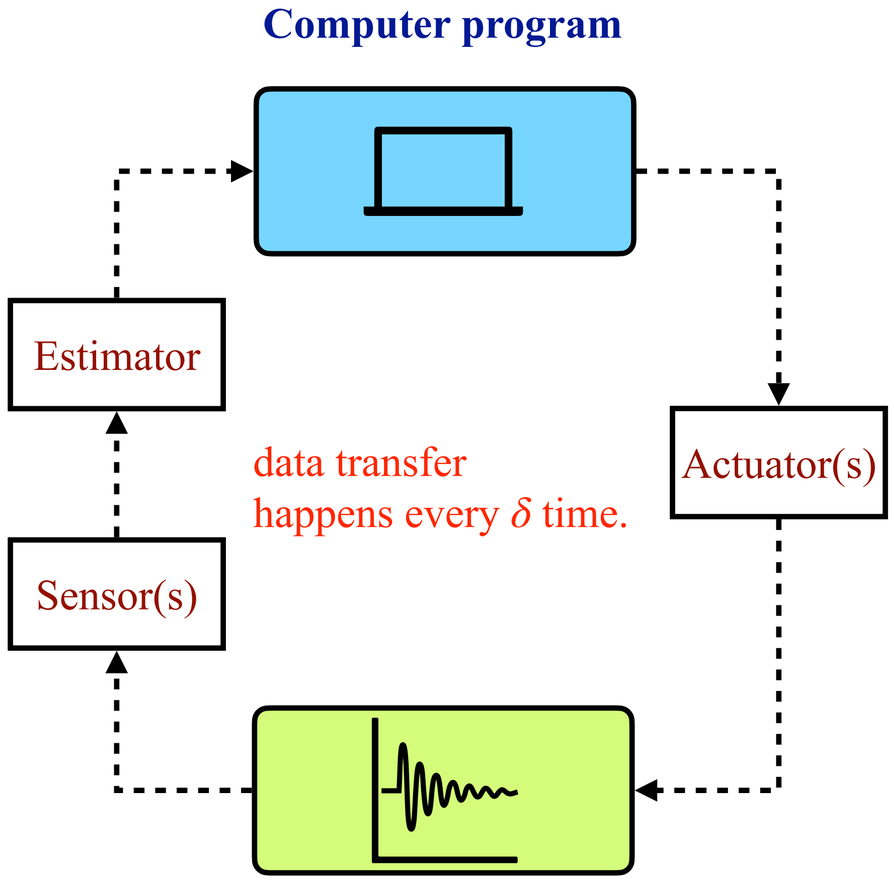
Introduction to *Reachability Analysis* of **Cyber-Physical Systems**

Xin Chen
Assistant Professor, University of Dayton, USA.

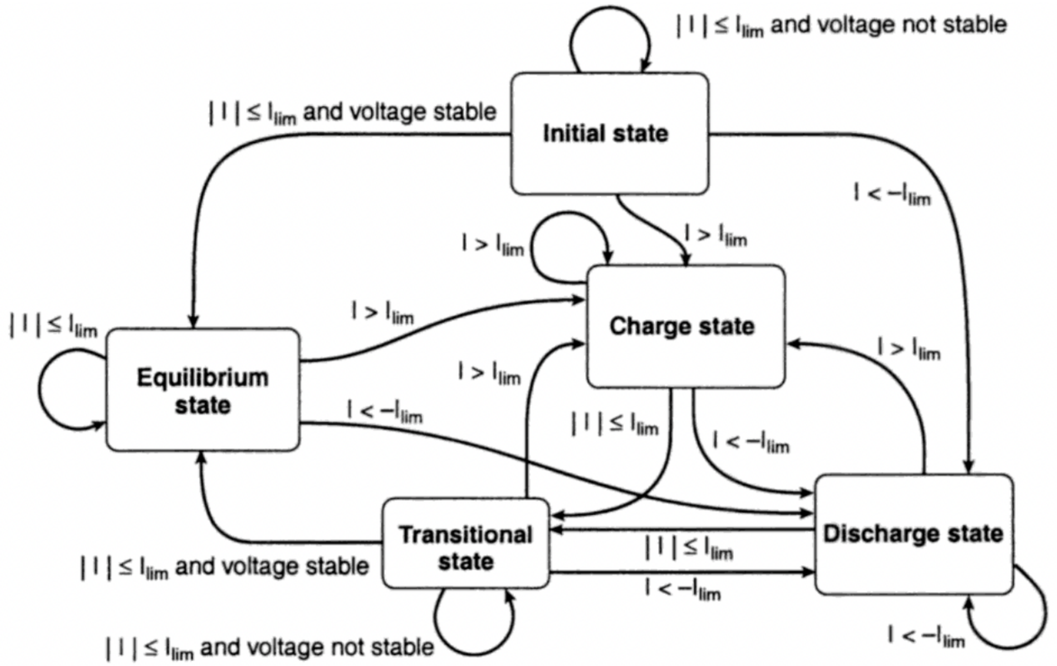
The 18th International Summer School on Trustworthy Software. 2022.

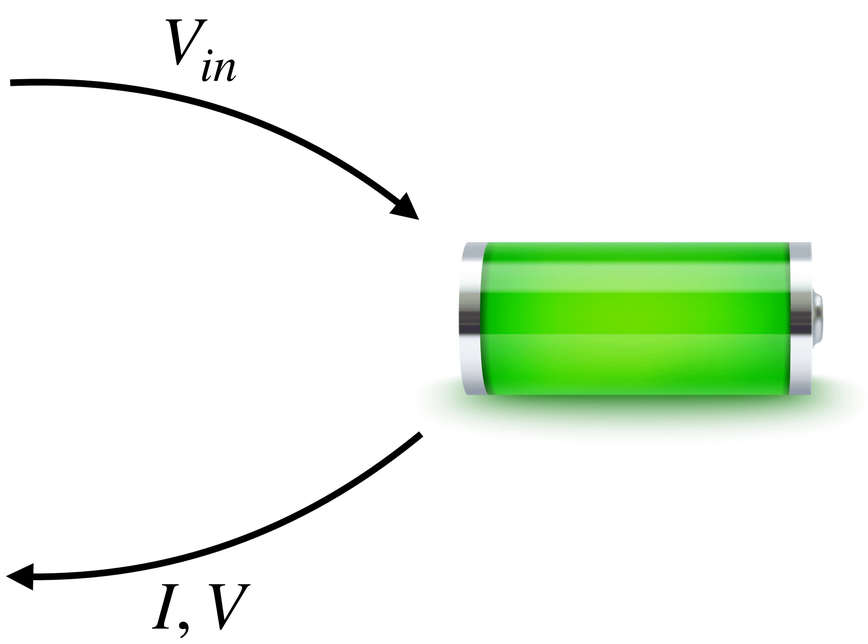
Cyber-Physical System (CPS)



Physical process: change of velocity, voltage, altitude, pitch angle, glucose level, …

CPS can be both simple and complex.

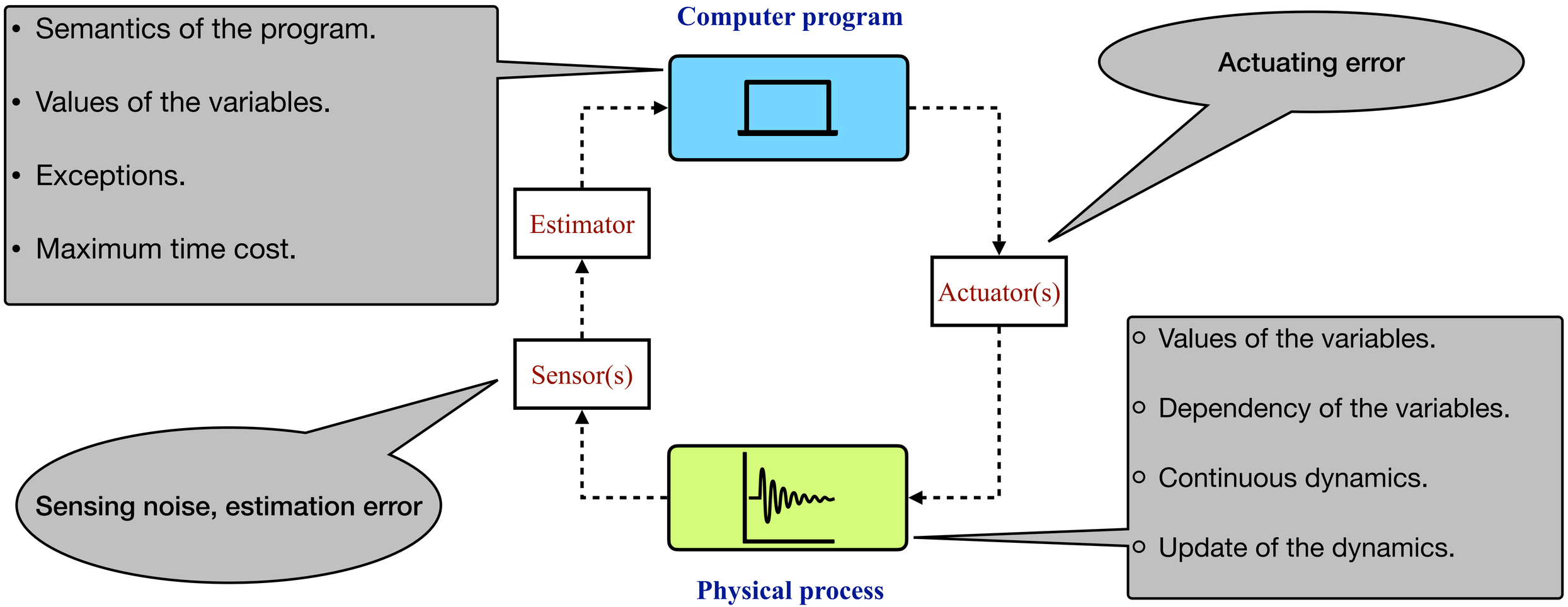




Properties: battery is charged, battery's lifespan is not reduced, battery is not overcharged, …

Henk Jan Bergveld, Wanda S. Kruijt, Peter H. L. Notten. *Battery Management Systems: Design by Modelling.* Springer 2002.

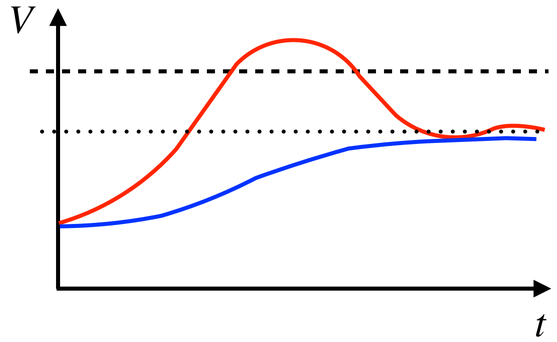
What to describe?

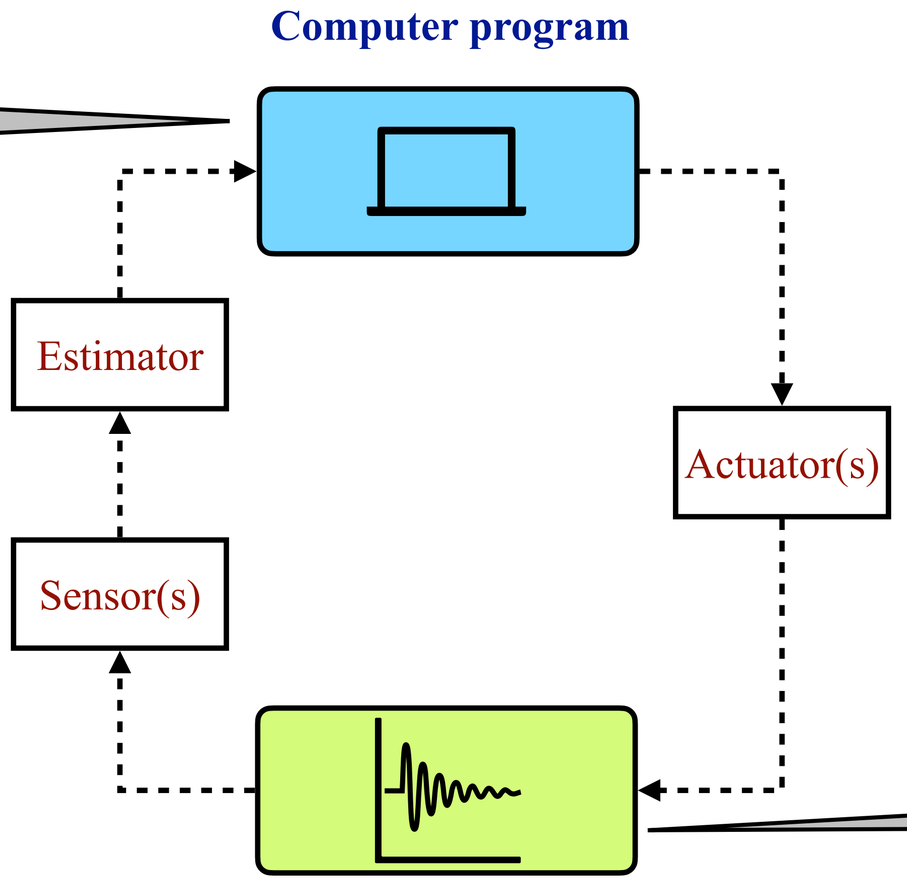


What to verify?

Goal of verification:

The physical process does not have any unsafe behavior (executions) under the control of the program.

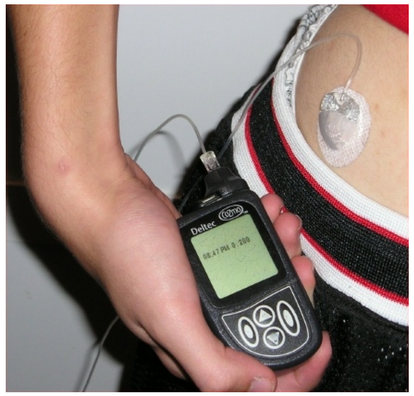


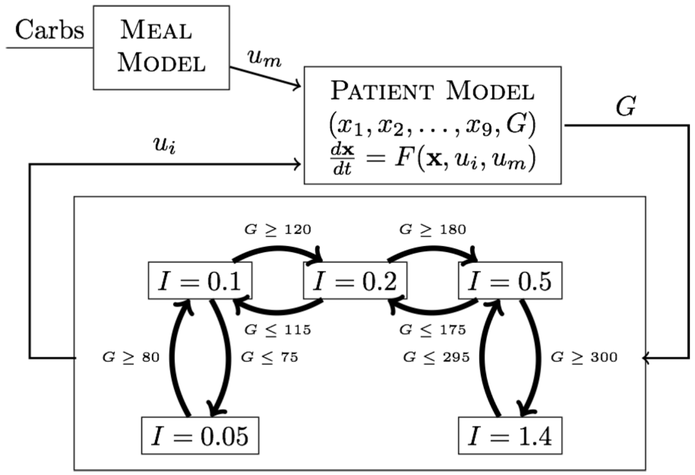


Properties to verify:

- **Safety** - *An unsafe situation should never happen.*
- **Reachability** - *The control target should eventually be reached.*

Example: Artificial Pancreas





9 states, 2 inputs, nonlinear model [Dalla Man et al.]

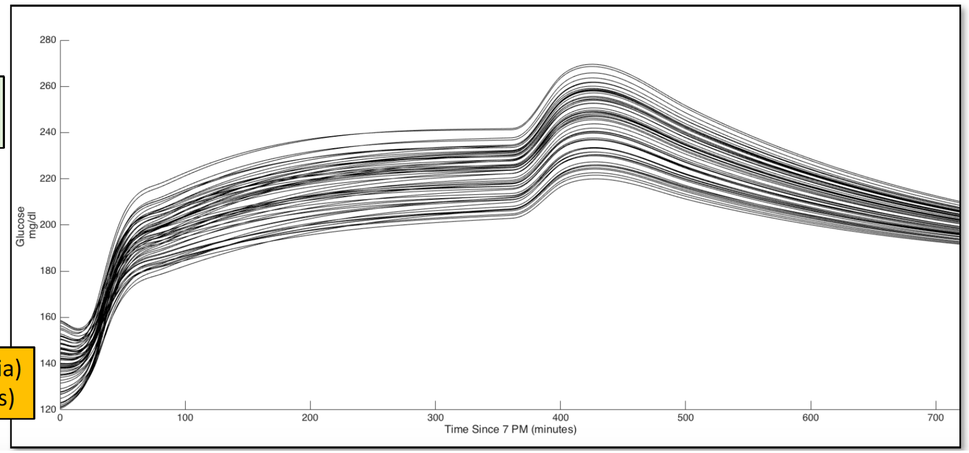
Simple rule-based controller: adjusts insulin infusion based on glucose levels

G_0: [120,160] mg/dl

t = 0
7 PM

t = 720 mins
7 AM

Meal Carbs
[50,90] gms CHO

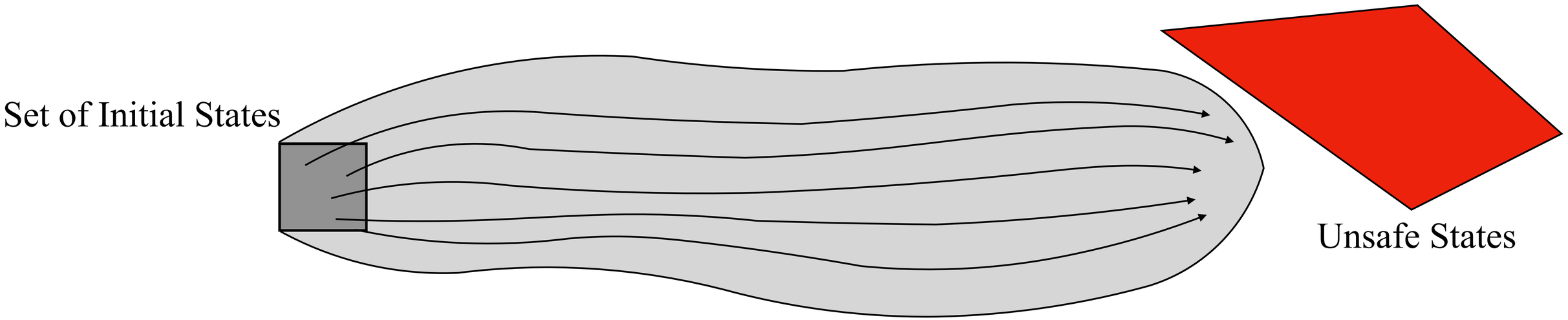


G(t) >= 70 mg/dl (hypoglycemia)
G(t) <= 300 mg/dl (Ketacidosis)

Safety: Ketoacidosis never happens in 24 hours.

Reachability: The glucose level should be regulated to a normal range.

Safety Verification via Reachability Analysis



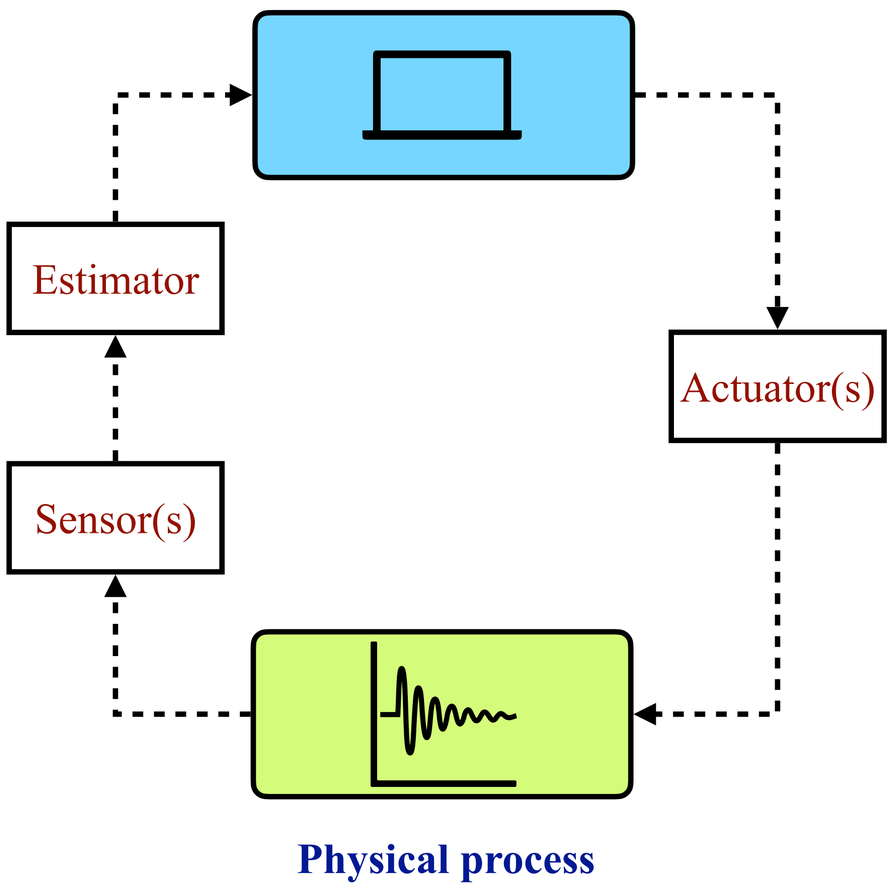
Hardness:

- Hybrid dynamics.
- Infinitely executions.
- Property verification on reachable sets.

Outline

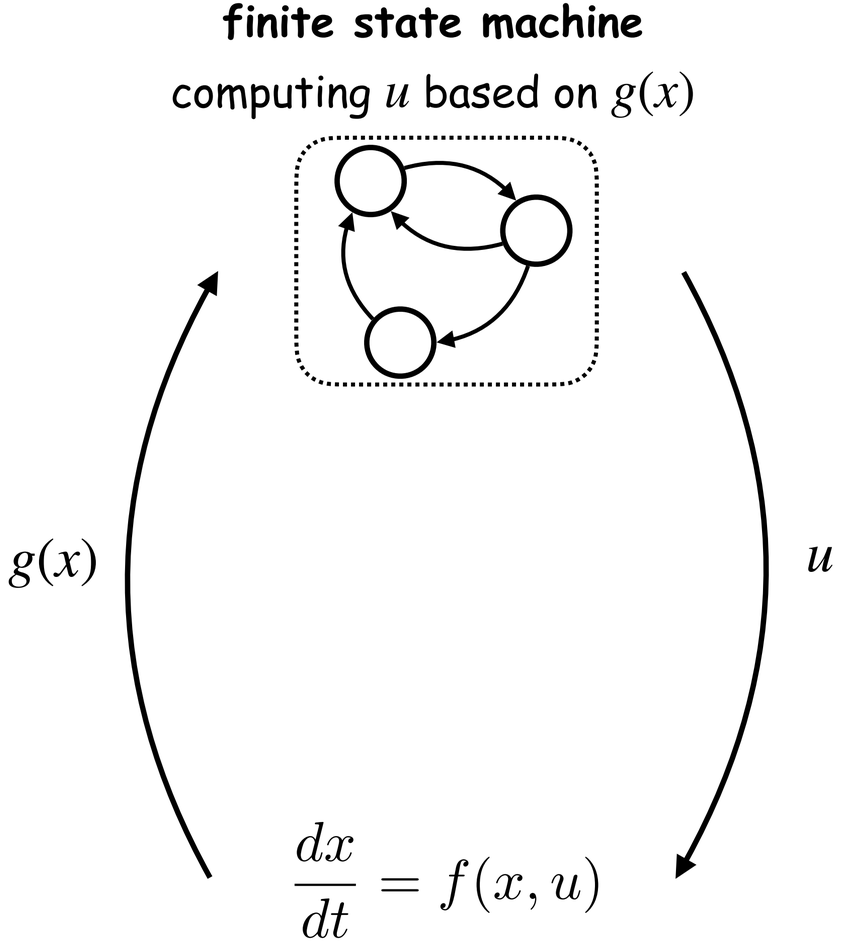
- Formal Model of State Feedback System.
- System Executions.
- Reachable Set Computation by Set Propagation.
- Set Representations.
- Reachable Set Computation.

Formal Model of State Feedback Systems



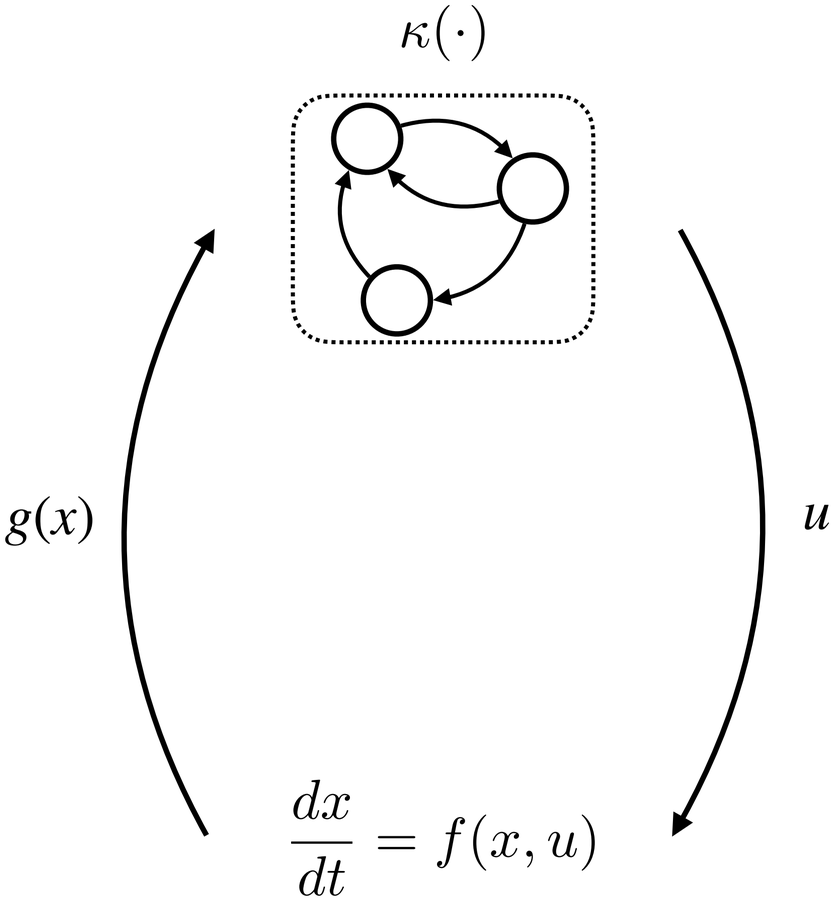
Formalization

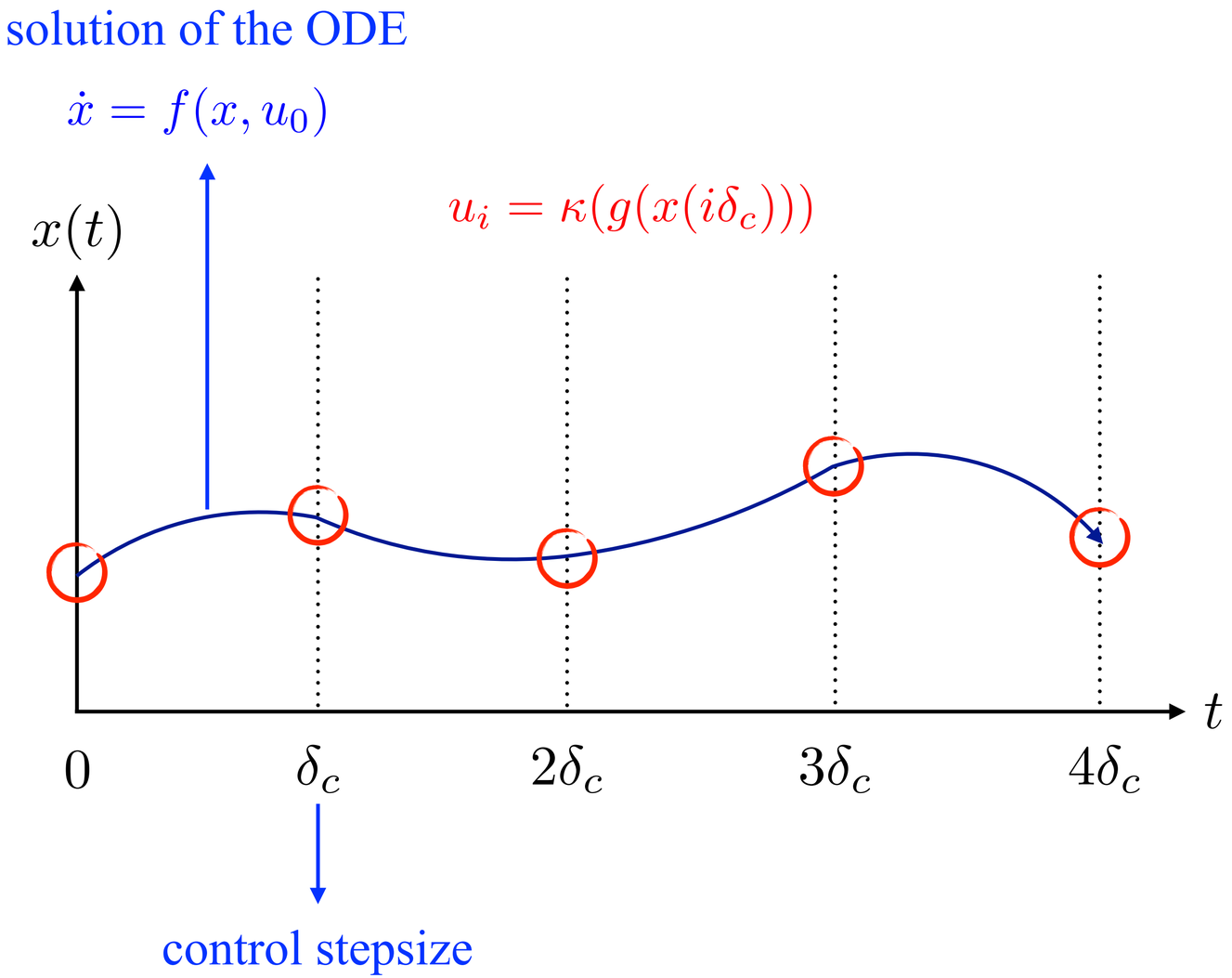
x: state variable(s)
u: control variable(s)



Ordinary Differential Equation (ODE)
of x and u

System Execution

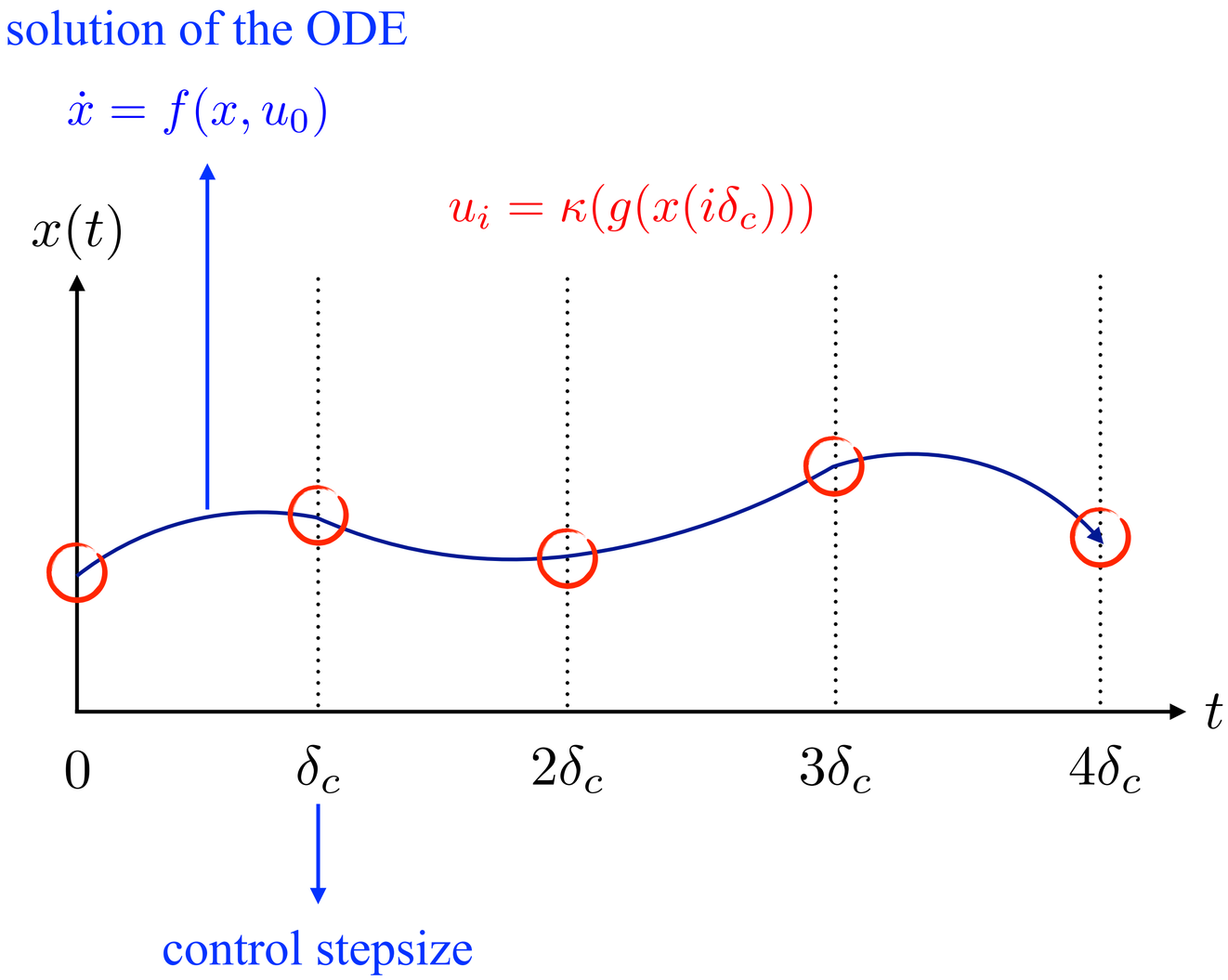




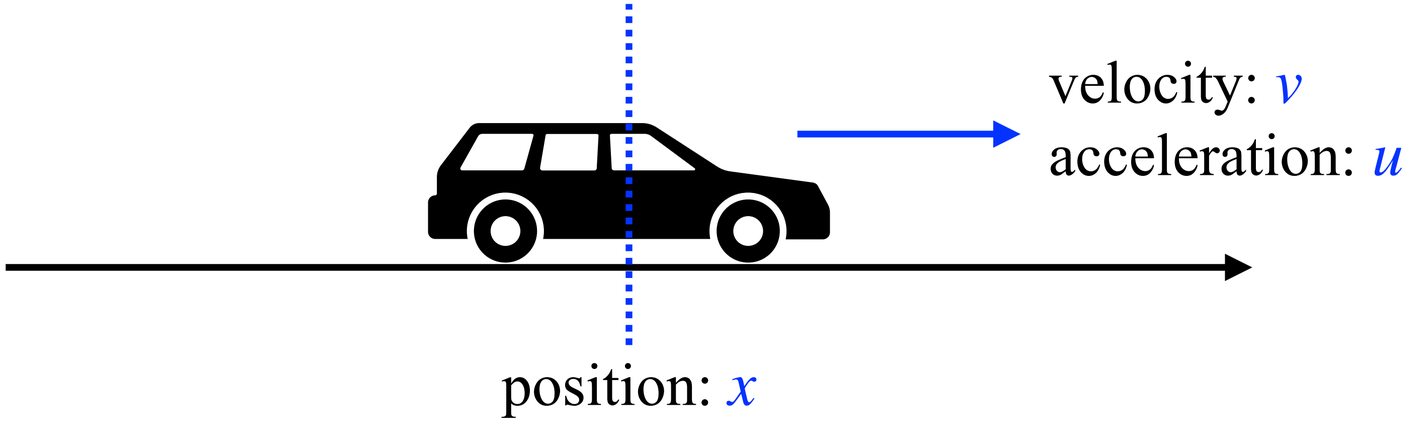
System Execution

Assumptions:

- κ is a function from the state space X to the control space U.
- $f(x, u)$ is at least locally Lipschitz continuous, i.e., the solution is unique w.r.t. any $x \in X$ and $u \in U$.
- Evaluating $\kappa(g(x))$ for any $x \in X$ costs no time.
- Hence, all executions are deterministic when there is no disturbances.

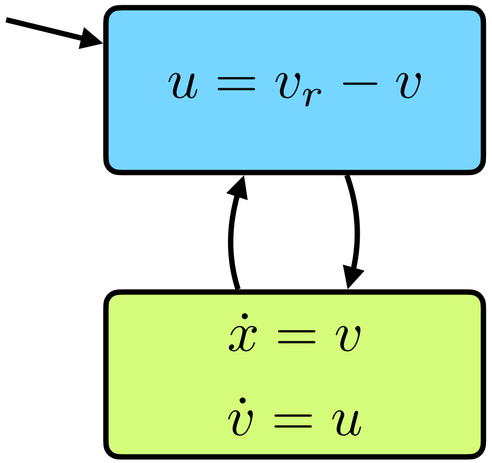


Example: Cruise Control

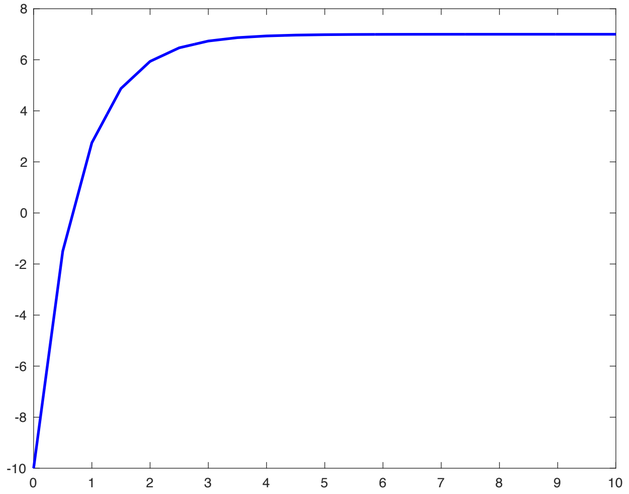


Computer program

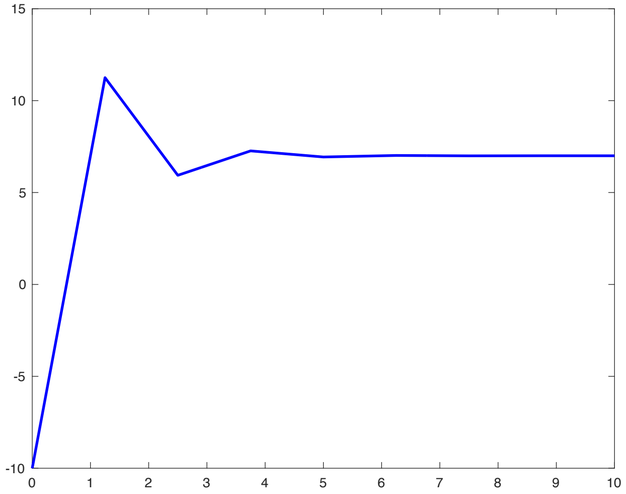
v_r: reference



Physical process



every 0.5 seconds



every 1.25 seconds

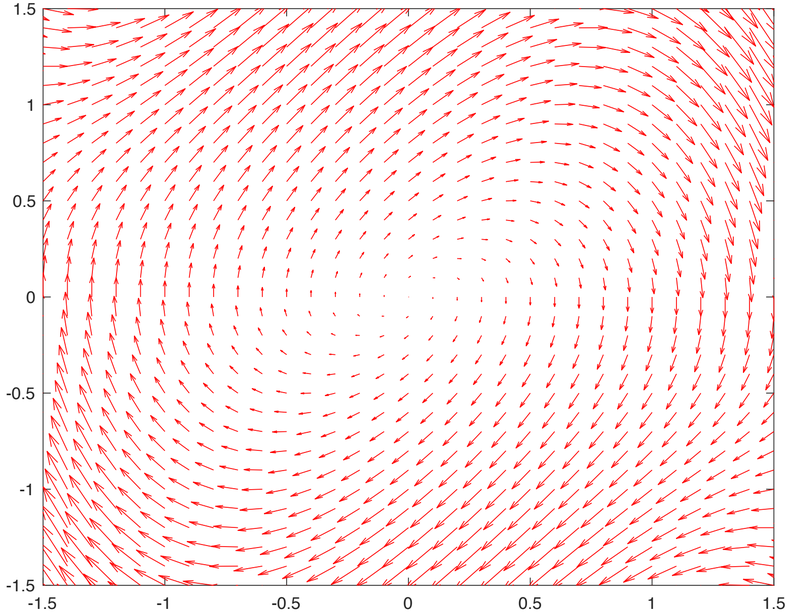
Ordinary Differential Equation

Definition: $\dot{x} = f(x, t)$

x is the state variable(s), t is the time variable.

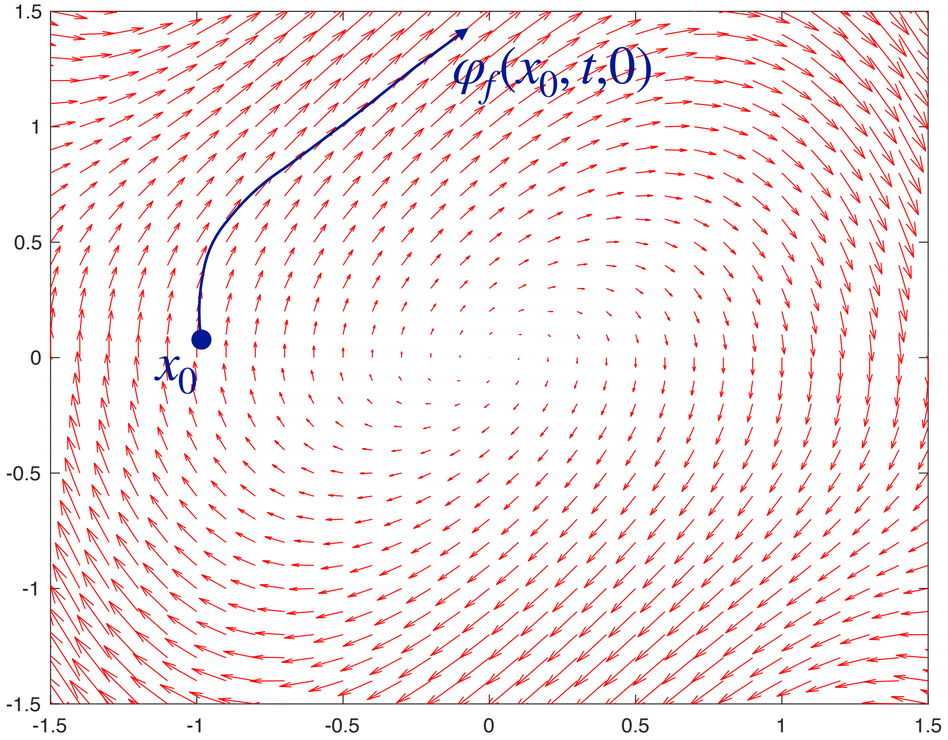
Example: Van der Pol Oscillator

$$\dot{x} = y, \;\; \dot{y} = (1 - x^2)y - x$$



Solution w.r.t. $x(0) = x_0$**:**

$$x(t) = \varphi_f(x_0, t, 0) \text{ and } \frac{d\varphi_f}{dt} = f(x, t).$$



Initial Value Problem (IVP)

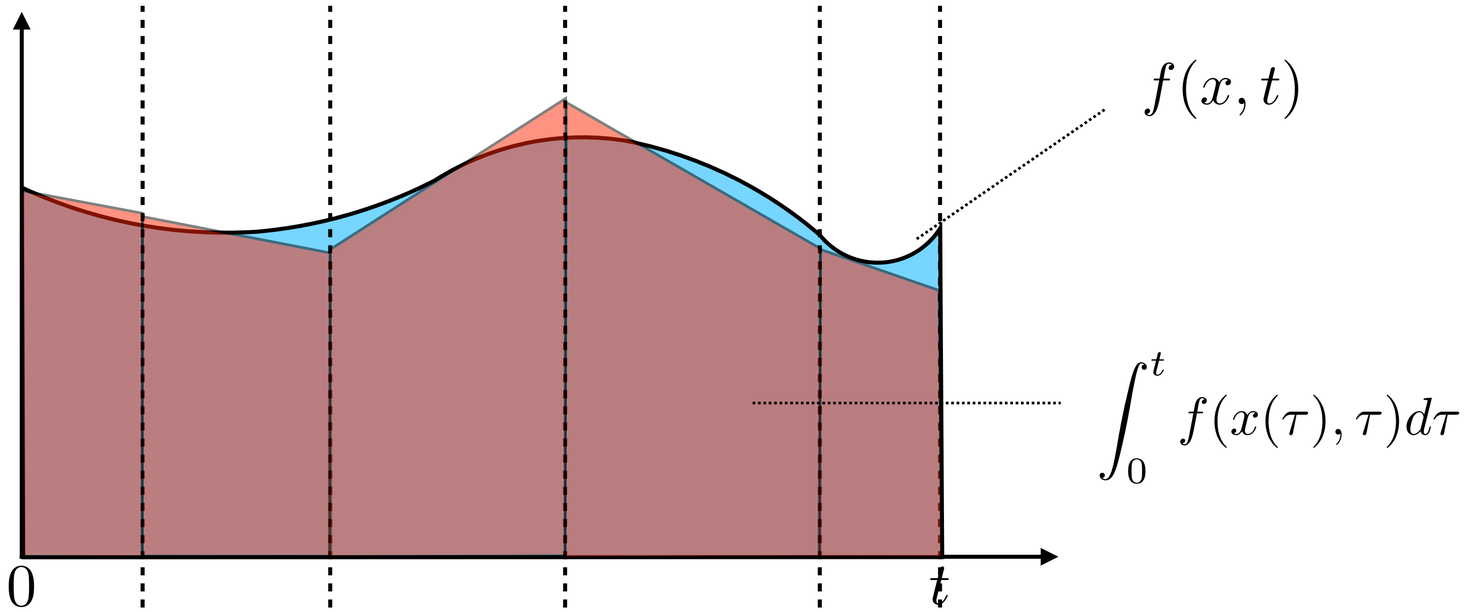
An **IVP** asks to find the solution φ_f to an **ODE** $\dot{x} = f(x, t)$ together with an **initial condition** $x(0) = x_0$.

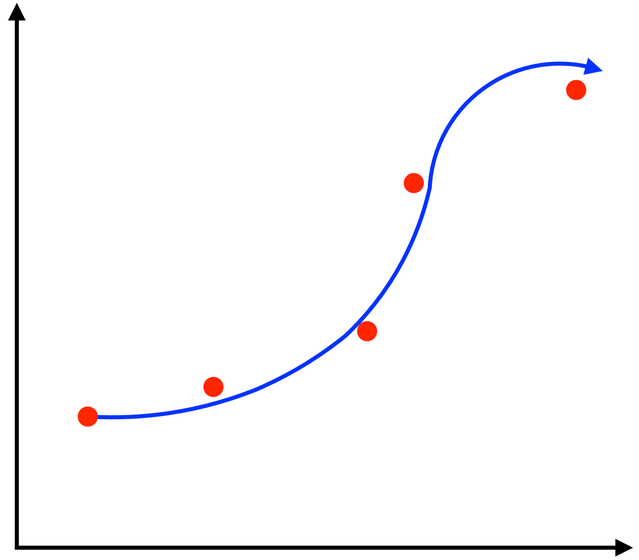
General solution form w.r.t. $x(t_0) = x_0$:

$$\varphi_f(x_0, t, t_0) = x_0 + \int_{t_0}^{t} f(\varphi_f(x_0, \tau, t_0), \tau) d\tau$$

- When the ODE is linear in the state variables, e.g., $\dot{x} = Ax + b$, it **has** closed-form solutions, e.g., $\varphi_f(x_0, t, 0) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}b d\tau$.
- When the ODE is nonlinear in the state variables, it often **does not** have closed-form solutions.
- Nonlinear ODEs are often solved by **numerical** approaches.

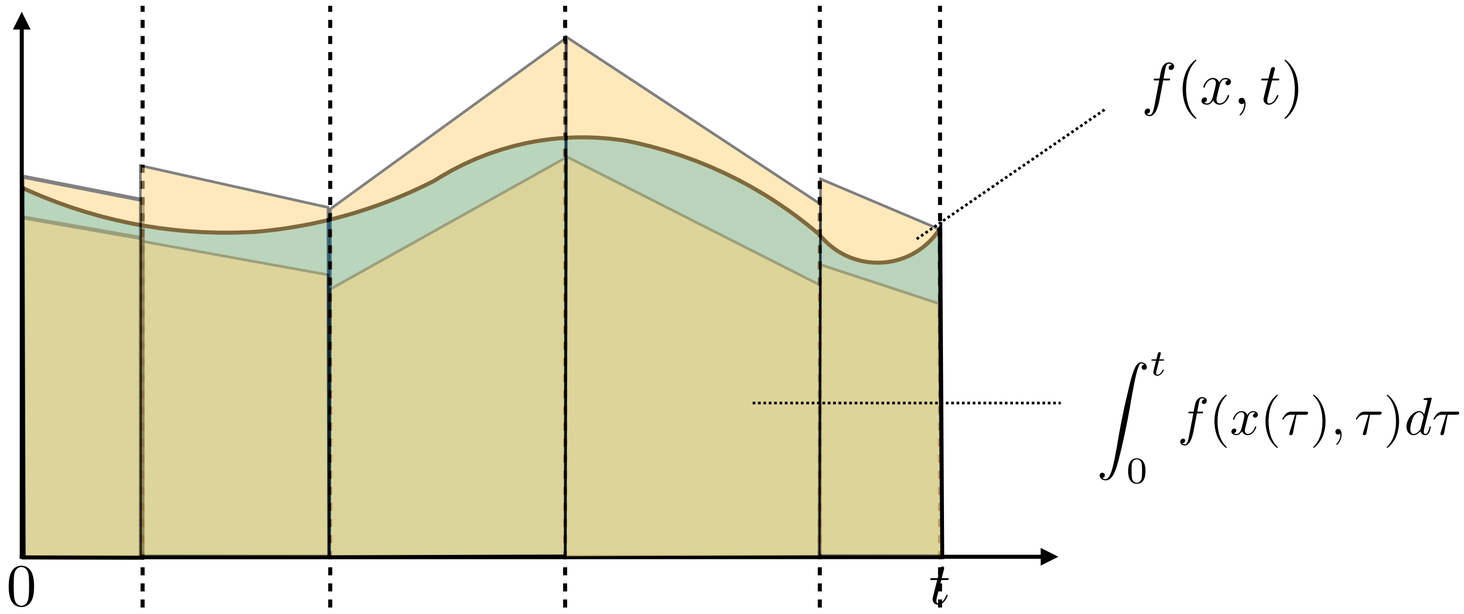
Numerical Integration

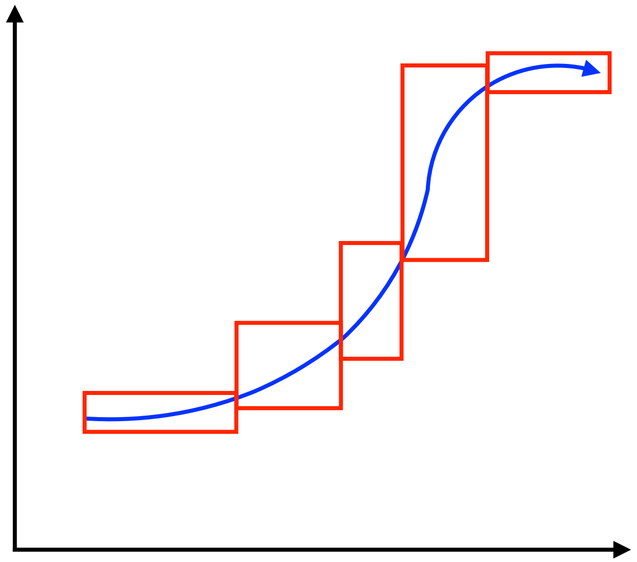




- The result is an approximation to the actual solution at finitely many time instances.
- Methods: Euler method, Taylor method, Runge–Kutta methods, …

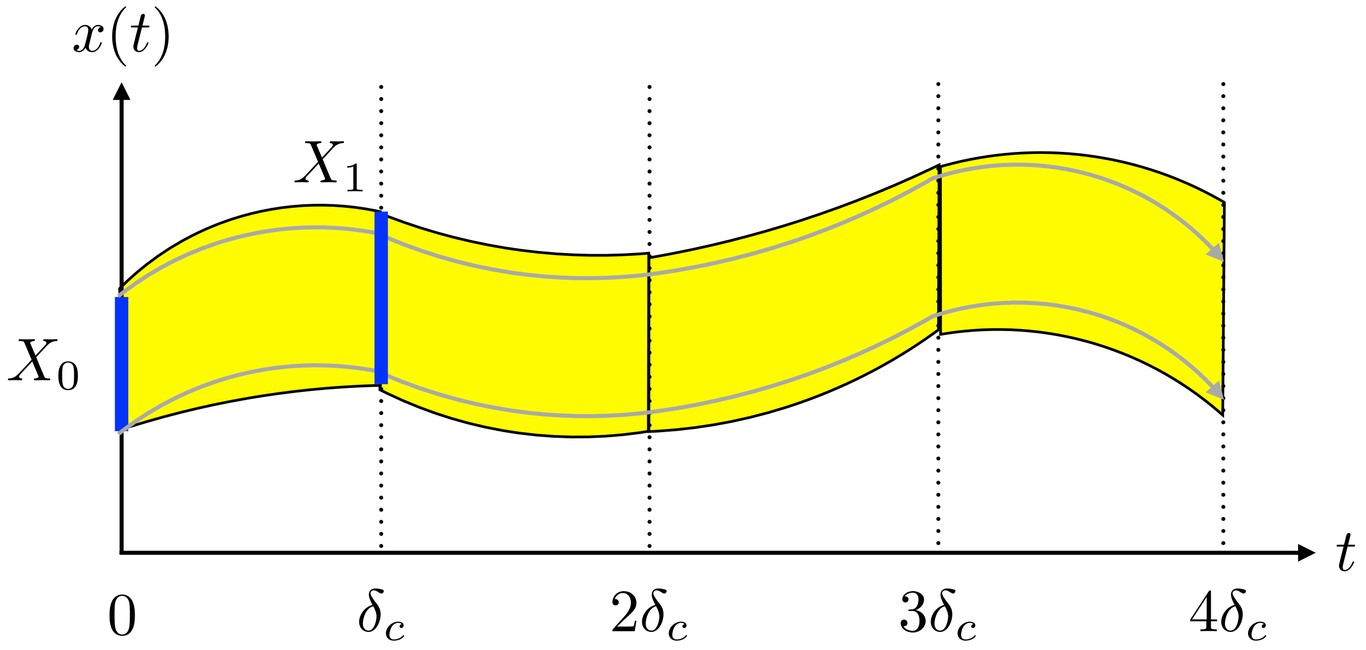
Verified Integration





- The result consists of finitely many sets and is guaranteed to contain the exact solution.
- The method also handles a set of initial states.
- Set representations: intervals, interval Taylor series, Taylor models.

Reachable Set Computation by Set Propagation



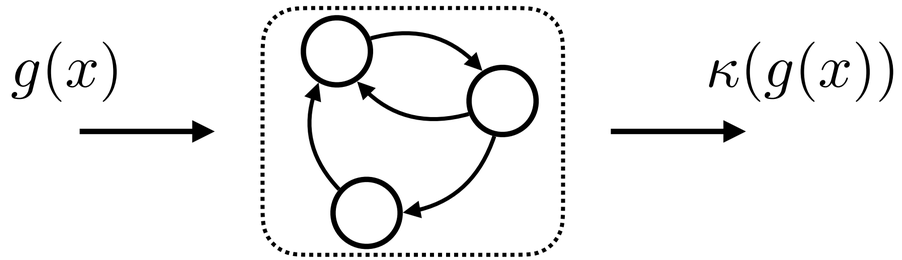
Time-bounded Reachability:

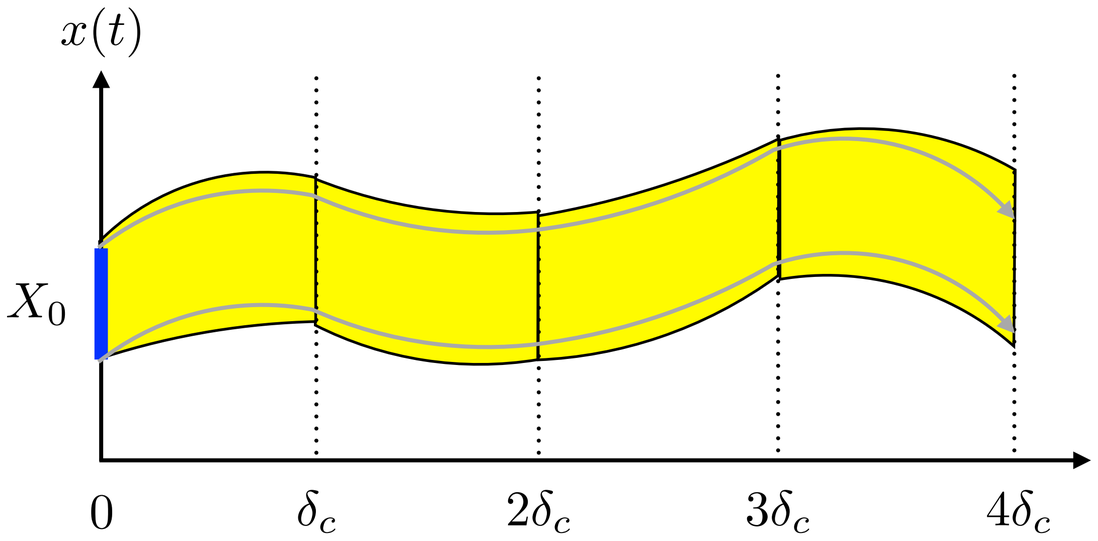
- Compute a set that is guaranteed to contain all executions in the next control step.
- Contract the set to only contain the reachable states at the end of the step.
- Repeat the above two steps.

Xin Chen and Sriram Sankaranarayanan. *Reachability Analysis for Cyber-Physical Systems: Are We There Yet?.* In NASA Formal Methods (NFM), volume 13260 of LNCS, pp. 109–130. Springer, 2022.

Necessary Operations in Set Propagation

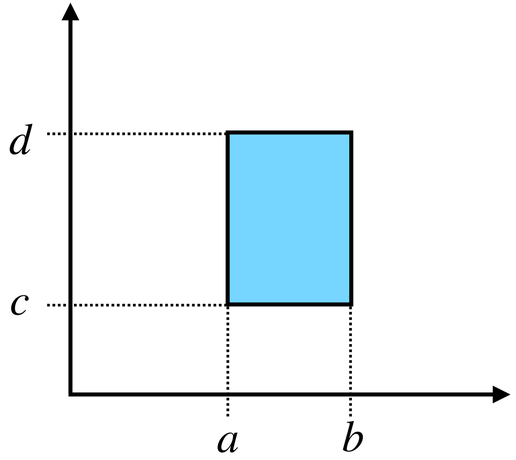
- Verified integration of ODEs. (Already know.)
- Switching continuous dynamics.
- Computing the output range of the control program.
- Intersecting reachable sets with unsafe set.
- Verifying the inclusion of a reachable set in the target set.

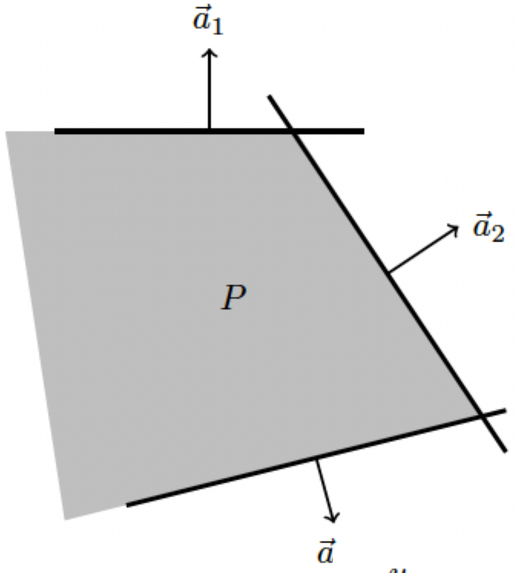




Representing Infinitely Many States

Interval: $[a, b] \times [c, d]$

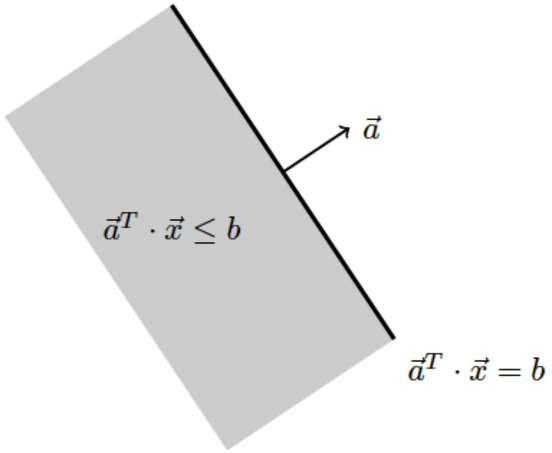




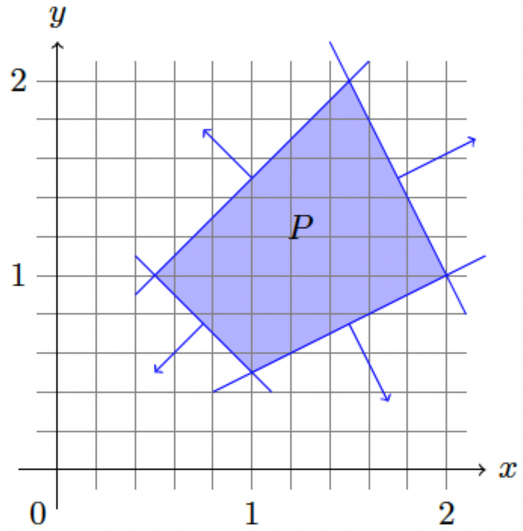
Polyhedron:

$$(\vec{a}_1, \vec{a}_2, \vec{a}_3)^T \vec{x} \leq (b_1, b_2, b_3)^T$$

Halfspace:



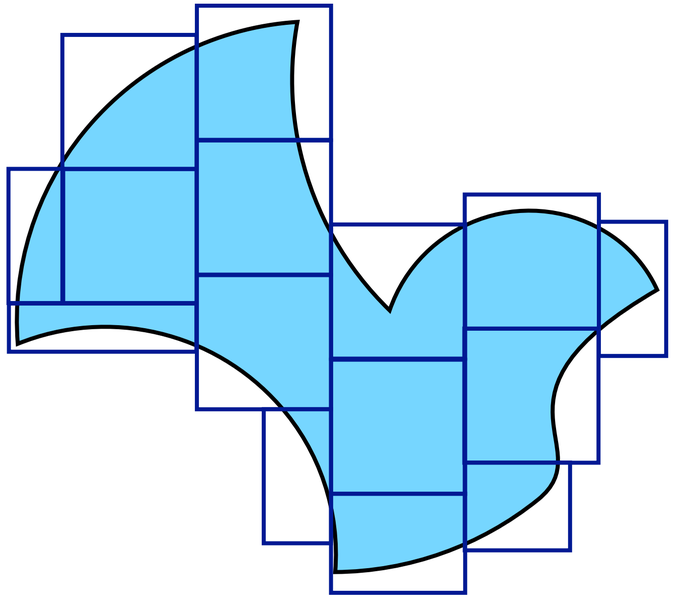
Example:



$$\begin{pmatrix} 2 & 1 \\ 1 & -2 \\ -1 & -1 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} 5 \\ 0 \\ -1.5 \\ 0.5 \end{pmatrix}$$

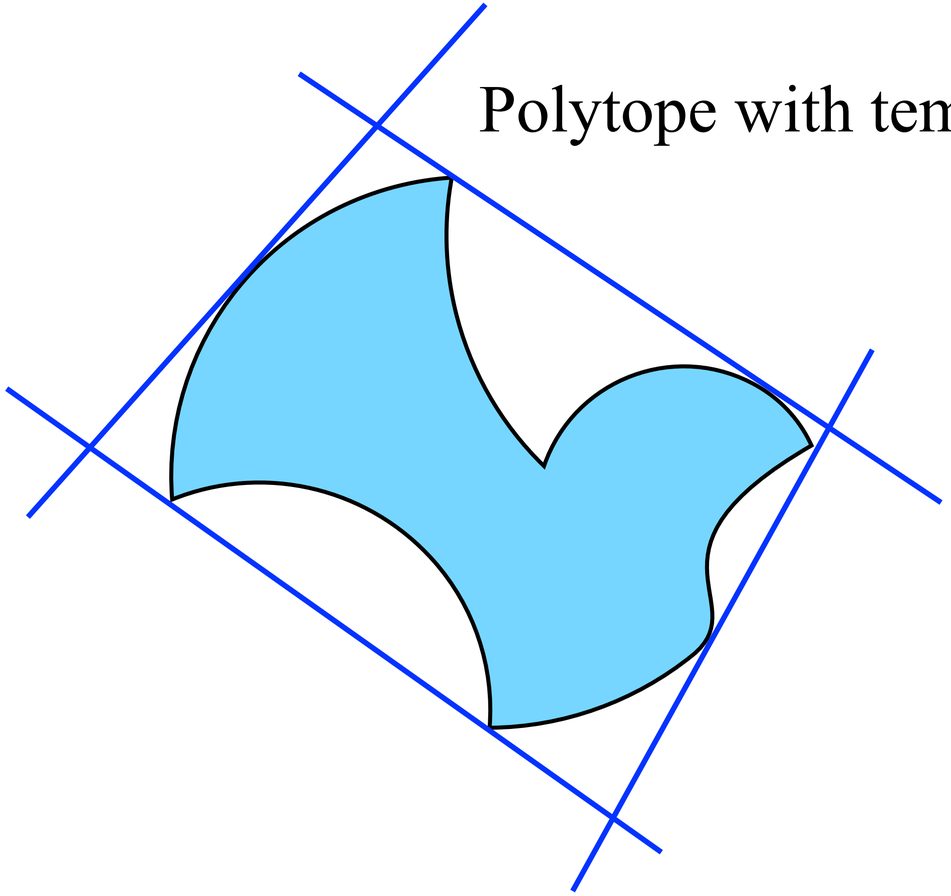
Overapproximate Representations

Set of intervals



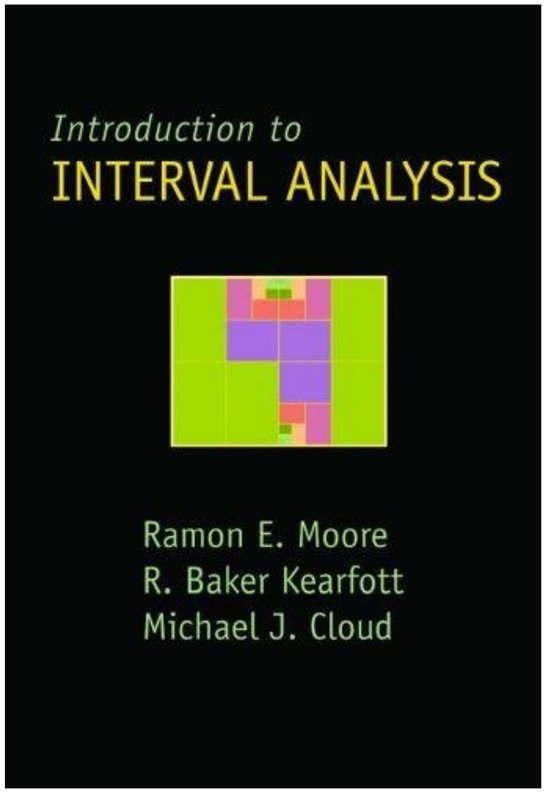
exponential in the state space dimension

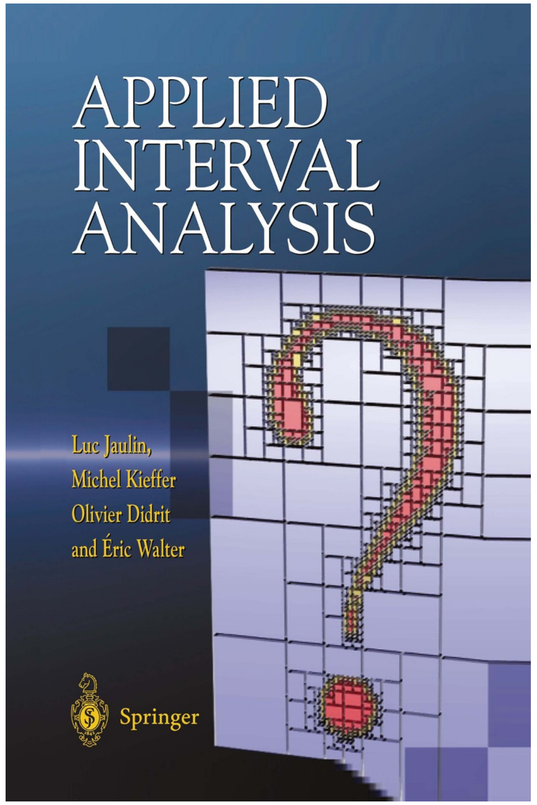
Polytope with template



cannot accurately overapproximate nonconvex sets

Interval Analysis





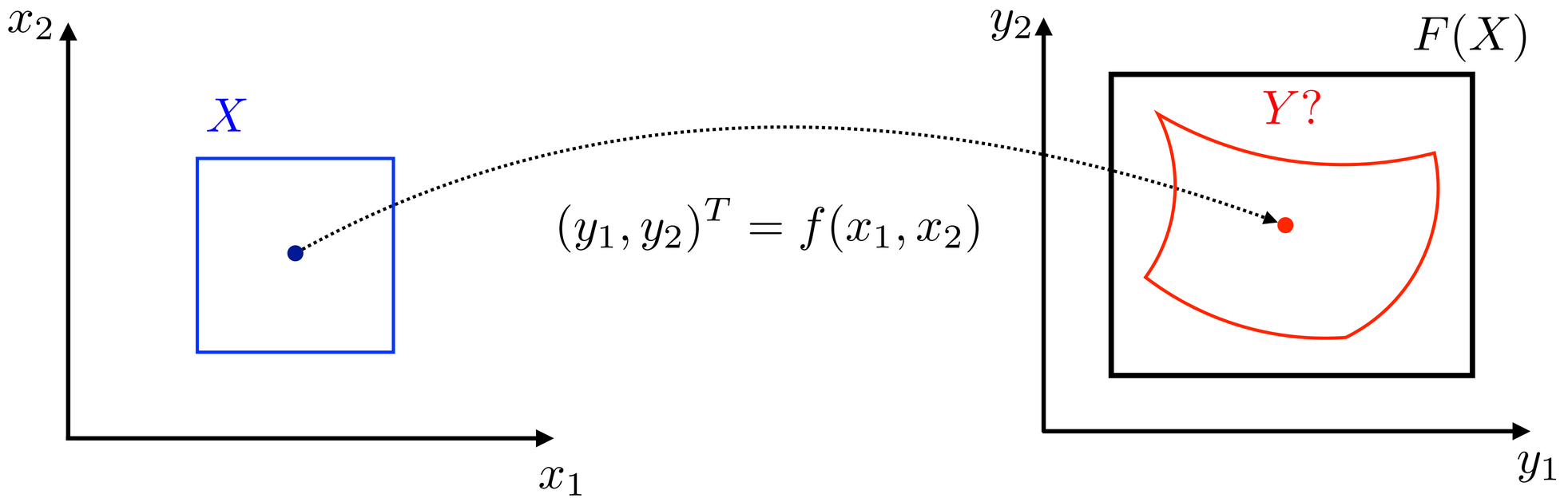
Basic Operations

$$
\begin{array}{lcccl}
\text{Addition:} & [a,b] & + & [c,d] & = [a+c, b+d] \\
\text{Subtraction:} & [a,b] & - & [c,d] & = [a-d, b-c] \\
\text{Multiplication:} & [a,b] & \cdot & [c,d] & = [\min\{a\cdot c, a\cdot d, b\cdot c, b\cdot d\}, \\
 & & & & \quad \max\{a\cdot c, a\cdot d, b\cdot c, b\cdot d\}] \\
\text{Division:} & [a,b] & / & [c,d] & = [a,b]\cdot[1/d, 1/c]
\end{array}
$$

$$
\begin{array}{lrcl}
\text{Exponential:} & \exp([a,b]) & = & [\exp(a), \exp(b)] \\
\text{Logarithm:} & \log_c([a,b]) & = & [\log_c(a), \log_c(b)] \\
\text{Square root:} & \sqrt{[a,b]} & = & [\sqrt{a}, \sqrt{b}]
\end{array}
$$

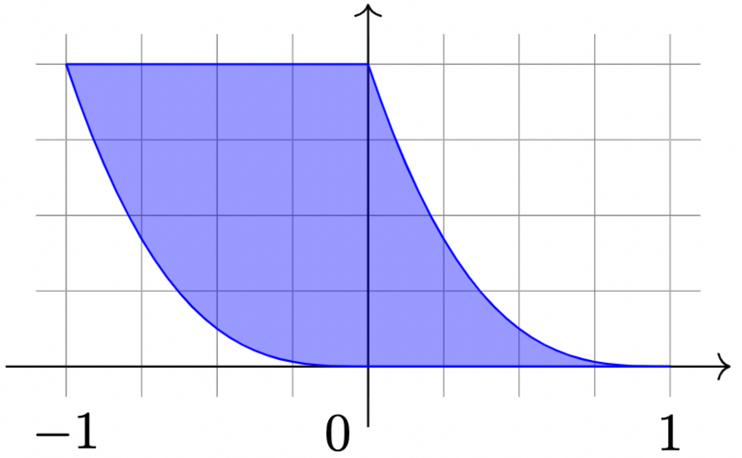
Interval Extension

F is an interval extension of f



Example

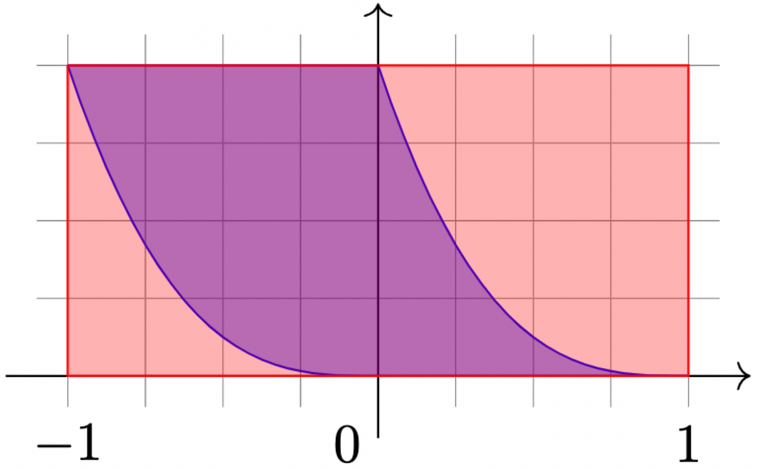
$$f(x_1, x_2) = \begin{pmatrix} x_1 - x_2 \\ x_2^3 \end{pmatrix}$$



$x_1 \in [0, 1],\ x_2 \in [0, 1]$

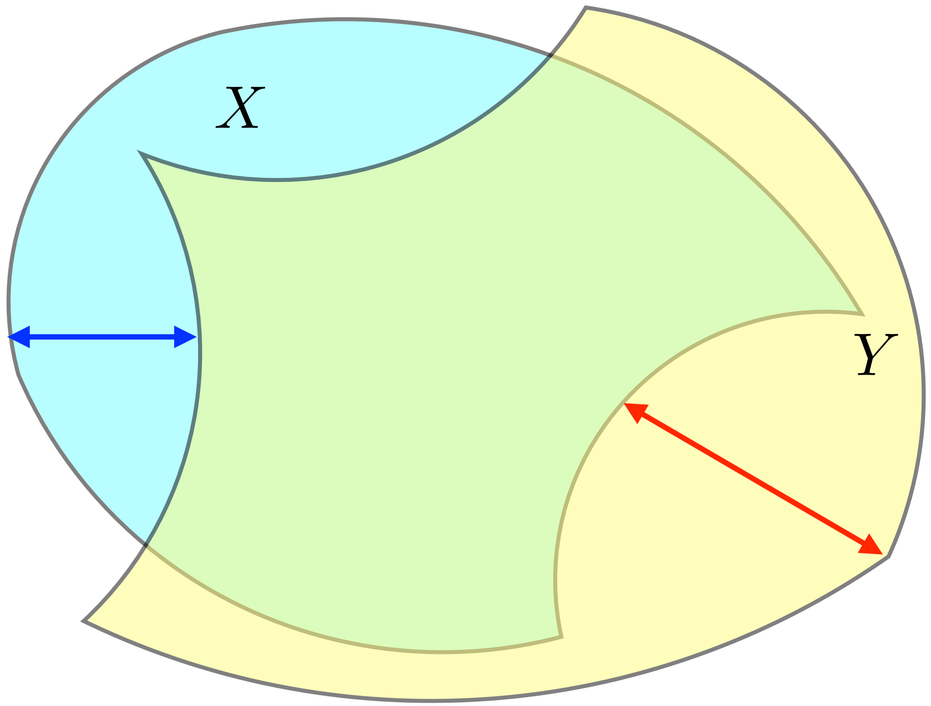
Interval extension:

$$F(X_1, X_2) = \begin{pmatrix} X_1 - X_2 \\ X_2^3 \end{pmatrix}$$



Overapproximation Error (Overestimation)

Hausdorff distance: $d_H(X,Y) = \sup\{\sup_{\vec{x}\in X}\inf_{\vec{y}\in Y}\|\vec{x}-\vec{y}\|, \sup_{\vec{y}\in Y}\inf_{\vec{x}\in X}\|\vec{x}-\vec{y}\|\}.$



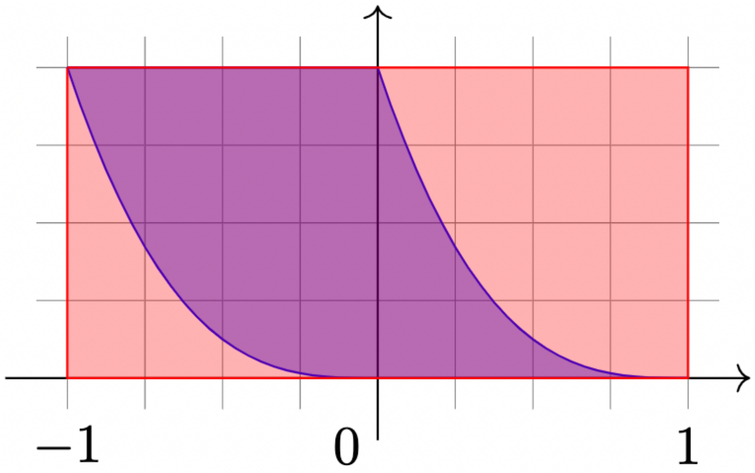
When the Hausdorff distance is 0 between X and Y, then X = Y.

However, we usually measure the overestimation only based on the size of the overapproximation.

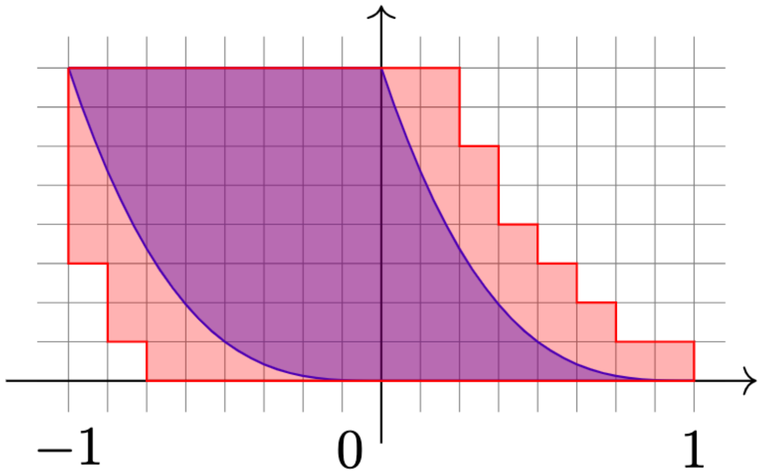
Overapproximation Refinement

$$F(X_1, X_2) = \begin{pmatrix} X_1 - X_2 \\ X_2^3 \end{pmatrix}$$

Interval evaluation from
$X_1 = [0,1]$, $X_2 = [0,1]$



Union of the interval evaluations from
an 8-subdivision of $X_1 = [0,1]$, $X_2 = [0,1]$



Computing Reachable Set under ODE dynamics

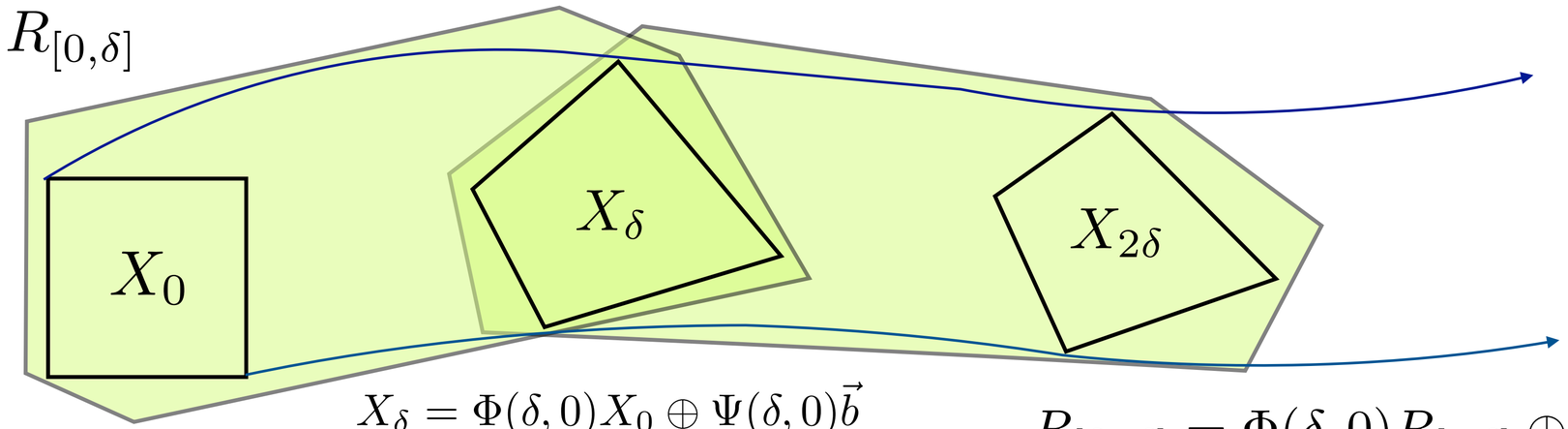
- If the ODE is linear in the state variables, we may directly use the closed-form solution. **[Girard 2005], [Le Guernic et al. 2009], [Frehse et al. 2011]**
- If the ODE is nonlinear, the reachable sets are computed based on the local linearizations. **[Althoff et al. 2008], [Dang et al. 2010], [Althoff 2013]**
- Verified integration methods. **[Berz et al. 1998], [Nedialkov et al. 1999], [Neher et al. 2007]**
- Only numerical arithmetic is involved in the above methods.

Example: Reachable Set Computation for Linear ODEs

ODE: $\dot{\vec{x}} = A\vec{x} + \vec{b}$

Solution: $\varphi_f(\vec{x}_0, t, 0) = e^{At}\vec{x}_0 + \int_0^t e^{A(t-\tau)} d\tau\, \vec{b} = \Phi(t,0)\vec{x}_0 + \Psi(t,0)\vec{b}$

Properties: $\Phi(t_1, t_3) = \Phi(t_1, t_2)\Phi(t_2, t_3)$ $\Psi(t_1, t_3) = \Psi(t_1, t_2) + \Phi(t_1, t_2)\Psi(t_2, t_3)$



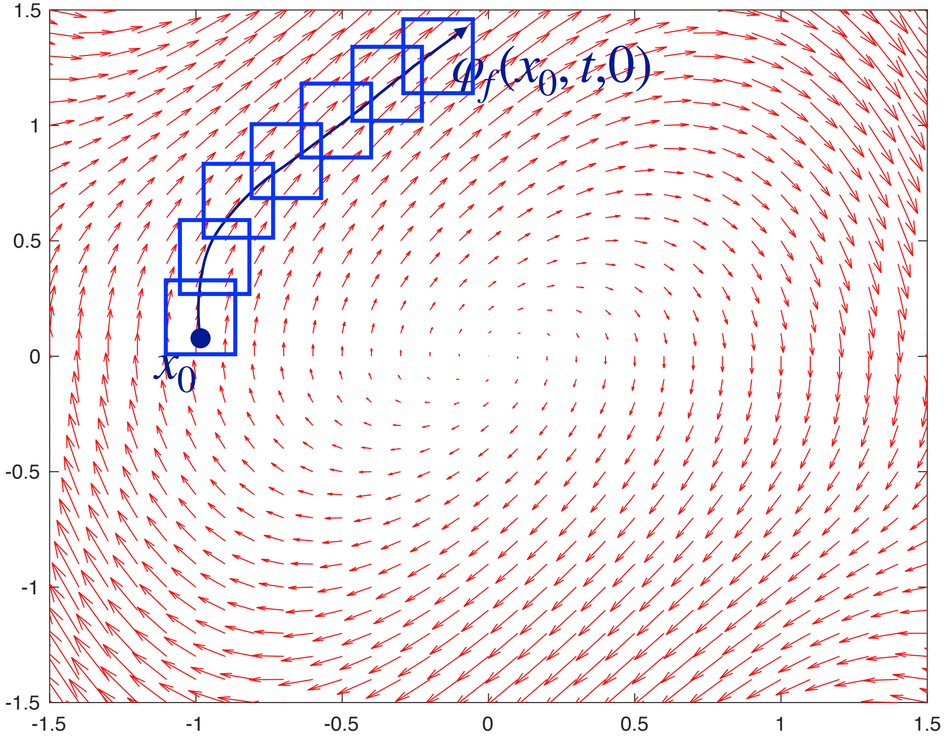
$X_\delta = \Phi(\delta, 0)X_0 \oplus \Psi(\delta, 0)\vec{b}$

$R_{[\delta,2\delta]} = \Phi(\delta, 0)R_{[0,\delta]} \oplus B,\ \{\Psi(\tau, 0)\vec{b} \,|\, \tau \in [0, \delta]\} \subseteq B$

Such a scheme does not work on nonlinear ODEs …

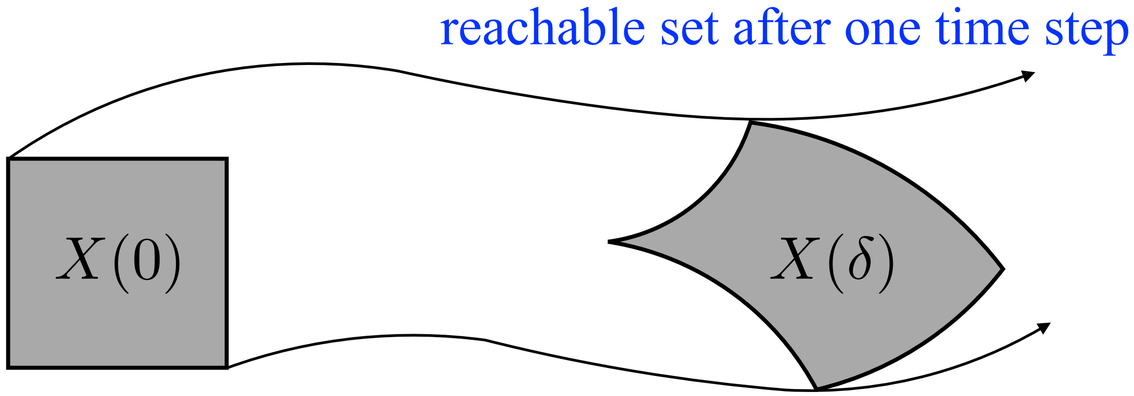
Verified/Validated Integration

Verified integration is originally proposed to solve **IVP**.



- **Result 1:** a validated set that contains the actual solution at a particular time.
- **Result 2:** a group of validated sets each of which contains the actual solution in a time interval.
- **Set Representations:** intervals, interval Taylor series.
- **Initial Condition:** a single state or an interval.

Framework of Verified Integration



- We do not know $X(\delta)$, but is able to compute the following Taylor approximation at $X(0)$:

$$X(t) \approx \tilde{X}(t) = X(0) + f(X(0))t + Y_2 t^2 + \cdots + Y_k t^k$$

such that $Y_i = \frac{1}{i}\frac{\partial f^{[i-1]}}{\partial x} f(X(0))$ is an interval. The above expression is called the **Interval Taylor Series (ITS)**.

- Evaluating a verified remainder I such that $X(t) \subseteq \tilde{X}(t) + I$ for all $t \in [0,\delta]$.

N. S. Nedialkov, K. R. Jackson, G. F. Corliss. *Validated Solutions of Initial Value Problems for Ordinary Differential Equations.* Applied Mathematics and Computation, Volume 105, Issue 1, pp. 21-68. 1999.

Remainder Evaluation

Picard Operation: $$\mathcal{P}(x(t)) = x(t_0) + \int_{t_0}^{t} f(x(\tau), \tau) d\tau$$

The solution to the IVP: $\dot{x} = f(x, t)$, $x(t_0) = x_0$ is a **fixed point** of the Picard operation:

$$\varphi_f(x_0, t, t_0) = x_0 + \int_{t_0}^{t} f(\varphi_f(x_0, \tau, t_0), \tau) d\tau$$

Banach Fixed-Point Theorem:
Let (X, d) be a nonempty complete metric space with a contraction mapping $\varphi : X \to X$, then φ has a unique fixed point in X.

Remainder Evaluation

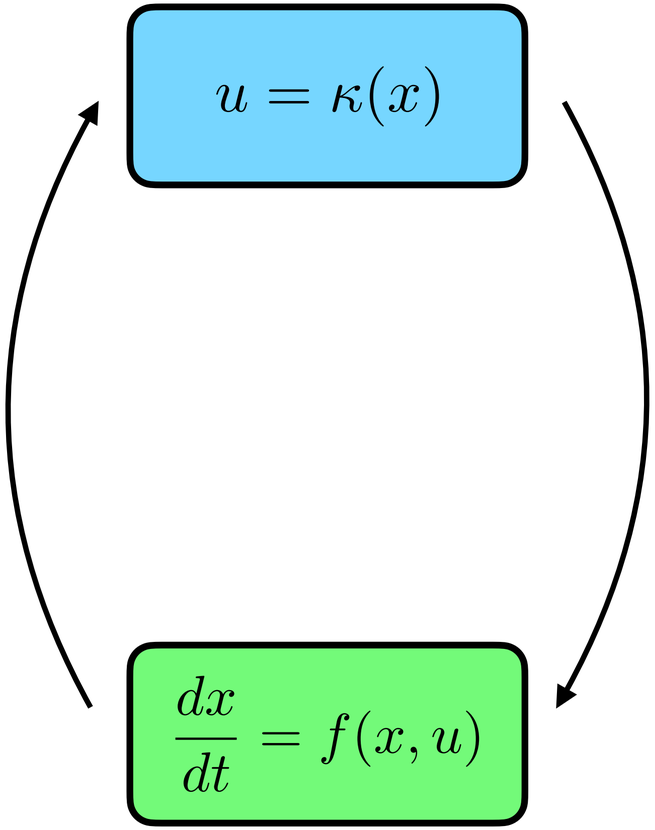
1. Compute an interval estimation I, and check if the Picard operation contracts on $\tilde{X}(t) + I$.

2. If so, go to the next step, otherwise enlarge I and check again.

3. Repeatedly refine the interval I using Picard operation until there is no big contraction.

Question: Are the interval coefficients in $\tilde{X}(t)$ also contracted by Picard operation?

Reachable Set Computation for Simple State Feedback Systems



1. $X_i = X_0$.
2. Computing the set $U_i = \{\kappa(x) \mid x \in X_i\}$ which consists of the control inputs used in the first step.
3. Verified integrating the ODE $\dot{x} = f(x, u), \dot{u} = 0$ from the initial condition $x(0) \in X_i, u \in U_i$ for δ_c time.
4. Updating X_i by the verified solution for $x(\delta_c)$.
5. Repeating the above 3 steps until the given time horizon is reached.

Pros and Cons of using Intervals

- **Compact Representation:** the size of an ITS is linear in the order, and also linear in the number of the state variables.
- **Low Computational Cost:** only numerical computation is used, no set subdivision is required.
- **Easy Implementation:** it requires only a library of interval analysis.
- **Low Flexibility.**
- **Heavy Accumulation of Overestimation.**
- **Expensive Refinement.**

Assignments

1. CAPD is a well developed C++ library for validated numerics for dynamical systems.

   http://capd.sourceforge.net/

2. Read the basic information of the tool.
3. Compile and run the examples for ODE integration and trajectory enclosure.
4. Use CAPD to compute the reachable set of the Van der Pol oscillator:

$$\dot{x} = y,\ \ \dot{y} = (1 - x^2)y - x$$

   from the initial set $x(0) \in [1.49,1.51], y(0) \in [2.39,2.41]$.