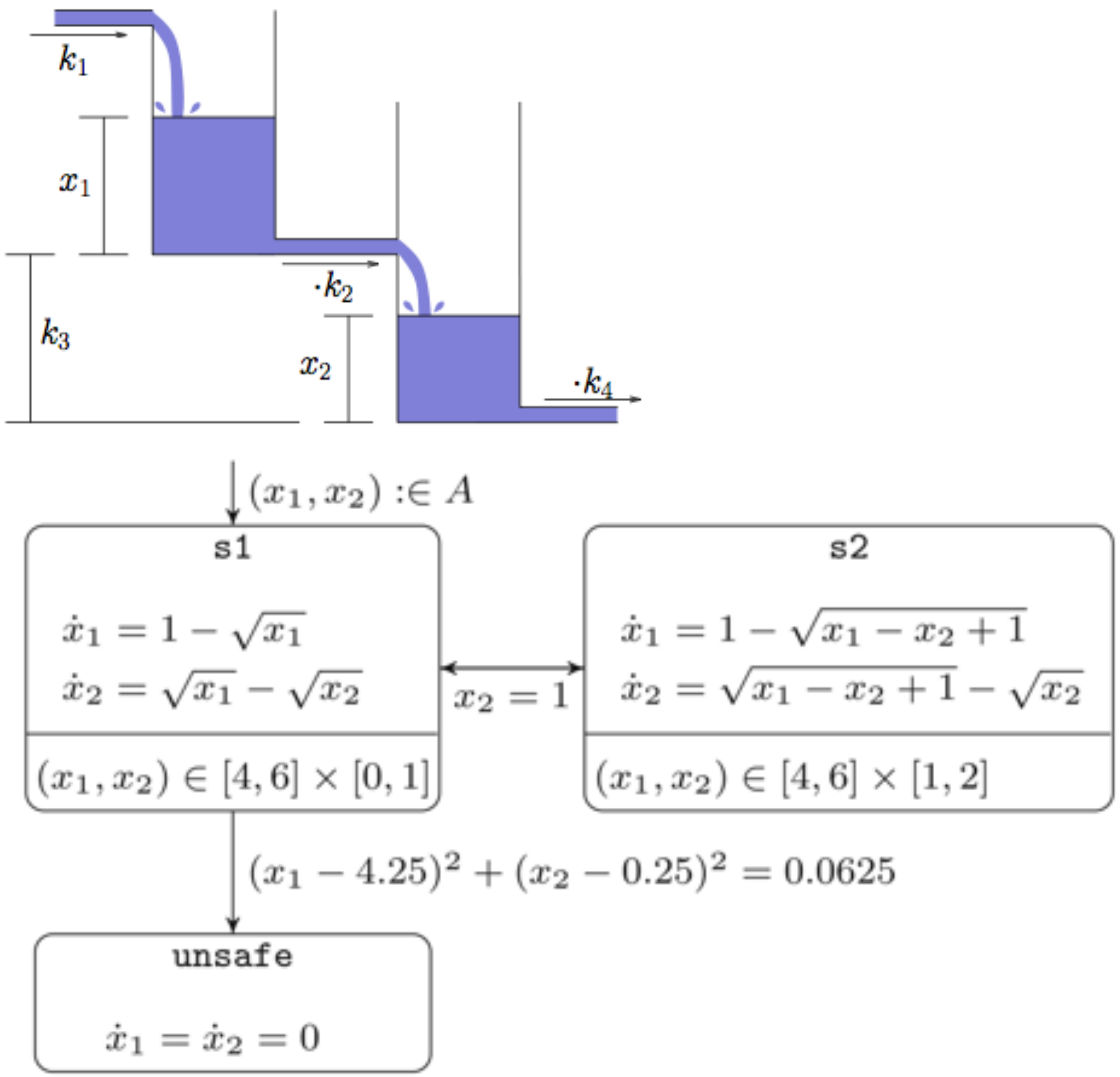
- **Example : 2-tanks system**



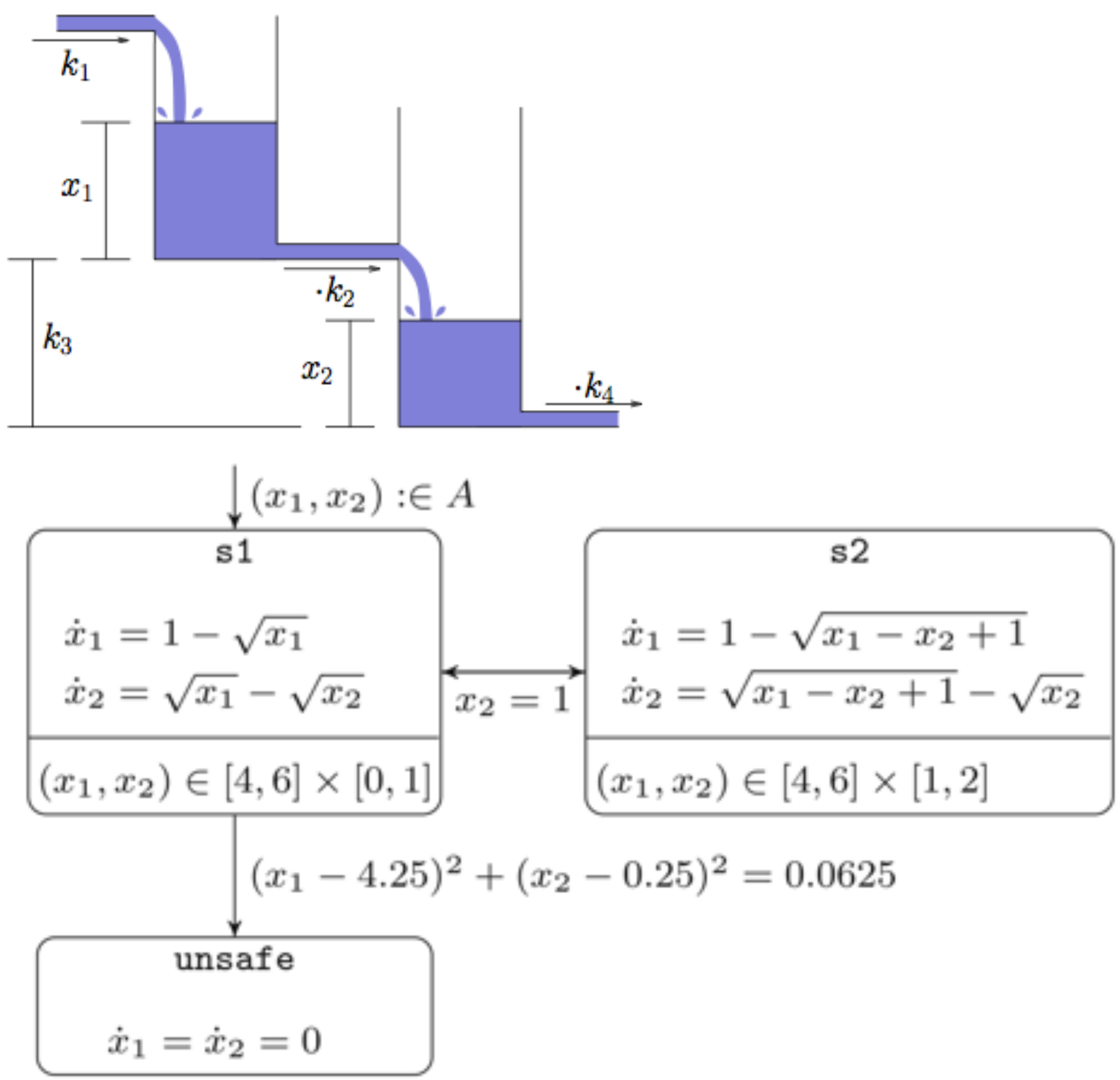


Verification

Example : 2-tanks system



Example : 2-tanks system

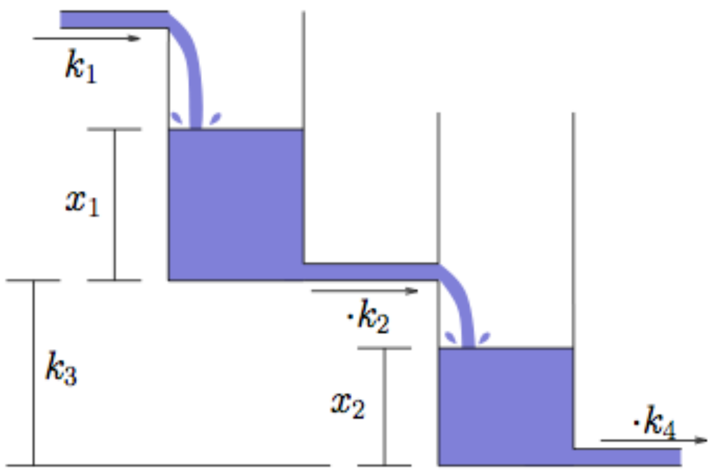


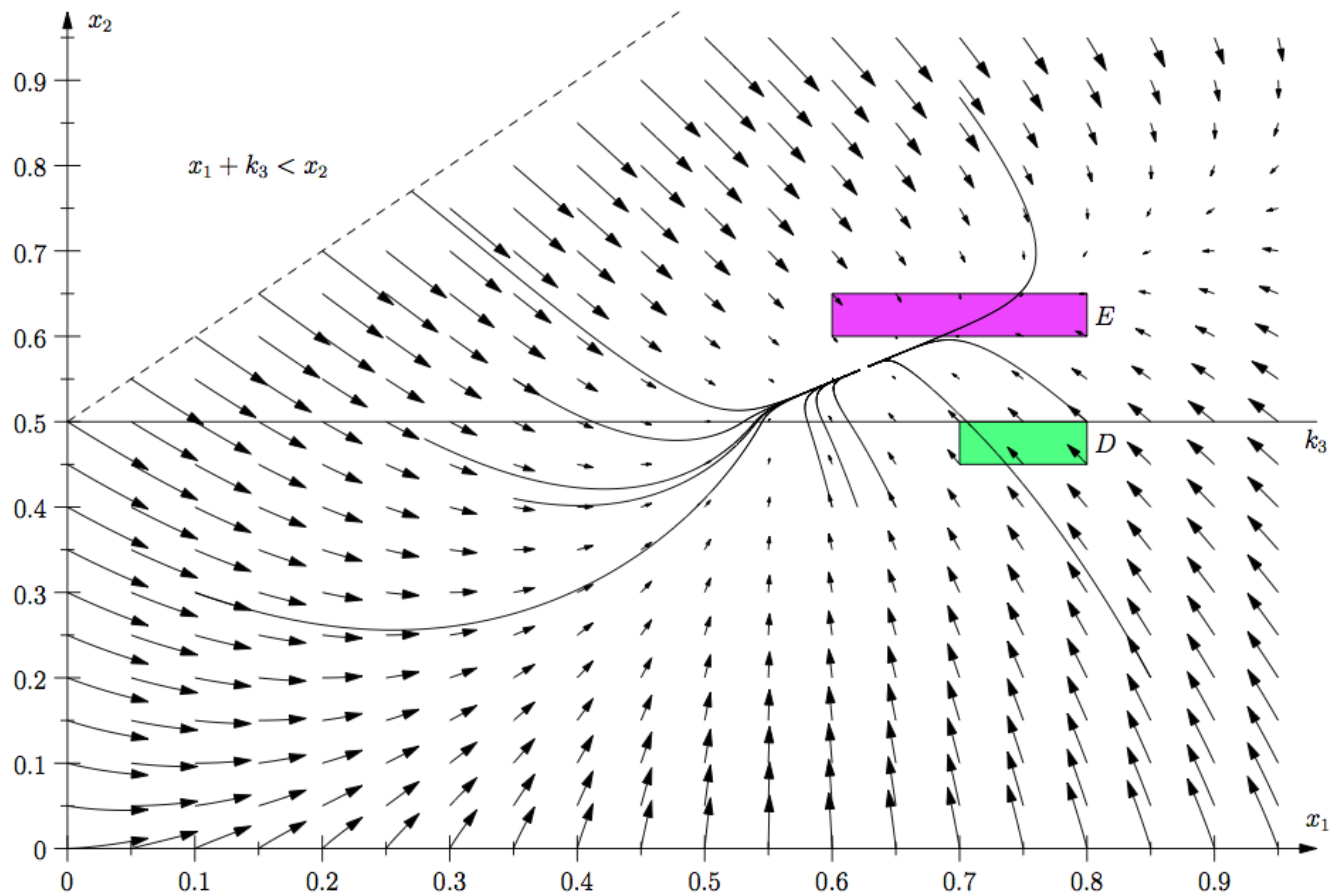

```
DECL
    float [−10, 10] x1, x2;
    float [0, 1000] time;
    float [0, 1000] delta_time;
    boole s1, s2;
    boole flow;
    boole unsafe;
INIT
    time = 0;
    x1 >= 5.25; x1 <= 5.75;
    x2 >= 0.01; x2 <= 0.5;
     s1;
    !s2;
    !unsafe;
    flow;
TRANS
    time' = time + delta_time;
    s1' + s2' = 1;

    flow and s1 −>
        (d.x1 / d.time = 1 − nrt(x1, 2));
    flow and s1 −>
        (d.x2 / d.time = nrt(x1, 2) − nrt(x2, 2));
    flow and s1 −> (x1(time) >= 4);
    flow and s1 −> (x1(time) <= 6);
    flow and s1 −> (x2(time) >= 0);
    flow and s1 −> (x2(time) <= 1);

    flow and s2 −>
        (d.x1 / d.time = 1 − nrt(x1 − x2 + 1, 2));
    flow and s2 −>
        (d.x2 / d.time = nrt(x1 − x2 + 1, 2) − nrt(x2, 2));
    flow and s2 −> (x1(time) >= 4);
    flow and s2 −> (x1(time) <= 6);
    flow and s2 −> (x2(time) >= 1);
    flow and s2 −> (x2(time) <= 2);
    flow −> ((s1 and s1') or (s2 and s2'));
    flow −> delta_time > 0;

    flow −> (!flow' or unsafe');

    !flow −> x2 = 1.0;
    !flow −> ((s1 and s2') or (s2 and s1'));
    !flow −> flow';
    !flow −> delta_time = 0;
    !flow −> (x1' = x1 and x2' = x2);

    unsafe' <−> (x1' − 4.25)^2 + (x2' − 0.25)^2 = 0.0625;
TARGET
    s1;
    unsafe;
```

Safety Verification

E non reachable from D. [Eggers, Ramdani, Nedialkov, Fränzle, 2015]

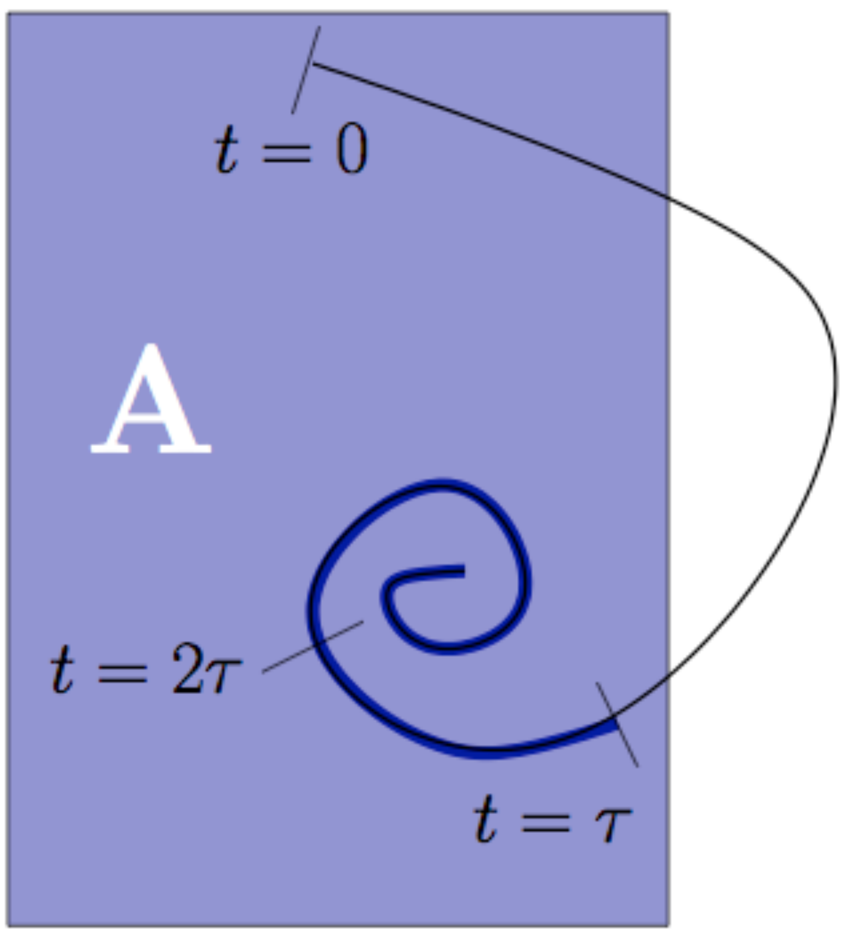
iSAT-ODE: Proof in 260s CPU 2.4 GHz AMD Opteron



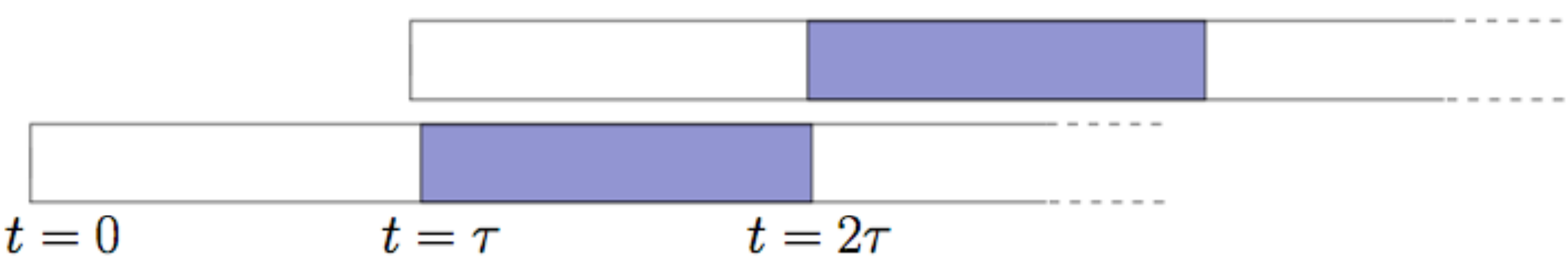


Region stability

[Podelski et Wagner, 2007] [Eggers, Ramdani, Nedialkov, Fränzle, 2015]



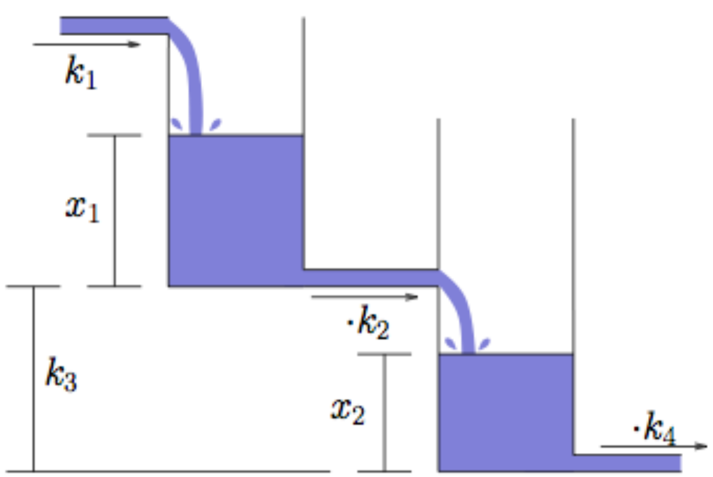
- Proof: a trajectory starting in A, stays in A during $[\tau, 2\tau]$
- SAT mod ODE formula
  Target :
  Non reached at 2τ
  or left A during $[\tau, 2\tau]$
- If UNSAT, reccurrence, time-invariance, infinite time property.

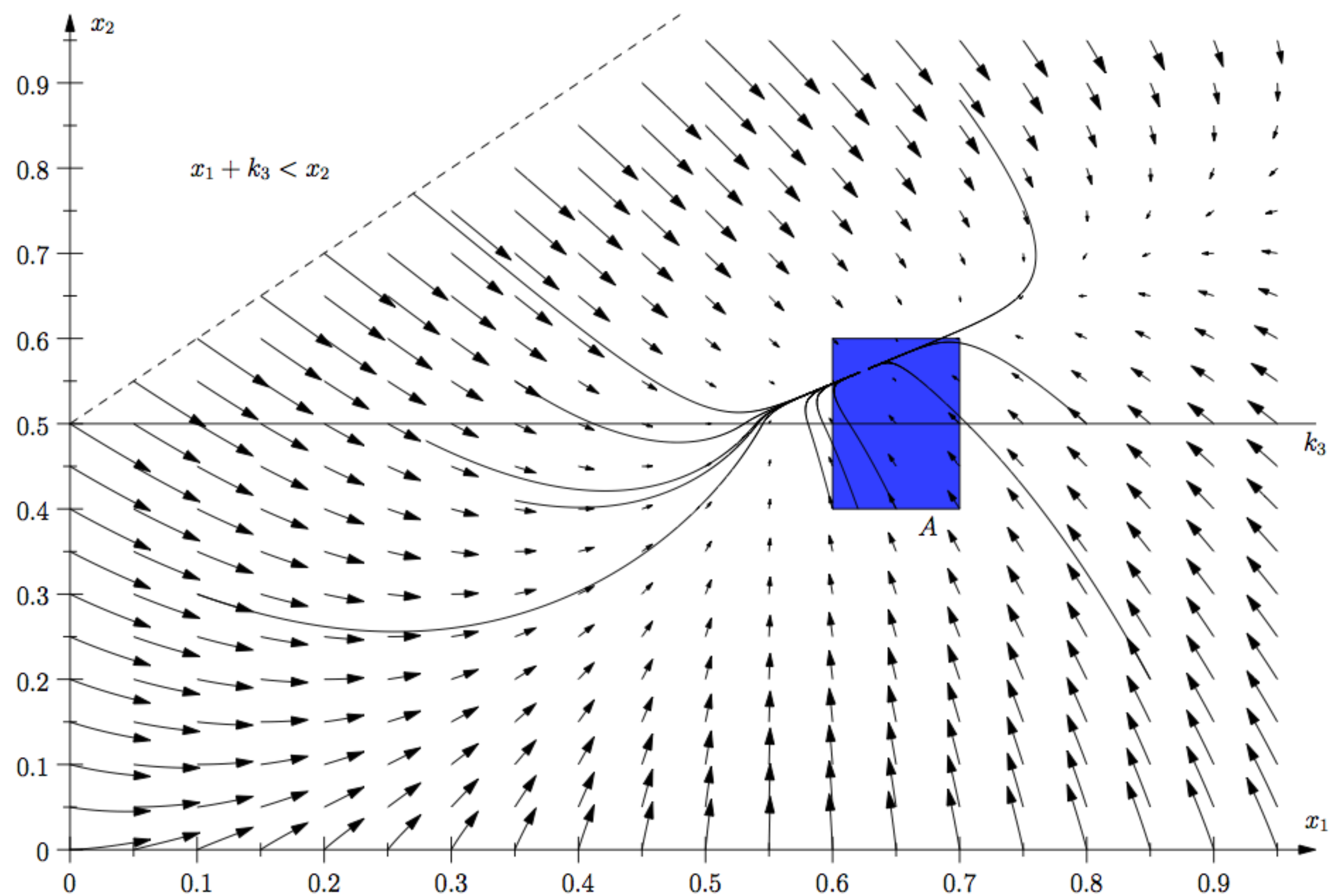


Region stability

[Eggers, Ramdani, Nedialkov, Fränzle, 2015]

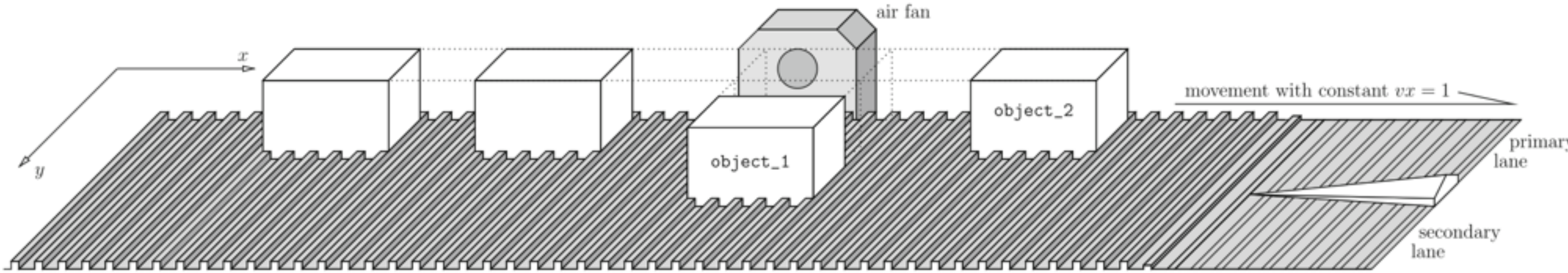
iSAT-ODE: proof in 150s CPU 2.4 GHz AMD Opteron

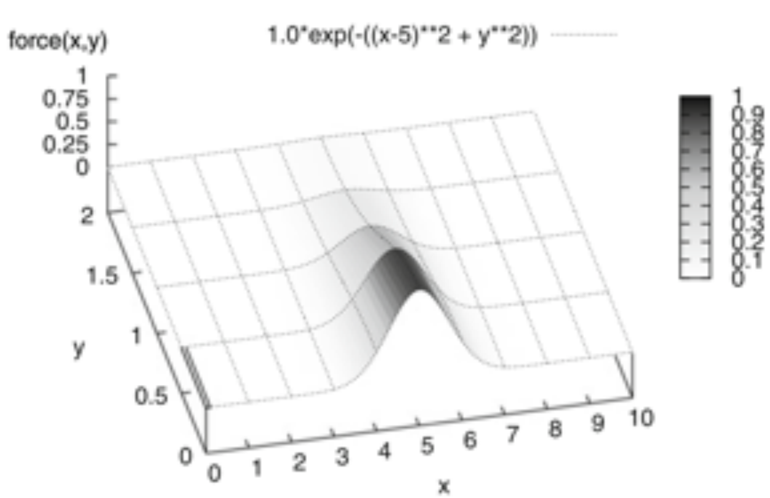




Verification

- **Example 2 : Conveyor belt.**

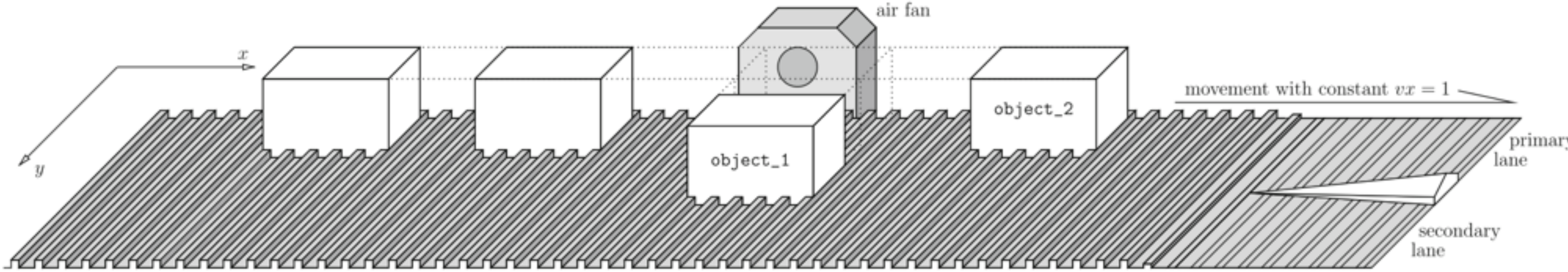




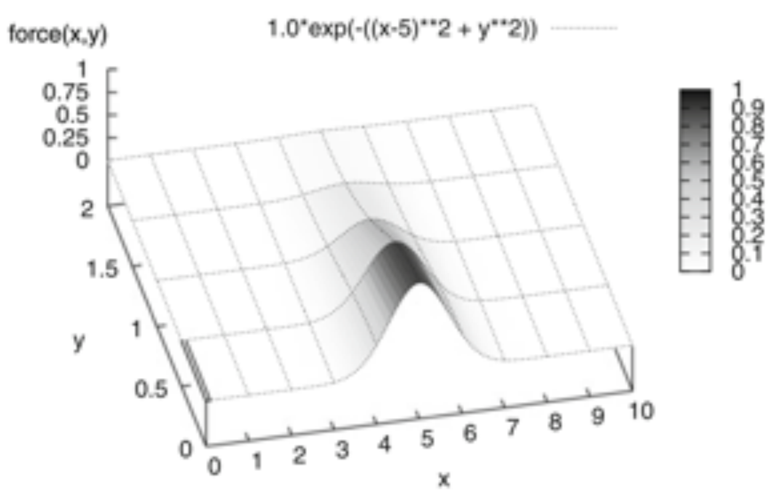
Air blast force distribution over (x, y) postion

Verification

- **Example 2 : Conveyor belt.**

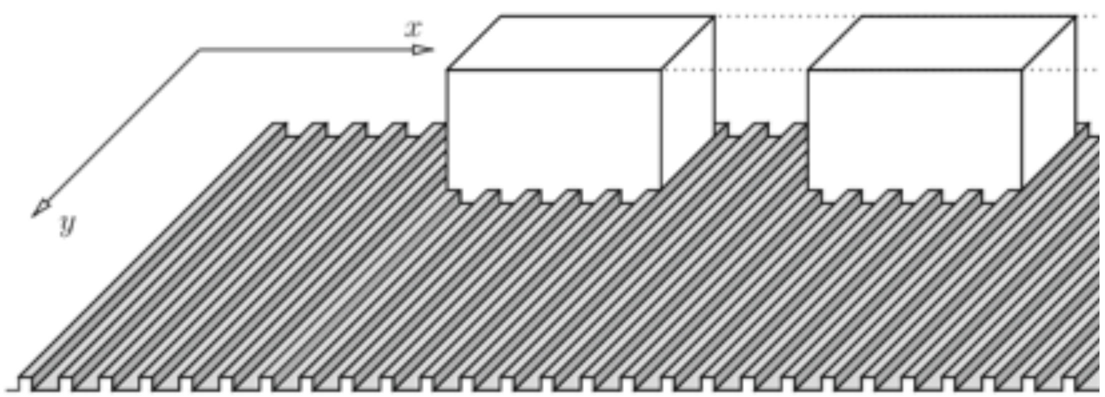


Conveyor belt modelled by parallel automata

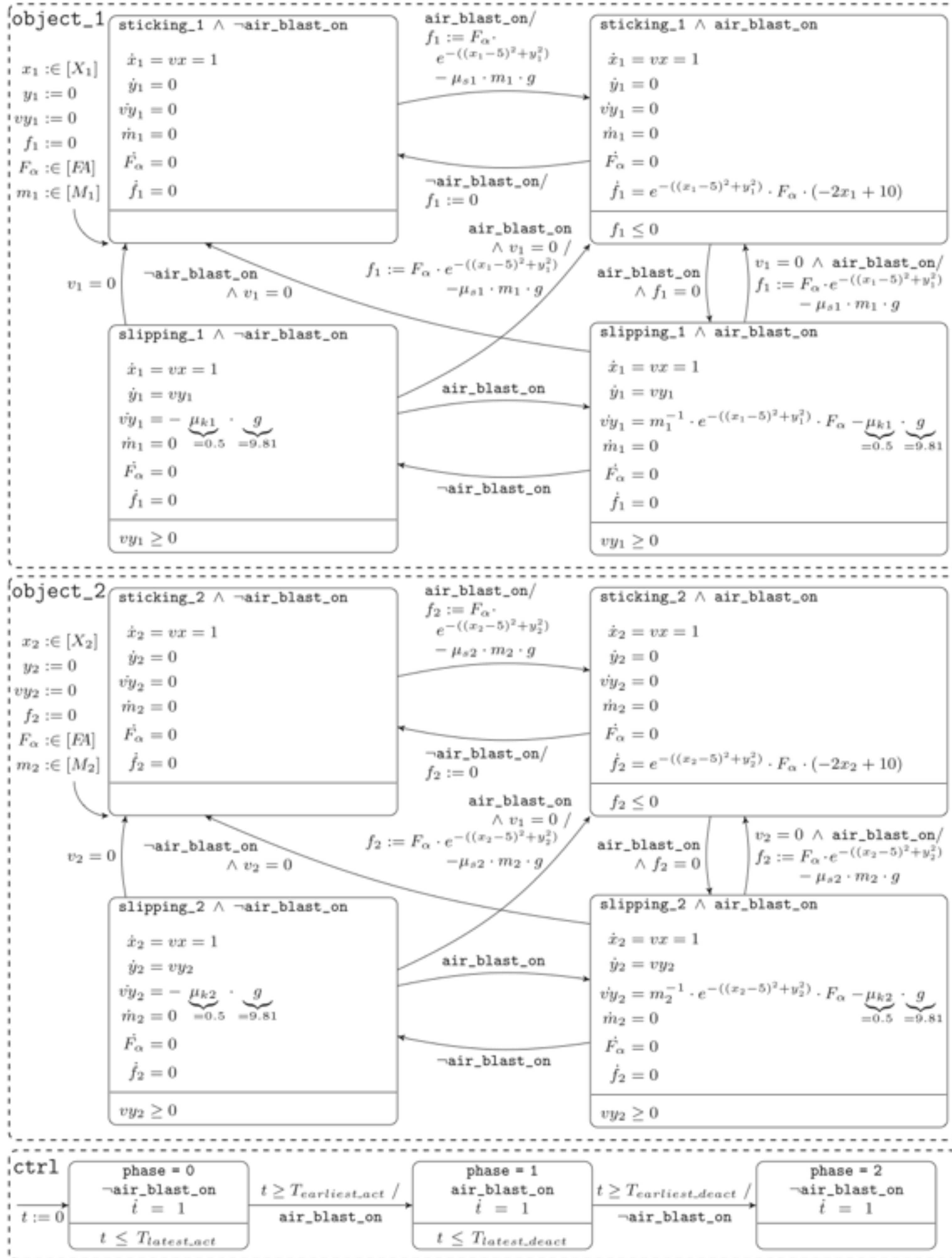


Air blast force distribution over (x, y) postion

- **Example 2 : Convey**



Conveyor belt modelled by parallel automata



UNIVERSITE D'ORLEANS

Conclusion

A. Eggers, N. Ramdani, N.S. Nedialkov, M. Fränzle,
Improving the SAT Modulo ODE Approach to Hybrid Systems Analysis by Combining Different Enclosure Methods,
Software & Systems Modeling 14(1) pp 121-148, 2015

All papers on my web site
http://lune.bourges.univ-orleans.fr/ramdani

Thank you !

Questions ?