Lecture 4: Introduction to Nonsmooth Differential Equations

Moritz Diehl and Armin Nurkanović

Systems Control and Optimization Laboratory (syscop)
Summer School on Direct Methods for Optimal Control of Nonsmooth Systems
September 11-15, 2023

universität freiburg

Outline of the lecture

Hybrid and nonsmooth dynamical systems exhibit both continuous and discrete behavior.

Hybrid and nonsmooth dynamical systems exhibit both continuous and discrete behavior.

Why hybrid systems?

- Arise whenever first principles are coupled with *if-else* statements.
- From macroscopic empirical laws (Coulomb friction).
- Discrete events cause switches and/or jumps in the dynamics or the trajectory itself.

Hybrid and nonsmooth dynamical systems exhibit both continuous and discrete behavior.

Why hybrid systems?

- Arise whenever first principles are coupled with *if-else* statements.
- From macroscopic empirical laws (Coulomb friction).
- Discrete events cause switches and/or jumps in the dynamics or the trajectory itself.
- Discrete/integer control decisions.

What is so special about nonsmooth system?

Some basic questions arise

1. What is the difference between nonsmooth and hybrid systems?

What is so special about nonsmooth system?

Some basic questions arise

1. What is the difference between nonsmooth and hybrid systems?
2. Are systems with integer controls (on/off decisions) *nonsmooth* or *hybrid* systems?

What is so special about nonsmooth system?

Some basic questions arise

1. What is the difference between nonsmooth and hybrid systems?
2. Are systems with integer controls (on/off decisions) *nonsmooth* or *hybrid* systems?
3. How to classify nonsmooth systems?

What is so special about nonsmooth system?

Some basic questions arise

1. What is the difference between nonsmooth and hybrid systems?
2. Are systems with integer controls (on/off decisions) *nonsmooth* or *hybrid* systems?
3. How to classify nonsmooth systems?
4. What is so special about nonsmoothness?

What is so special about nonsmooth system?

Some basic questions arise

1. What is the difference between nonsmooth and hybrid systems?
2. Are systems with integer controls (on/off decisions) *nonsmooth* or *hybrid* systems?
3. How to classify nonsmooth systems?
4. What is so special about nonsmoothness?
5. How to treat nonsmooth systems numerically?
6. How to mathematically describe such systems?

What is so special about nonsmooth system?

Some basic questions arise

1. What is the difference between nonsmooth and hybrid systems?
2. Are systems with integer controls (on/off decisions) *nonsmooth* or *hybrid* systems?
3. How to classify nonsmooth systems?
4. What is so special about nonsmoothness?
5. How to treat nonsmooth systems numerically?
6. How to mathematically describe such systems?
7. Why not smooth everything?

Hybrid and/or nonsmooth systems

Nonsmooth dynamical systems

- Abstract nonsmooth ODE: $\dot{x} = f(x(t))$

Hybrid dynamical systems

- Hybrid system: very general term covers also nonsmooth sys.

Hybrid and/or nonsmooth systems

1. What is the difference between nonsmooth and hybrid systems?

Nonsmooth dynamical systems

- Abstract nonsmooth ODE: $\dot{x} = f(x(t))$
- The solution $x(t)$ and dynamics f do not have smoothness properties

Hybrid dynamical systems

- Hybrid system: very general term covers also nonsmooth sys.
- In literature often *hybrid finite automaton* meant

Hybrid and/or nonsmooth systems

1. What is the difference between nonsmooth and hybrid systems?

Nonsmooth dynamical systems

- Abstract nonsmooth ODE: $\dot{x} = f(x(t))$
- The solution $x(t)$ and dynamics f do not have smoothness properties
- Combinatorial structure implicitly encoded in nonsmooth equations

Hybrid dynamical systems

- Hybrid system: very general term covers also nonsmooth sys.
- In literature often *hybrid finite automaton* meant
- Combinatorial structure explicitly encoded in finite automaton

Hybrid and/or nonsmooth systems

1. What is the difference between nonsmooth and hybrid systems?

Nonsmooth dynamical systems

- Abstract nonsmooth ODE: $\dot{x} = f(x(t))$
- The solution $x(t)$ and dynamics f do not have smoothness properties
- Combinatorial structure implicitly encoded in nonsmooth equations
- Easy to write down, difficult to solve

Hybrid dynamical systems

- Hybrid system: very general term covers also nonsmooth sys.
- In literature often *hybrid finite automaton* meant
- Combinatorial structure explicitly encoded in finite automaton
- Difficult to write down, easier to solve

Hybrid and/or nonsmooth systems

1. What is the difference between nonsmooth and hybrid systems?

Nonsmooth dynamical systems

- Abstract nonsmooth ODE: $\dot{x} = f(x(t))$
- The solution $x(t)$ and dynamics f do not have smoothness properties
- Combinatorial structure implicitly encoded in nonsmooth equations
- Easy to write down, difficult to solve
- More present in applied mathematics literature

Hybrid dynamical systems

- Hybrid system: very general term covers also nonsmooth sys.
- In literature often *hybrid finite automaton* meant
- Combinatorial structure explicitly encoded in finite automaton
- Difficult to write down, easier to solve
- More present in control theory literature

Difference between hybrid and nonsmooth systems - Example 1

Nonsmooth dynamical system

$$\ddot{q} = -g + \lambda$$

$$0 \leq q \perp \lambda \geq 0$$

$$\text{if } q(t) = 0,\ \ v(t^-) \text{ and } \ \leq 0,$$

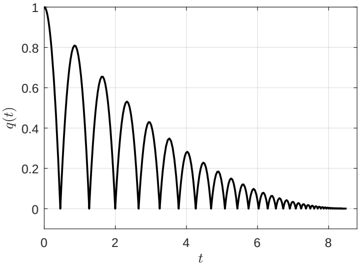
$$\text{then } v(t^+) = -\epsilon_{\mathrm{r}} v(t^-)$$

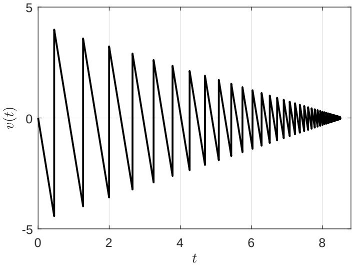
Hybrid dynamical system

$$q(t) = 0, v(t^-) \leq 0$$

$$\ddot{q} = -q$$

$$v(t^+) = -\epsilon_{\mathrm{r}} v(t^-)$$





Difference between hybrid and nonsmooth systems - Example 2

Nonsmooth dynamical system

$$\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix} = \begin{bmatrix}u_1+\lambda_1\\ -g+u_2+\lambda_2\end{bmatrix}$$

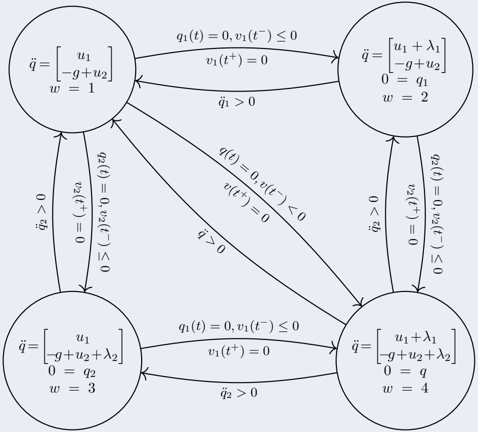
$$0 \leq q_1 \perp \lambda_1 \geq 0$$

$$0 \leq q_2 \perp \lambda_2 \geq 0$$

$$\text{if } q_i(t) = 0 \text{ and } v_i(t^-) \leq 0,$$

$$\text{then } v_i(t^+) = 0 \; i = 1, 2$$

Hybrid dynamical system

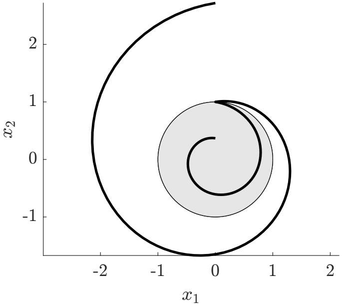


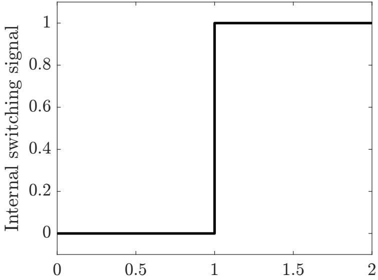
Nonsmooth/hybrid systems experience switches and jumps

Type of switches

Depending on how the discrete events or switches are triggered, we distinguish between:

1.) **internal switches**: triggered implicitly, depending on the systems' differential state





Switch can happen only when $x(t)$ reaches some surface in the state space

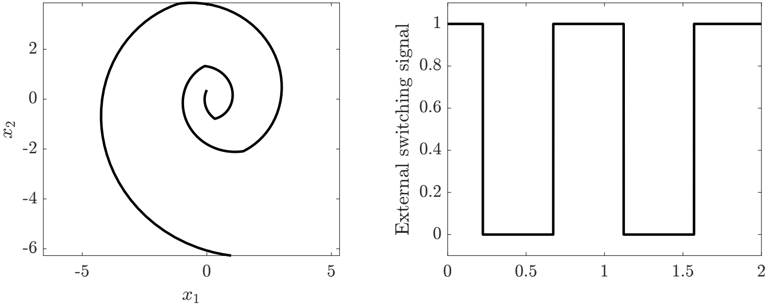
Nonsmooth/hybrid systems experience switches and jumps

Type of switches

Depending on how the discrete events or switches are triggered, we distinguish between:

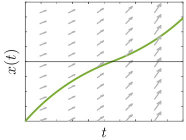
1.) **internal switches**: triggered implicitly, depending on the systems' differential state

2.) external switches: triggered explicitly, independent of the differential state

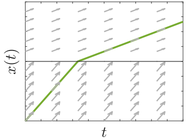


Switch can happen anytime - no matter where $x(t)$ is in the state space

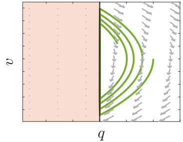
Regard an ordinary differential equation (ODE) with a **nonsmooth** right-hand side (RHS).
Distinguish three cases:



NSD1: nondifferentiable RHS, e.g., $\dot{x} = 1 + |x|$



NSD2: state dependent switch of RHS, e.g., $\dot{x} = 2 - \operatorname{sign}(x)$



NSD3: state dependent jump, e.g., bouncing ball, $v(t_+) = -0.9\, v(t_-)$

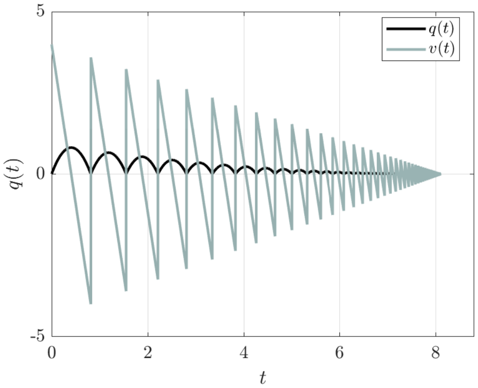
Outline of the lecture

Infinitely many switches in finite time - Zeno's phenomenon

The bouncing ball example - NSD3

$$\begin{aligned} \dot{q}(t) &= v(t) \\ m\,\dot{v}(t) &= -g \\ v(t^+) &= -\epsilon_{\mathrm{r}} v(t^-),\ \text{if } v(t^-) \leq 0 \text{ and } q(t) = 0 \\ q(0) &= 0,\ \ v(0) > 0 \end{aligned}$$

- Coefficient of restitution $\epsilon_{\mathrm{r}} \in [0,1]$, e.g., $\epsilon_{\mathrm{r}} = 0.9$
- $t_1 = \frac{2v(0)}{g},\ t_2 = t_1 + \frac{2\epsilon_{\mathrm{r}} v(0)}{g}, \ldots$
- $\Delta_{k+1} = t_{k+1} - t_k = \frac{2\,\epsilon_{\mathrm{r}}^k\,v(0)}{g}$
- Since $\epsilon_{\mathrm{r}} < 1$ it follows that $\lim_{k\to\infty} \Delta_k = 0$

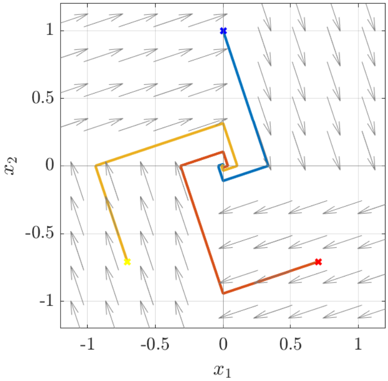


Infinitely many switches in finite time - Zeno's phenomenon

A Filippov system - NSD2

$$\dot{x}_1 \in -\text{sign}(x_1) + 2\text{sign}(x_2)$$
$$\dot{x}_2 \in -2\text{sign}(x_1) - \text{sign}(x_2)$$

- Real world system do not experience Zeno
- By modeling and design one wants to avoid this behavior
- Might complicate the numerical computations sometimes

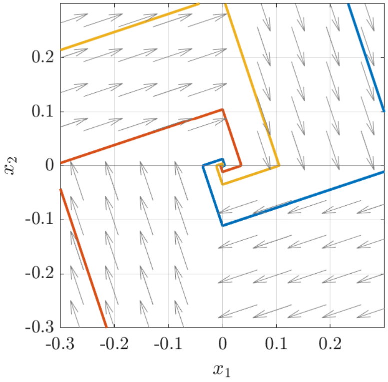


Infinitely many switches in finite time - Zeno's phenomenon

A Filippov system - NSD2

$$\dot{x}_1 \in -\mathrm{sign}(x_1) + 2\mathrm{sign}(x_2)$$
$$\dot{x}_2 \in -2\mathrm{sign}(x_1) - \mathrm{sign}(x_2)$$

- Real world system do not experience Zeno
- By modeling and design one wants to avoid this behavior
- Might complicate the numerical computations sometimes



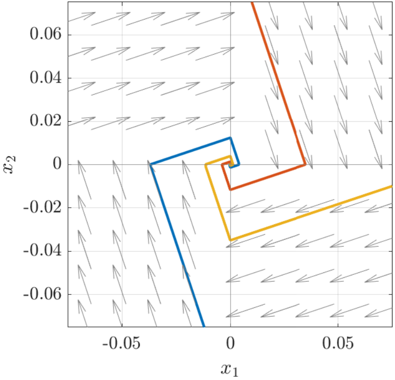
Zoom in: trajectories spiral down

Infinitely many switches in finite time - Zeno's phenomenon

A Filippov system - NSD2

$$
\begin{aligned}
\dot{x}_1 &\in -\operatorname{sign}(x_1) + 2\operatorname{sign}(x_2)\\
\dot{x}_2 &\in -2\operatorname{sign}(x_1) - \operatorname{sign}(x_2)
\end{aligned}
$$

- Real world system do not experience Zeno
- By modeling and design one wants to avoid this behavior
- Might complicate the numerical computations sometimes



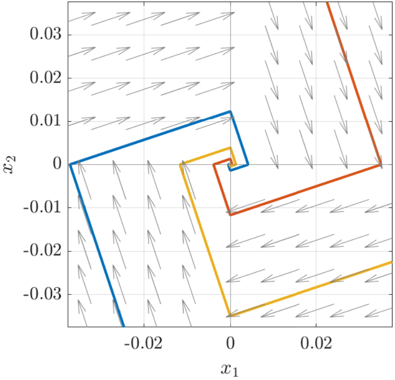
Zoom in: trajectories spiral down

Infinitely many switches in finite time - Zeno's phenomenon

A Filippov system - NSD2

$$\dot{x}_1 \in -\text{sign}(x_1) + 2\text{sign}(x_2)$$
$$\dot{x}_2 \in -2\text{sign}(x_1) - \text{sign}(x_2)$$

- Real world system do not experience Zeno
- By modeling and design one wants to avoid this behavior
- Might complicate the numerical computations sometimes



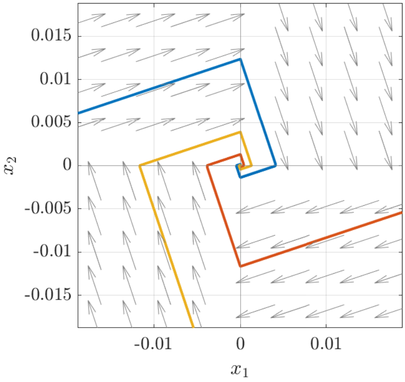
Zoom in: trajectories spiral down

Infinitely many switches in finite time - Zeno's phenomenon

A Filippov system - NSD2

$$\dot{x}_1 \in -\mathrm{sign}(x_1) + 2\mathrm{sign}(x_2)$$
$$\dot{x}_2 \in -2\mathrm{sign}(x_1) - \mathrm{sign}(x_2)$$

- Real world system do not experience Zeno
- By modeling and design one wants to avoid this behavior
- Might complicate the numerical computations sometimes



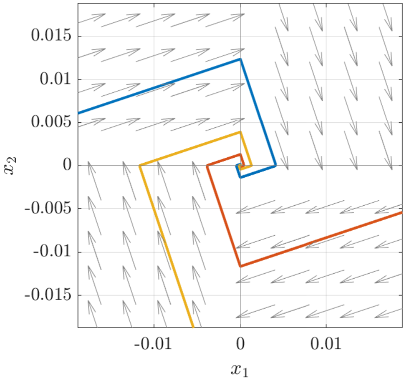
Zoom in: trajectories spiral down

Infinitely many switches in finite time - Zeno's phenomenon

A Filippov system - NSD2

$$\dot{x}_1 \in -\mathrm{sign}(x_1) + 2\mathrm{sign}(x_2)$$
$$\dot{x}_2 \in -2\mathrm{sign}(x_1) - \mathrm{sign}(x_2)$$

- Real world system do not experience Zeno
- By modeling and design one wants to avoid this behavior
- Might complicate the numerical computations sometimes



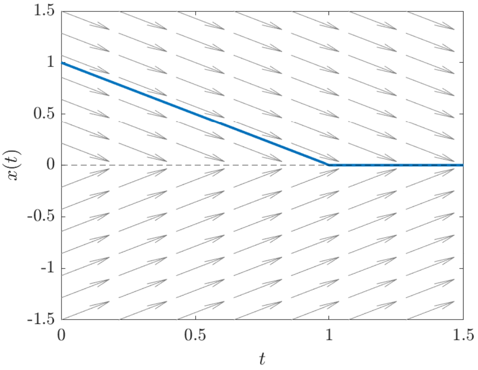
Zoom in: trajectories spiral down

Reduced systems dimensions and sliding modes

A sliding mode example

$$\dot{x} \in -\mathrm{sign}(x)$$

- System evolves on surface of discontinuity
- Need to define meaningful dynamics (treated later in detail)

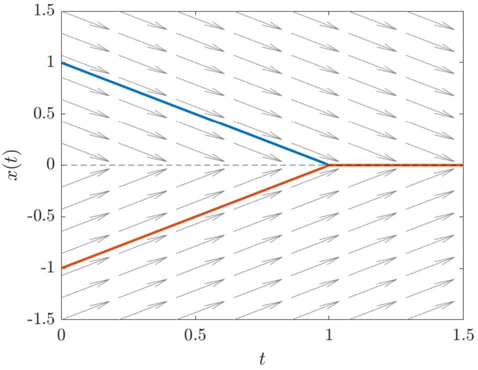


Reduced systems dimensions and sliding modes

A sliding mode example

$$\dot{x} \in -\mathrm{sign}(x)$$

- System evolves on surface of discontinuity
- Need to define meaningful dynamics (treated later in detail)
- Reduced system dimension

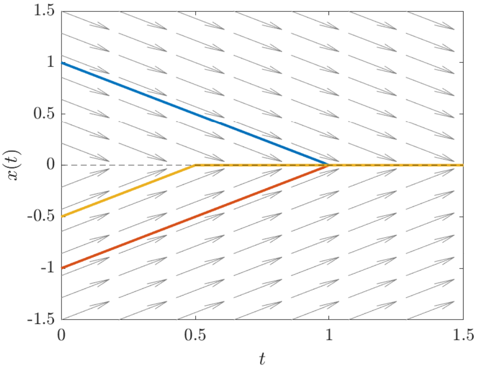


Reduced systems dimensions and sliding modes

A sliding mode example

$$\dot{x} \in -\text{sign}(x)$$

- System evolves on surface of discontinuity
- Need to define meaningful dynamics (treated later in detail)
- Reduced system dimension
- Solution not unique backwards in time

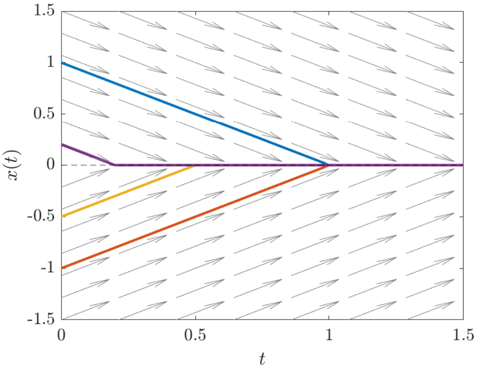


Reduced systems dimensions and sliding modes

A sliding mode example

$$\dot{x} \in -\mathrm{sign}(x)$$

- System evolves on surface of discontinuity
- Need to define meaningful dynamics (treated later in detail)
- Reduced system dimension
- Solution not unique backwards in time
- Dynamics switch from ODE to DAE of higher index

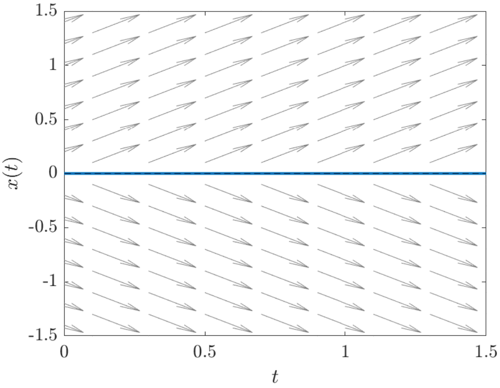


Nonunique solutions

Nonunique solutions example

$$\dot{x} \in \mathrm{sign}(x),\ x(0) = 0$$

- In nonsmooth systems examples with nonunique solutions easily constructed
- It may not be clear what numerical algorithms do

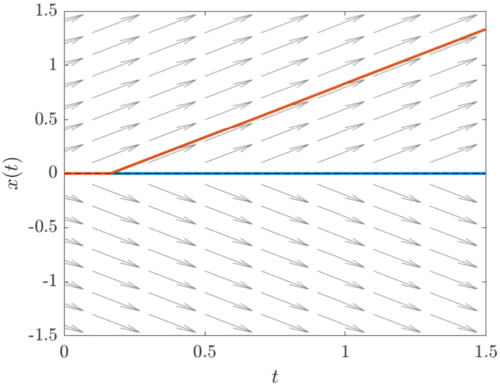


Nonunique solutions

Nonunique solutions example

$$\dot{x} \in \operatorname{sign}(x),\ x(0) = 0$$

- In nonsmooth systems examples with nonunique solutions easily constructed
- It may not be clear what numerical algorithms do
- To be avoided in controller design

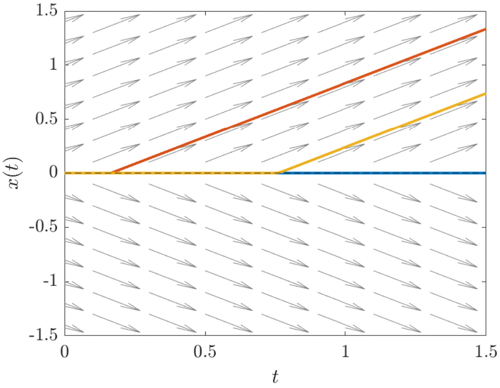


Nonunique solutions

Nonunique solutions example

$$\dot{x} \in \operatorname{sign}(x),\ x(0) = 0$$

- In nonsmooth systems examples with nonunique solutions easily constructed
- It may not be clear what numerical algorithms do
- To be avoided in controller design
- Dynamics switch from DAE of higher index to ODE

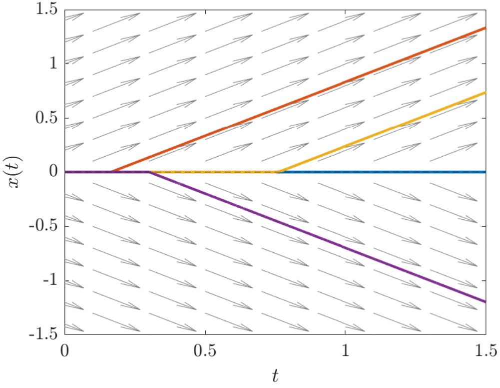


Nonunique solutions

Nonunique solutions example

$$\dot{x} \in \operatorname{sign}(x),\ x(0) = 0$$

- In nonsmooth systems examples with nonunique solutions easily constructed
- It may not be clear what numerical algorithms do
- To be avoided in controller design
- Dynamics switch from DAE of higher index to ODE



Stability and instability due to switches and jumps

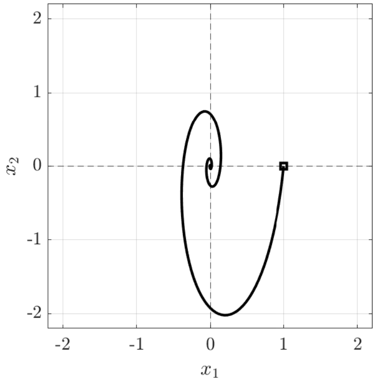
Unstable piecewise affine systems

$$\dot{x} = f(x) := \begin{cases} A_1 x, & \text{if } x_1 x_2 \leq 0 \\ A_2 x, & \text{if } x_1 x_2 > 0 \end{cases}$$

with

$$A_1 = \begin{bmatrix} -1 & 1 \\ -10 & -1 \end{bmatrix}, \; A_2 = \begin{bmatrix} -1 & 10 \\ -1 & -1 \end{bmatrix}$$

- First and third quadrant: $\dot{x} = A_1 x$



$\dot{x} = A_1 x$ - stable

Stability and instability due to switches and jumps

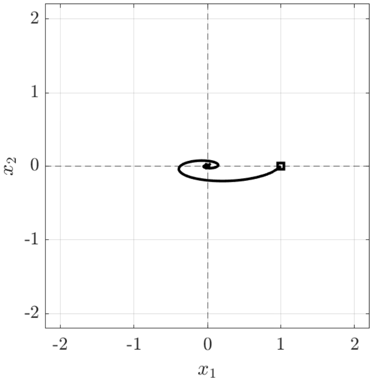
Unstable piecewise affine systems

$$\dot{x} = f(x) := \begin{cases} A_1 x, & \text{if } x_1 x_2 \leq 0 \\ A_2 x, & \text{if } x_1 x_2 > 0 \end{cases}$$

with

$$A_1 = \begin{bmatrix} -1 & 1 \\ -10 & -1 \end{bmatrix}, \; A_2 = \begin{bmatrix} -1 & 10 \\ -1 & -1 \end{bmatrix}$$

- First and third quadrant: $\dot{x} = A_1 x$
- Second and fourth quadrant: $\dot{x} = A_2 x$



$\dot{x} = A_2 x$ - stable

Stability and instability due to switches and jumps

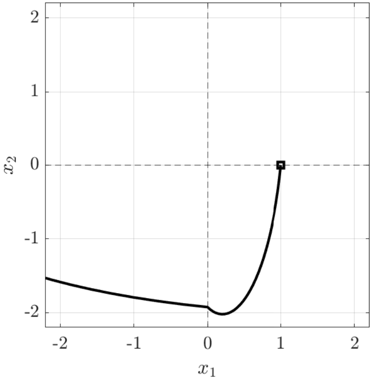
Unstable piecewise affine systems

$$\dot{x} = f(x) := \begin{cases} A_1 x, & \text{if } x_1 x_2 \leq 0 \\ A_2 x, & \text{if } x_1 x_2 > 0 \end{cases}$$

with

$$A_1 = \begin{bmatrix} -1 & 1 \\ -10 & -1 \end{bmatrix}, \; A_2 = \begin{bmatrix} -1 & 10 \\ -1 & -1 \end{bmatrix}$$

- First and third quadrant: $\dot{x} = A_1 x$
- Second and fourth quadrant: $\dot{x} = A_2 x$
- Nonsmooth systems can have stable *modes* but still be overall unstable



$\dot{x} = f(x)$ - **unstable**

Stability and instability due to switches and jumps

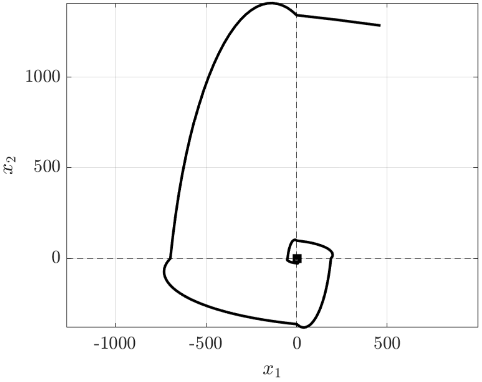
Unstable piecewise affine systems

$$\dot{x} = f(x) := \begin{cases} A_1 x, & \text{if } x_1 x_2 \leq 0 \\ A_2 x, & \text{if } x_1 x_2 > 0 \end{cases}$$

with

$$A_1 = \begin{bmatrix} -1 & 1 \\ -10 & -1 \end{bmatrix}, \; A_2 = \begin{bmatrix} -1 & 10 \\ -1 & -1 \end{bmatrix}$$

- First and third quadrant: $\dot{x} = A_1 x$
- Second and fourth quadrant: $\dot{x} = A_2 x$
- Nonsmooth systems can have stable *modes* but still be overall unstable



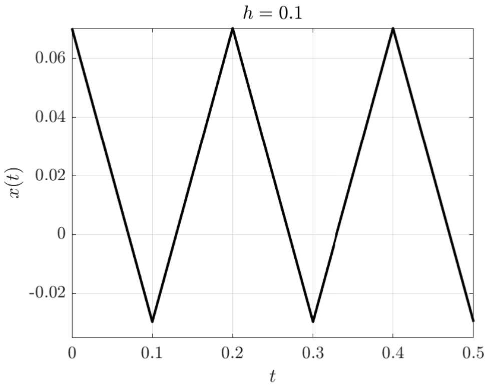
$\dot{x} = f(x)$ - **unstable**

Numerical chattering in sliding mode examples

Explicit Euler for nonsmooth systems

$$x_{k+1} = x_k - h\,\mathrm{sign}(x_k)$$

- In presence of discontinuities numerical solutions can *chatter* around a discontinuity

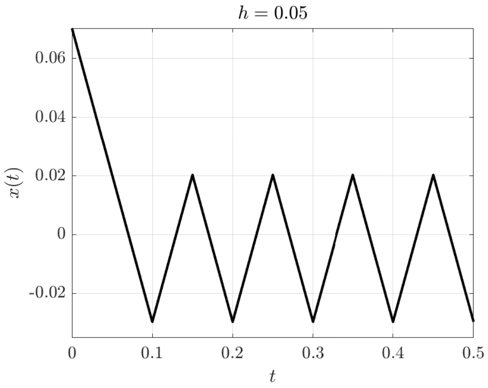


Numerical chattering in sliding mode examples

Explicit Euler for nonsmooth systems

$$x_{k+1} = x_k - h\,\mathrm{sign}(x_k)$$

- In presence of discontinuities numerical solutions can *chatter* around a discontinuity
- Decreasing the step size might worsen things

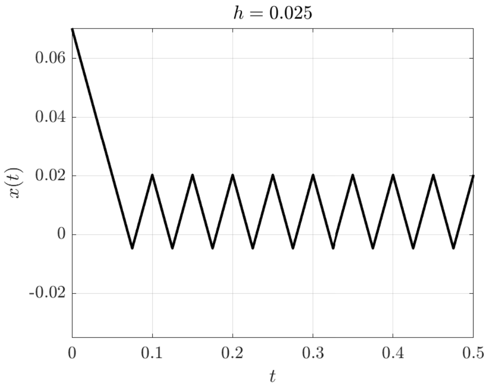


Numerical chattering in sliding mode examples

Explicit Euler for nonsmooth systems

$$x_{k+1} = x_k - h\,\mathrm{sign}(x_k)$$

- In presence of discontinuities numerical solutions can *chatter* around a discontinuity
- Decreasing the step size might worsen things

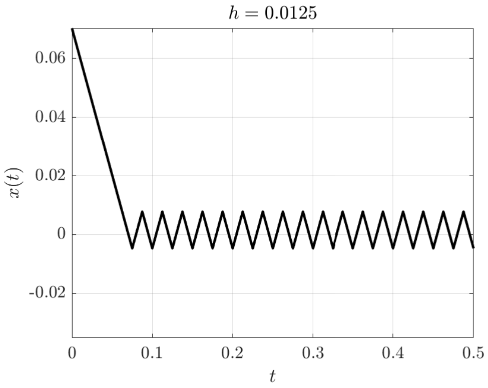


Numerical chattering in sliding mode examples

Explicit Euler for nonsmooth systems

$$x_{k+1} = x_k - h\,\mathrm{sign}(x_k)$$

- In presence of discontinuities numerical solutions can *chatter* around a discontinuity
- Decreasing the step size might worsen things

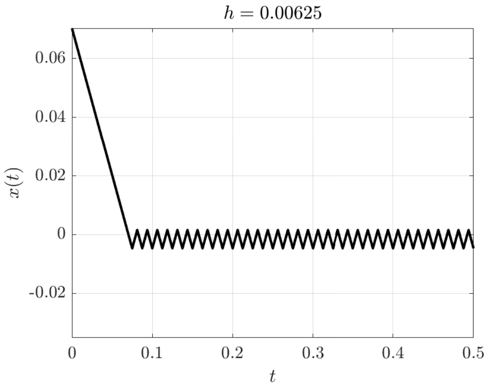


Numerical chattering in sliding mode examples

Explicit Euler for nonsmooth systems

$$x_{k+1} = x_k - h\,\mathrm{sign}(x_k)$$

- In presence of discontinuities numerical solutions can *chatter* around a discontinuity
- Decreasing the step size might worsen things

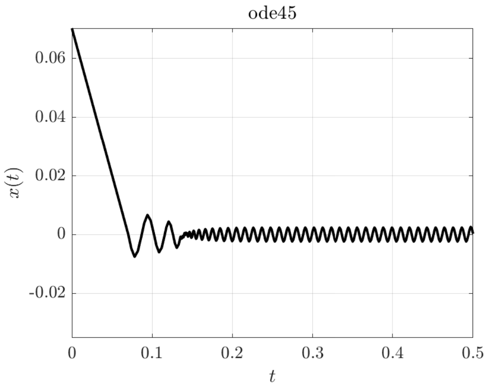


Numerical chattering in sliding mode examples

Explicit Euler for nonsmooth systems

$$x_{k+1} = x_k - h\,\mathrm{sign}(x_k)$$

- In presence of discontinuities numerical solutions can *chatter* around a discontinuity
- Decreasing the step size might worsen things
- Even sophisticated codes may struggle

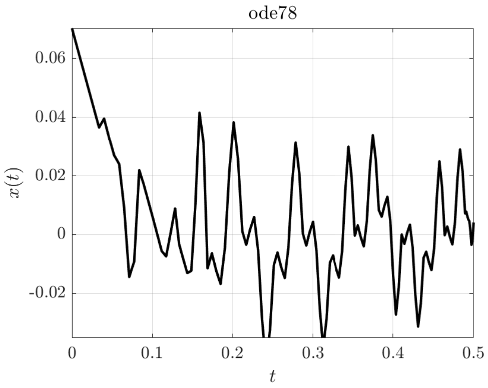


Numerical chattering in sliding mode examples

Explicit Euler for nonsmooth systems

$$x_{k+1} = x_k - h\,\mathrm{sign}(x_k)$$

- In presence of discontinuities numerical solutions can *chatter* around a discontinuity
- Decreasing the step size might worsen things
- Even sophisticated codes may struggle
- Method converges - but qualitative behavior is not good

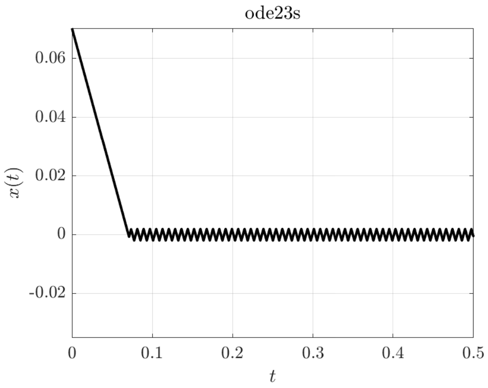


Numerical chattering in sliding mode examples

Explicit Euler for nonsmooth systems

$$x_{k+1} = x_k - h\,\mathrm{sign}(x_k)$$

- In presence of discontinuities numerical solutions can *chatter* around a discontinuity
- Decreasing the step size might worsen things
- Even sophisticated codes may struggle
- Method converges - but qualitative behavior is not good

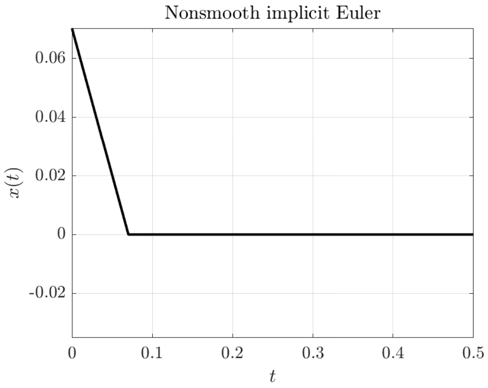


Numerical chattering in sliding mode examples

Explicit Euler for nonsmooth systems

$$x_{k+1} = x_k - h\,\mathrm{sign}(x_k)$$

- In presence of discontinuities numerical solutions can *chatter* around a discontinuity
- Decreasing the step size might worsen things
- Even sophisticated codes may struggle
- Method converges - but qualitative behavior is not good
- Nonsmooth implicit methods resolve the issue (treated later)



Sensitivities - derivatives of ODE solutions

- *Direct optimal control* solves nonlinear programs via Newton-type methods
- Newton-type optimization (SQP or IP), needs to compute 1^{st} and 2^{nd}-order derivatives

Sensitivities - derivatives of ODE solutions

- *Direct optimal control* solves nonlinear programs via Newton-type methods
- Newton-type optimization (SQP or IP), needs to compute 1^{st} and 2^{nd}-order derivatives
- Given a solution $x(T; x_0, \hat{u})$ of the IVP: $\dot{x} = f(x, \hat{u})$, $x(0) = x_0$, the sensitivities are:

$$\frac{\partial x(T; x_0, \hat{u})}{\partial x_0}, \ \frac{\partial x(T; x_0, \hat{u})}{\partial \hat{u}}$$

Sensitivities - derivatives of ODE solutions

- *Direct optimal control* solves nonlinear programs via Newton-type methods
- Newton-type optimization (SQP or IP), needs to compute 1^{st} and 2^{nd}-order derivatives
- Given a solution $x(T; x_0, \hat{u})$ of the IVP: $\dot{x} = f(x, \hat{u})$, $x(0) = x_0$, the sensitivities are:

$$\frac{\partial x(T; x_0, \hat{u})}{\partial x_0}, \quad \frac{\partial x(T; x_0, \hat{u})}{\partial \hat{u}}$$

- In direct optimal control we need to linearize $x_{k+1} = \phi_f(x_k, \hat{u})$ and have

$$\Delta x_{k+1} = \frac{\partial \phi_f(x_k, \hat{u})}{\partial x_k} \Delta x_k + \frac{\partial \phi_f(x_k, \hat{u})}{\partial \hat{u}} \Delta \hat{u}$$

Sensitivities - derivatives of ODE solutions

- *Direct optimal control* solves nonlinear programs via Newton-type methods
- Newton-type optimization (SQP or IP), needs to compute 1^{st} and 2^{nd}-order derivatives
- Given a solution $x(T; x_0, \hat{u})$ of the IVP: $\dot{x} = f(x, \hat{u})$, $x(0) = x_0$, the sensitivities are:

$$\frac{\partial x(T; x_0, \hat{u})}{\partial x_0}, \quad \frac{\partial x(T; x_0, \hat{u})}{\partial \hat{u}}$$

- In direct optimal control we need to linearize $x_{k+1} = \phi_f(x_k, \hat{u})$ and have

$$\Delta x_{k+1} = \frac{\partial \phi_f(x_k, \hat{u})}{\partial x_k} \Delta x_k + \frac{\partial \phi_f(x_k, \hat{u})}{\partial \hat{u}} \Delta \hat{u}$$

- Under mild assumptions:

$$\lim_{h \to 0} \frac{\partial x_N}{\partial x_0} = \frac{\partial x(T; x_0, \hat{u})}{\partial x_0}$$

Sensitivities - derivatives of ODE solutions

- *Direct optimal control* solves nonlinear programs via Newton-type methods
- Newton-type optimization (SQP or IP), needs to compute 1^{st} and 2^{nd}-order derivatives
- Given a solution $x(T; x_0, \hat{u})$ of the IVP: $\dot{x} = f(x, \hat{u})$, $x(0) = x_0$, the sensitivities are:

$$\frac{\partial x(T; x_0, \hat{u})}{\partial x_0}, \quad \frac{\partial x(T; x_0, \hat{u})}{\partial \hat{u}}$$

- In direct optimal control we need to linearize $x_{k+1} = \phi_f(x_k, \hat{u})$ and have

$$\Delta x_{k+1} = \frac{\partial \phi_f(x_k, \hat{u})}{\partial x_k} \Delta x_k + \frac{\partial \phi_f(x_k, \hat{u})}{\partial \hat{u}} \Delta \hat{u}$$

- Under mild assumptions:

$$\lim_{h \to 0} \frac{\partial x_N}{\partial x_0} = \frac{\partial x(T; x_0, \hat{u})}{\partial x_0}$$

What do the sensitivities look like if $x(T; x_0, \hat{u})$ has kinks and jumps?

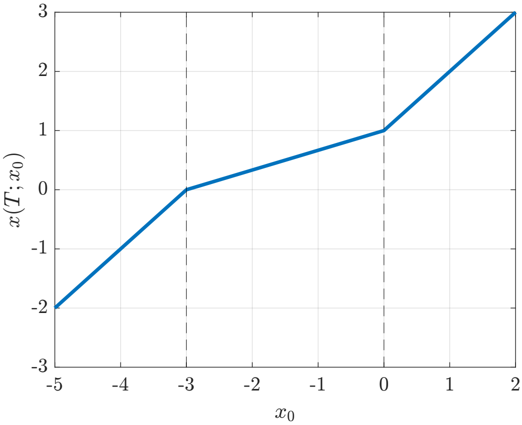
The sensitivities are nonsmooth

NSD2 tutorial example

$$\dot{x}(t) = \begin{cases} 3, & \text{if } x < 0 \\ 1, & \text{if } x > 0 \end{cases}, \qquad t \in [0, T]$$

$$x(0) = x_0$$

- Solution map $x(T, x_0)$ has kinks



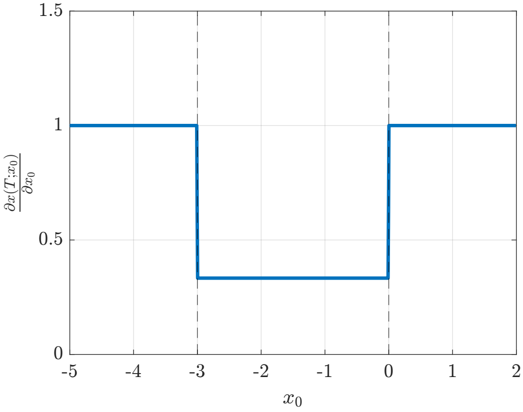
The sensitivities are nonsmooth

NSD2 tutorial example

$$\dot{x}(t) = \begin{cases} 3, & \text{if } x < 0 \\ 1, & \text{if } x > 0 \end{cases}, \quad t \in [0, T]$$

$$x(0) = x_0$$

- Solution map $x(T, x_0)$ has kinks
- Sensitivity $\frac{\partial x(T;x_0)}{\partial x_0}$ has jumps



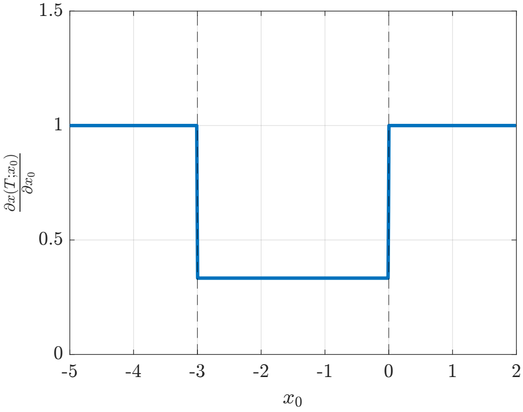
The sensitivities are nonsmooth

NSD2 tutorial example

$$\dot{x}(t) = \begin{cases} 3, & \text{if } x < 0 \\ 1, & \text{if } x > 0 \end{cases}, \quad t \in [0, T]$$

$$x(0) = x_0$$

- Solution map $x(T, x_0)$ has kinks
- Sensitivity $\frac{\partial x(T;x_0)}{\partial x_0}$ has jumps
- When does $\lim_{h\to 0} \frac{\partial x_N}{\partial x_0} = \frac{\partial x(T;x_0)}{\partial x_0}$?



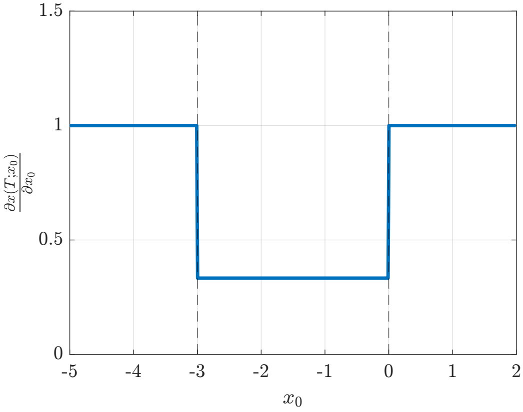
The sensitivities are nonsmooth

NSD2 tutorial example

$$\dot{x}(t) = \begin{cases} 3, & \text{if } x < 0 \\ 1, & \text{if } x > 0 \end{cases}, \quad t \in [0, T]$$

$$x(0) = x_0$$

- Solution map $x(T, x_0)$ has kinks
- Sensitivity $\frac{\partial x(T;x_0)}{\partial x_0}$ has jumps
- When does $\lim_{h\to 0} \frac{\partial x_N}{\partial x_0} = \frac{\partial x(T;x_0)}{\partial x_0}$?
- What do nonsmooth sensitivities mean for Newton's method?



The sensitivities are nonsmooth

NSD3 tutorial example

$$\ddot{q} = \begin{bmatrix} \lambda_1 \\ -g + \lambda_2 \end{bmatrix}$$

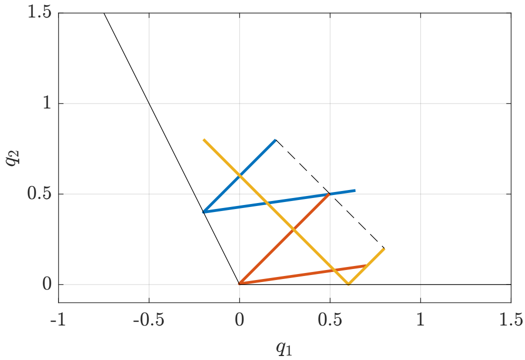
$$0 \leq q_2 + 2q_1 \perp \lambda_1 \geq 0$$

$$0 \leq q_2 \perp \lambda_2 \geq 0$$

$$\text{if } c_i(q(t)) = 0 \text{ and } n_i(q)^\top v(t^-) \leq 0, \text{ then}$$

$$n_i(q)^\top v(t^+) = -\epsilon_{\mathrm{r}} n_i(q)^\top v(t^-), \ i = 1, 2,$$

- $c_1(q) = q_2 + 2q_1$, $c_2(q) = q_2$ gap functions
- λ_1, λ_2 - normal contact forces, $n_1(q)$ and $n_2(q)$ contact normals



The sensitivities are nonsmooth

NSD3 tutorial example

$$\ddot{q} = \begin{bmatrix} \lambda_1 \\ -g + \lambda_2 \end{bmatrix}$$

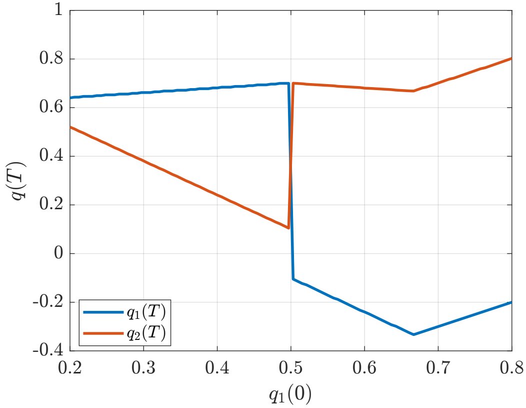
$$0 \leq q_2 + 2q_1 \perp \lambda_1 \geq 0$$

$$0 \leq q_2 \perp \lambda_2 \geq 0$$

$$\text{if } c_i(q(t)) = 0 \text{ and } n_i(q)^\top v(t^-) \leq 0, \text{ then}$$

$$n_i(q)^\top v(t^+) = -\epsilon_{\mathrm{r}} n_i(q)^\top v(t^-), \; i = 1, 2,$$

- $c_1(q) = q_2 + 2q_1$, $c_2(q) = q_2$ gap functions
- λ_1, λ_2 - normal contact forces, $n_1(q)$ and $n_2(q)$ contact normals
- Sensitivity $\frac{\partial x(T;x_0)}{\partial x_0}$ has kinks and jumps



The sensitivities are nonsmooth

NSD3 tutorial example

$$\ddot{q} = \begin{bmatrix} \lambda_1 \\ -g + \lambda_2 \end{bmatrix}$$

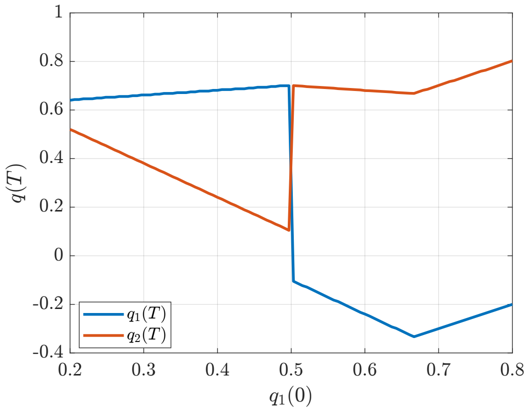
$$0 \leq q_2 + 2q_1 \perp \lambda_1 \geq 0$$

$$0 \leq q_2 \perp \lambda_2 \geq 0$$

$$\text{if } c_i(q(t)) = 0 \text{ and } n_i(q)^\top v(t^-) \leq 0, \text{ then}$$

$$n_i(q)^\top v(t^+) = -\epsilon_\mathrm{r} n_i(q)^\top v(t^-), \; i = 1, 2,$$

- $c_1(q) = q_2 + 2q_1$, $c_2(q) = q_2$ gap functions
- λ_1, λ_2 - normal contact forces, $n_1(q)$ and $n_2(q)$ contact normals
- Sensitivity $\frac{\partial x(T;x_0)}{\partial x_0}$ has kinks and jumps
- When does $\lim_{h\to 0} \frac{\partial x_N}{\partial x_0} = \frac{\partial x(T;x_0)}{\partial x_0}$?

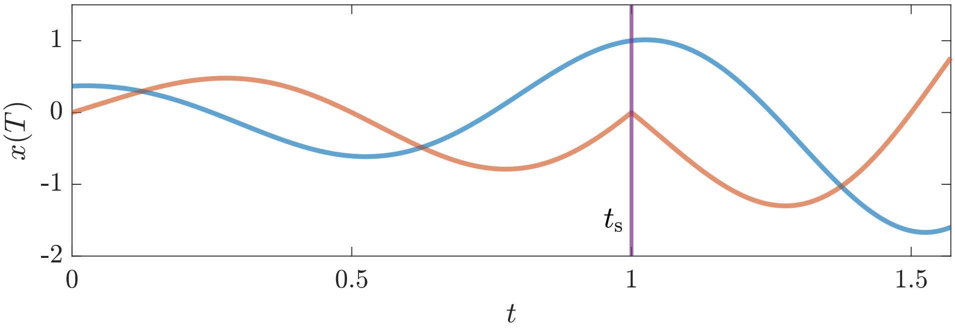


Outline of this lecture

Time discretization methods for nonsmooth ODEs

Approaches to discretize and simulate a nonsmooth ODE

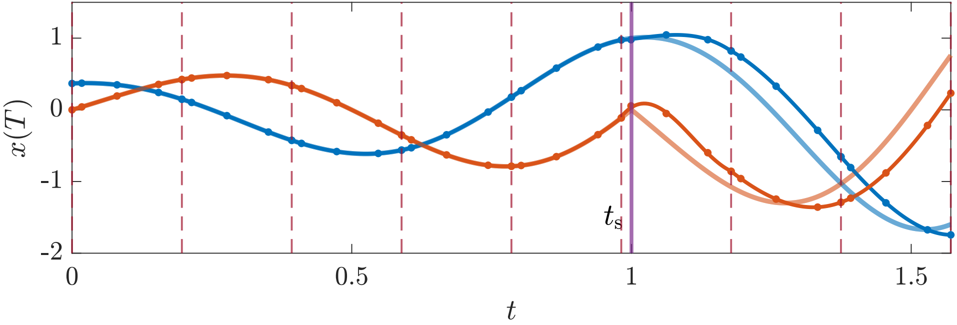
1) event-capturing, time-stepping methods (can handle Zeno, low accuracy)



Time discretization methods for nonsmooth ODEs

Approaches to discretize and simulate a nonsmooth ODE

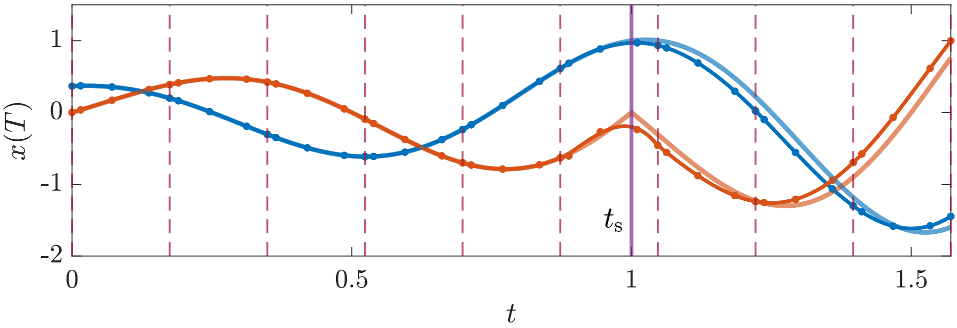
1) event-capturing, time-stepping methods (can handle Zeno, low accuracy)



Time discretization methods for nonsmooth ODEs

Approaches to discretize and simulate a nonsmooth ODE

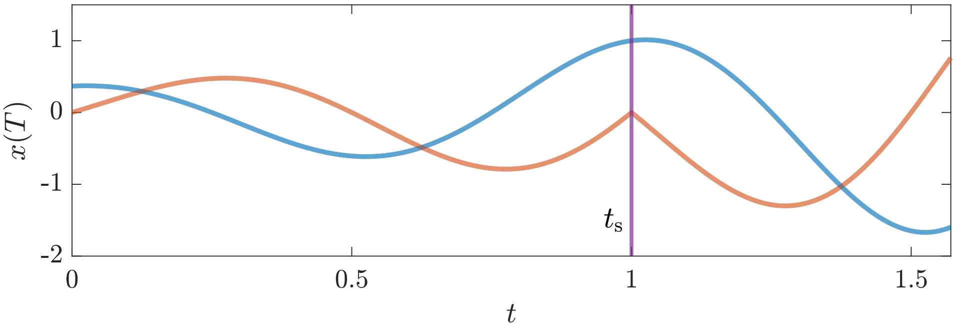
1) event-capturing, time-stepping methods (can handle Zeno, low accuracy)



Time discretization methods for nonsmooth ODEs

Approaches to discretize and simulate a nonsmooth ODE

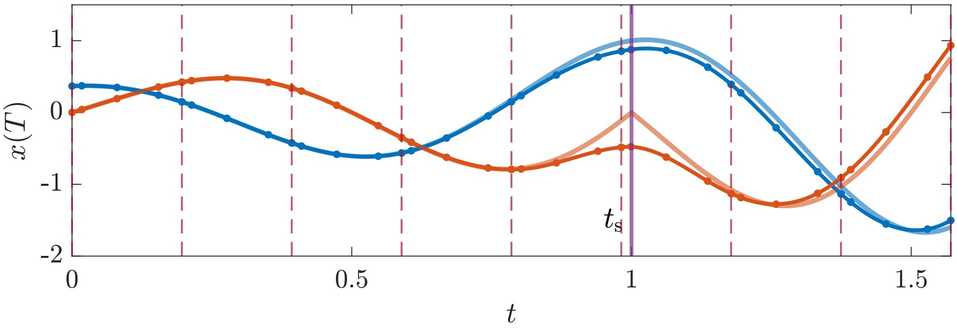
1) event-capturing, time-stepping methods (can handle Zeno, low accuracy)
2) smoothing and penalty methods (low accuracy, easy to implement)



Time discretization methods for nonsmooth ODEs

Approaches to discretize and simulate a nonsmooth ODE

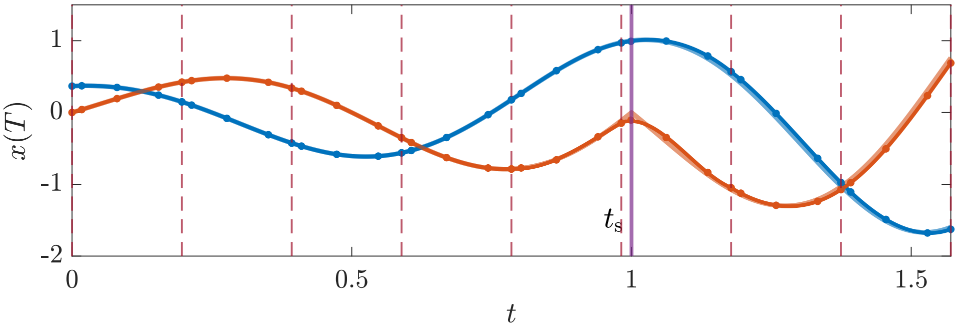
1) event-capturing, time-stepping methods (can handle Zeno, low accuracy)
2) smoothing and penalty methods (low accuracy, easy to implement)



Time discretization methods for nonsmooth ODEs

Approaches to discretize and simulate a nonsmooth ODE

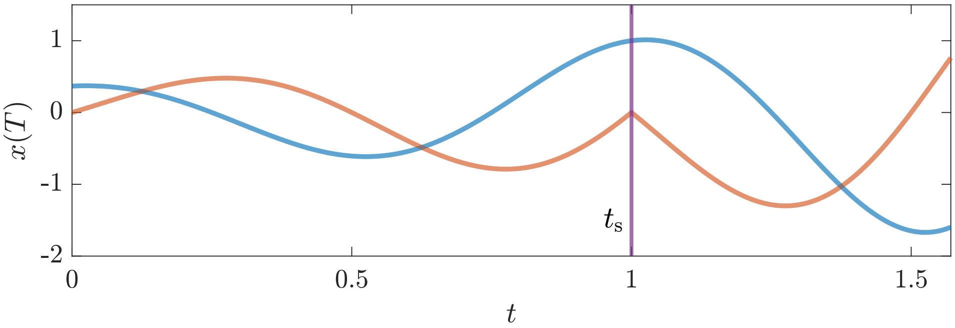
1) event-capturing, time-stepping methods (can handle Zeno, low accuracy)
2) smoothing and penalty methods (low accuracy, easy to implement)



Time discretization methods for nonsmooth ODEs

Approaches to discretize and simulate a nonsmooth ODE

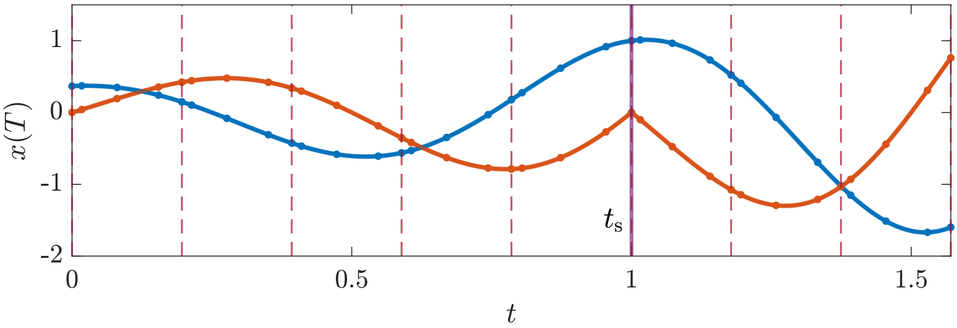
1) event-capturing, time-stepping methods (can handle Zeno, low accuracy)
2) smoothing and penalty methods (low accuracy, easy to implement)
3) event-driven, switch-detecting, active-set methods (cannot handle Zeno, high accuracy)



Time discretization methods for nonsmooth ODEs

Approaches to discretize and simulate a nonsmooth ODE

1) event-capturing, time-stepping methods (can handle Zeno, low accuracy)
2) smoothing and penalty methods (low accuracy, easy to implement)
3) event-driven, switch-detecting, active-set methods (cannot handle Zeno, high accuracy)



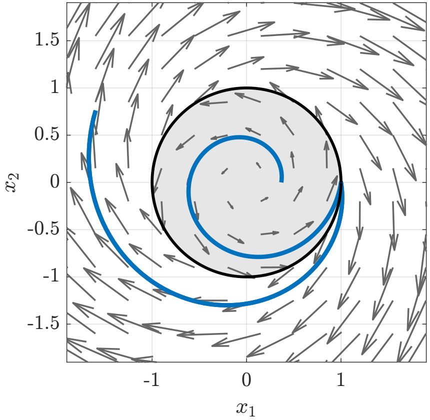
Integration order plots for different simulation methods

Compute global integration error $E(T)$ using different strategies

Tutorial example

$$\dot{x} = \begin{cases} A_1 x, & \|x\|_2^2 < 1, \\ A_2 x, & \|x\|_2^2 > 1, \end{cases}$$

with $A_1 = \begin{bmatrix} 1 & 2\pi \\ -2\pi & 1 \end{bmatrix}$, $A_2 = \begin{bmatrix} 1 & -2\pi \\ 2\pi & 1 \end{bmatrix}$, $x(0) = (e^{-1}, 0)$ for $t \in [0, T]$.



Compute solution approximation:

1. with fixed step size IRK methods (time-stepping),

Integration order plots for different simulation methods

Compute global integration error $E(T)$ using different strategies

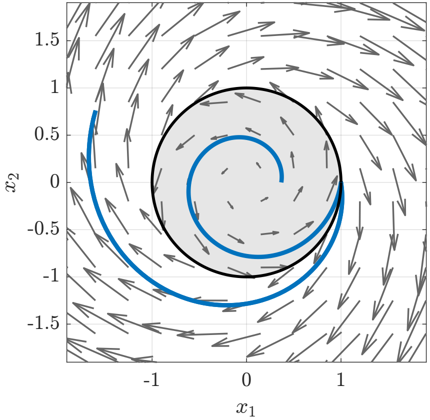
Tutorial example

$$\dot{x} = \begin{cases} A_1 x, & \|x\|_2^2 < 1, \\ A_2 x, & \|x\|_2^2 > 1, \end{cases}$$

with $A_1 = \begin{bmatrix} 1 & 2\pi \\ -2\pi & 1 \end{bmatrix}$, $A_2 = \begin{bmatrix} 1 & -2\pi \\ 2\pi & 1 \end{bmatrix}$, $x(0) = (e^{-1}, 0)$ for $t \in [0, T]$.

Compute solution approximation:

1. with fixed step size IRK methods (time-stepping),
2. with sophisticated adaptive step size methods (time-stepping),



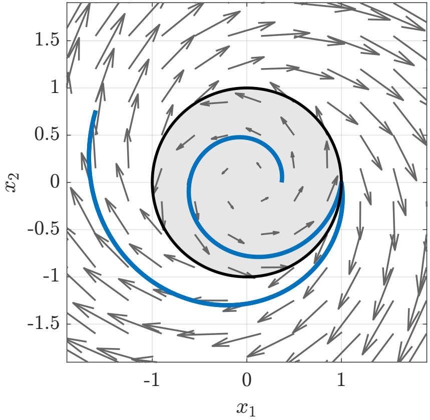
Integration order plots for different simulation methods

Compute global integration error $E(T)$ using different strategies

Tutorial example

$$\dot{x} = \begin{cases} A_1 x, & \|x\|_2^2 < 1, \\ A_2 x, & \|x\|_2^2 > 1, \end{cases}$$

with $A_1 = \begin{bmatrix} 1 & 2\pi \\ -2\pi & 1 \end{bmatrix}$, $A_2 = \begin{bmatrix} 1 & -2\pi \\ 2\pi & 1 \end{bmatrix}$, $x(0) = (e^{-1}, 0)$ for $t \in [0, T]$.



Compute solution approximation:

1. with fixed step size IRK methods (time-stepping),
2. with sophisticated adaptive step size methods (time-stepping),
3. with switch detecting integrators,

Integration order plots for different simulation methods

Compute global integration error $E(T)$ using different strategies

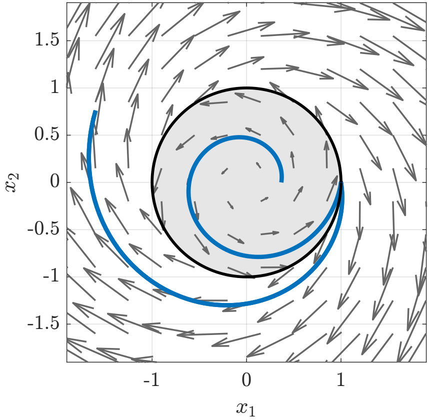
Tutorial example

$$\dot{x} = \begin{cases} A_1 x, & \|x\|_2^2 < 1, \\ A_2 x, & \|x\|_2^2 > 1, \end{cases}$$

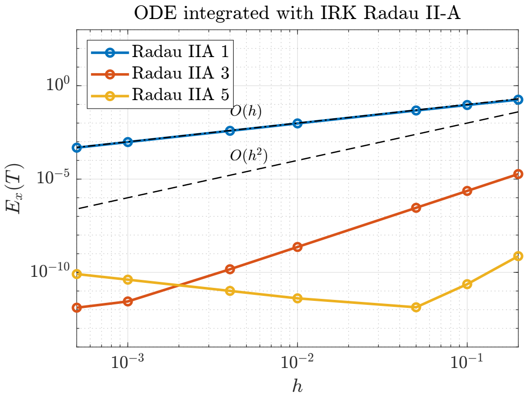
with $A_1 = \begin{bmatrix} 1 & 2\pi \\ -2\pi & 1 \end{bmatrix}$, $A_2 = \begin{bmatrix} 1 & -2\pi \\ 2\pi & 1 \end{bmatrix}$, $x(0) = (e^{-1}, 0)$ for $t \in [0, T]$.

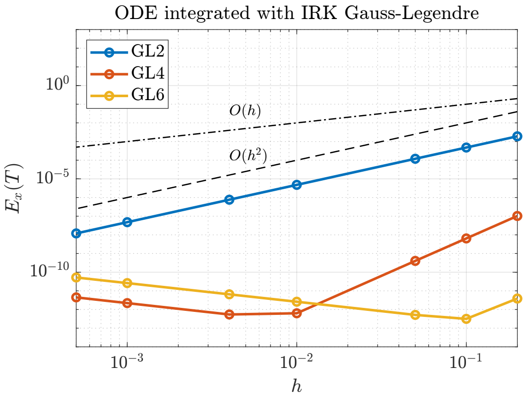
Compute solution approximation:

1. with fixed step size IRK methods (time-stepping),
2. with sophisticated adaptive step size methods (time-stepping),
3. with switch detecting integrators,
4. via smoothed approximations.

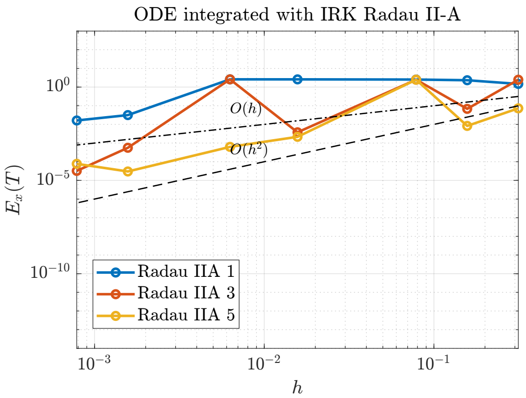


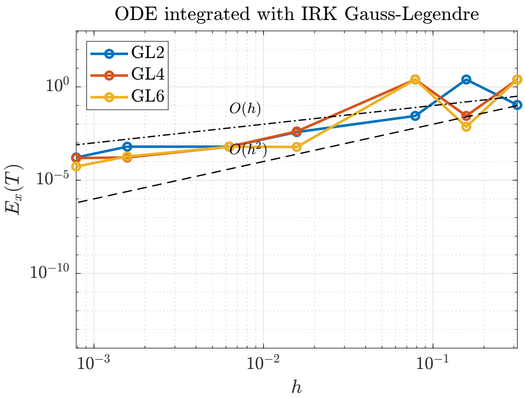
Integration order plots fixed step size Implicit Runge-Kutta methods





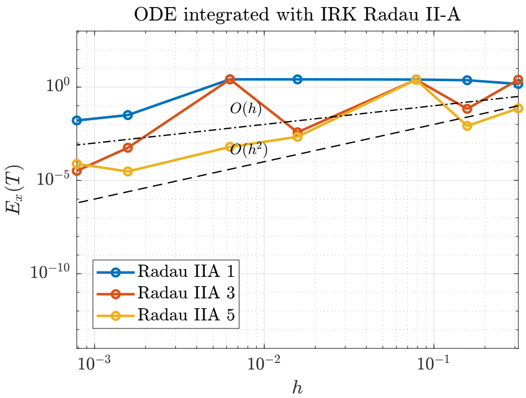
Simulation time $T = 1$ - no switch - high accuracy

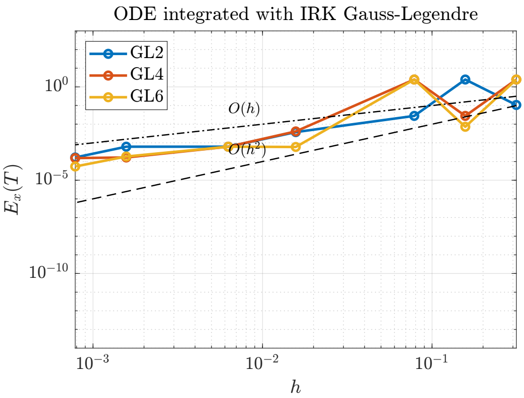




Simulation time $T = \pi/2$ - switch happens - low accuracy

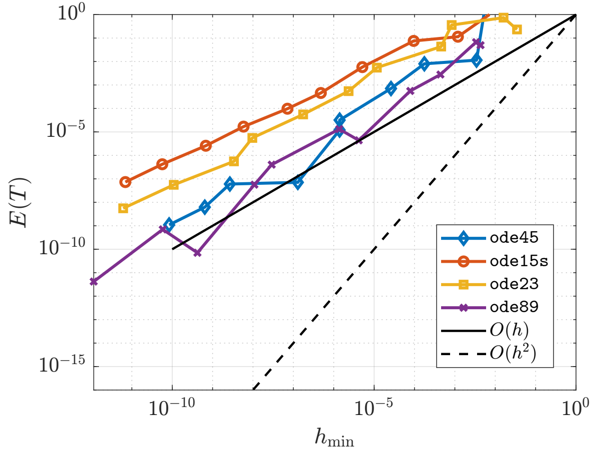
Integration order plots fixed step size Implicit Runge-Kutta methods



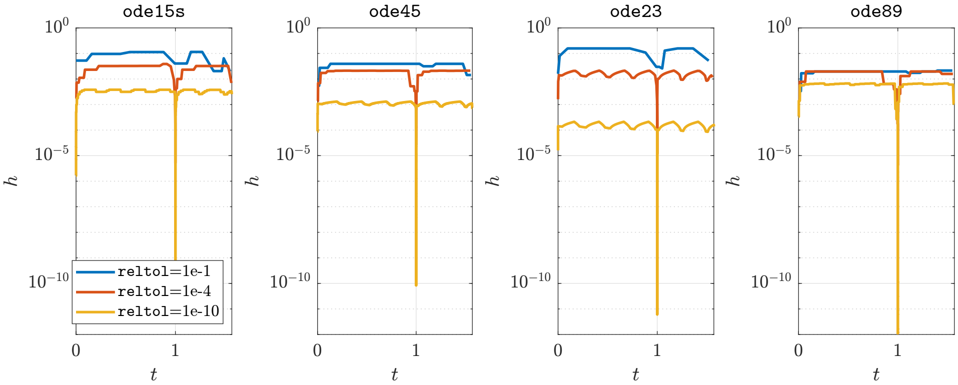


The nonsmoothness leads to sever order reduction, all methods have $O(h)$ accuracy.

Integration order plots adaptive step size methods

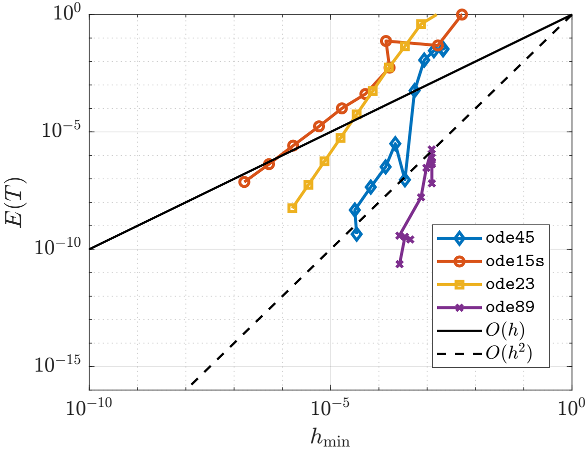


Very small step size necessary to achieve high accuracy even with very sophisticated methods.



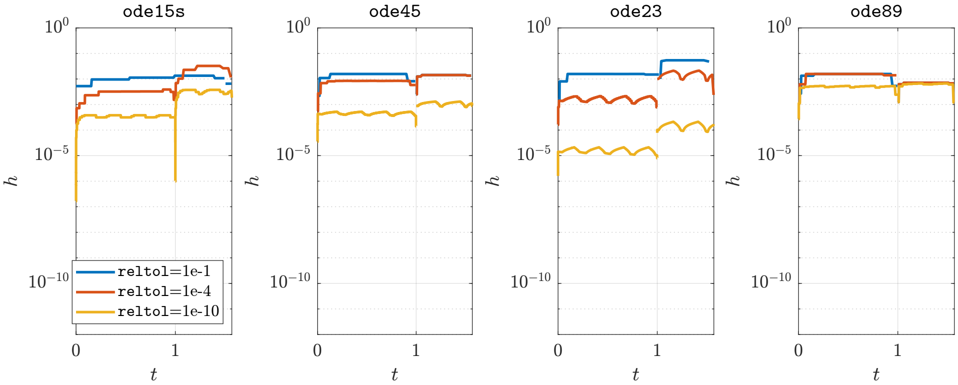
Step size small around switch - many switches = very slow integration.

Adaptive step size methods **with switch detection**



Switch detected explicitly - high accuracy properties recovered.

Adaptive step size methods **with switch detection**



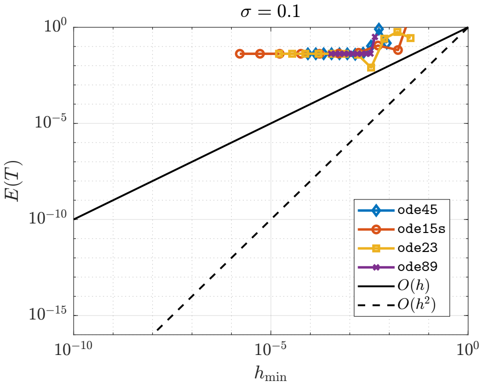
No extremely small step sizes around the switch.

Accuracy order plots for smoothing

Error dominated by σ

Smooth approximation parameterized by σ

$$\dot{x} = (1 - \alpha_\sigma(x))A_1 x + \alpha_\sigma(x) A_2 x, \ \ \alpha_\sigma(x) = \frac{1}{2}\left(1 - \tanh\left(\frac{\|x\|_2^2 - 1}{\sigma}\right)\right)$$

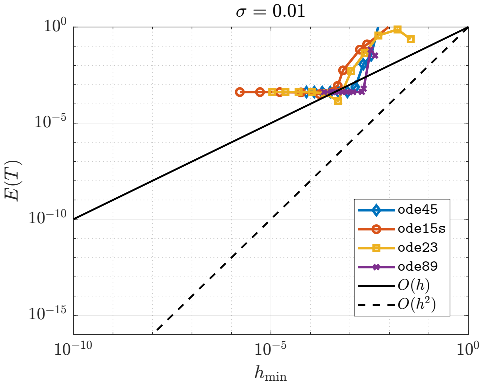


Accuracy order plots for smoothing

Error dominated by σ

Smooth approximation parameterized by σ

$$\dot{x} = (1 - \alpha_\sigma(x))A_1 x + \alpha_\sigma(x) A_2 x, \ \ \alpha_\sigma(x) = \frac{1}{2}\left(1 - \tanh\left(\frac{\|x\|_2^2 - 1}{\sigma}\right)\right)$$

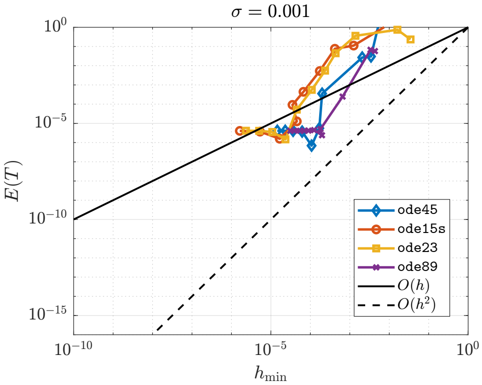


Accuracy order plots for smoothing

Error dominated by σ

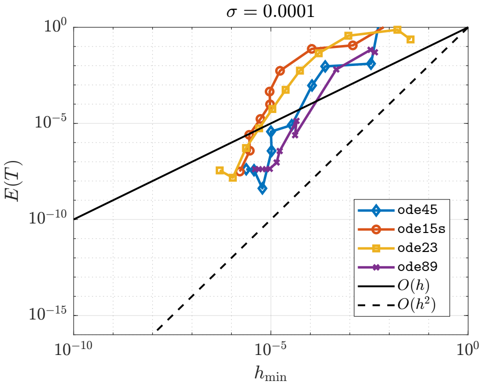
Smooth approximation parameterized by σ

$$\dot{x} = (1-\alpha_\sigma(x))A_1 x + \alpha_\sigma(x) A_2 x, \;\; \alpha_\sigma(x) = \frac{1}{2}\left(1 - \tanh\left(\frac{\|x\|_2^2 - 1}{\sigma}\right)\right)$$



Smooth approximation parameterized by σ

$$\dot{x} = (1 - \alpha_\sigma(x))A_1 x + \alpha_\sigma(x) A_2 x, \;\; \alpha_\sigma(x) = \frac{1}{2}\left(1 - \tanh\left(\frac{\|x\|_2^2 - 1}{\sigma}\right)\right)$$

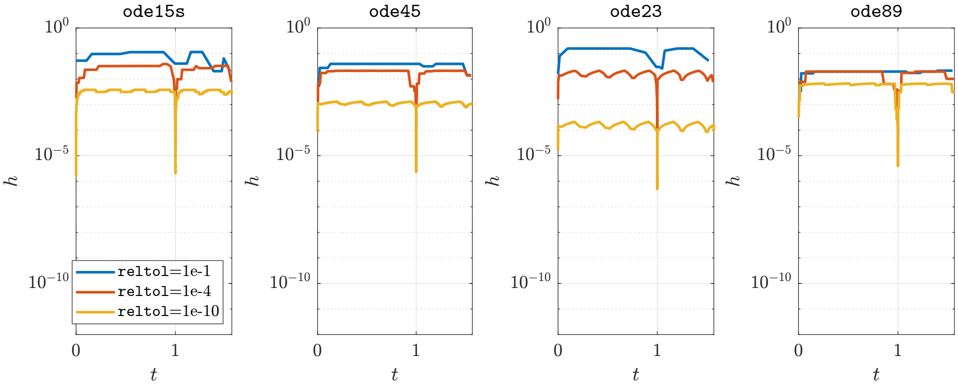


Accuracy order plots for smoothing

Error dominated by σ

Smooth approximation parameterized by σ

$$\dot{x} = (1 - \alpha_\sigma(x))A_1 x + \alpha_\sigma(x) A_2 x, \;\; \alpha_\sigma(x) = \frac{1}{2}\left(1 - \tanh\left(\frac{\|x\|_2^2 - 1}{\sigma}\right)\right)$$

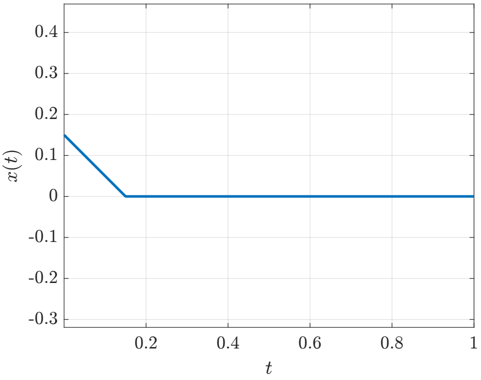


Smoothed sliding mode example

Error dominated by σ

Smooth approximation parameterized by $\sigma = 10^{-5}$

$$\dot{x} = -\text{sign}(x)$$

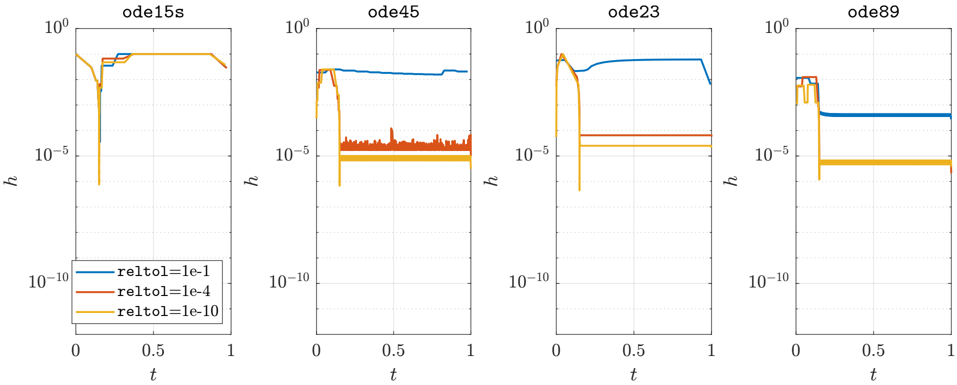


Smoothed sliding mode example

Error dominated by σ

Smooth approximation parameterized by $\sigma = 10^{-5}$

$$\dot{x} = -\tanh\left(\frac{x}{\sigma}\right)$$



Small σ makes system very stiff - small step sizes.

Outline of this lecture

6. How to mathematically described nonsmooth systems?

Differential inclusions

A very general class of nonsmooth dynamical systems is obtained by replacing the right-hand side of a smooth ODE with a set.

Differential Inclusions (DI)

The following equations is called a differential inclusion:

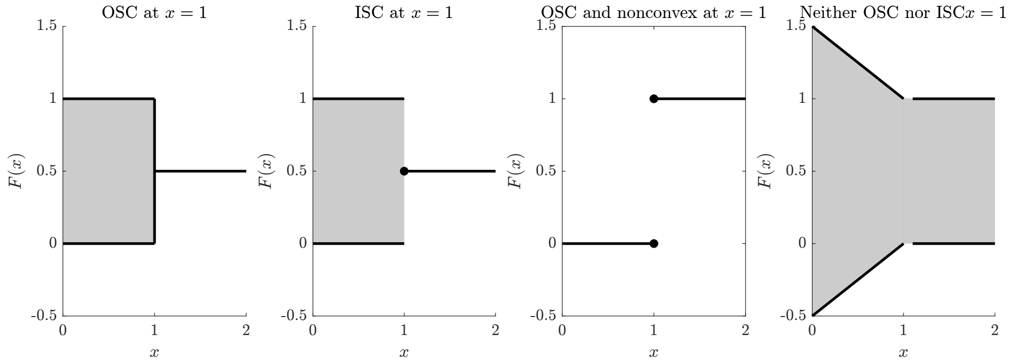
$$\dot{x}(t) \in F(t, x(t)) \text{ for almost all } t \in [0, T], \tag{1}$$

Here $F : \mathbb{R} \times \mathbb{R}^{n_x} \to \mathcal{P}(\mathbb{R}^{n_x})$ is a set-valued map which assigns to any point in time t and $x \in \mathbb{R}^{n_x}$ a set $F(t, x) \subseteq \mathbb{R}^{n_x}$. An element $y \in F(t, x(t))$ for a fixed $(t, x(t))$ is called a *selection*.

Outer and inner semi-continuous set-valued functions

Definition (OSC, ISC, continuity)

A set-valued function $F(\cdot)$ is outer-semi continuous (OSC) (resp. inner semi-continuous (ISC)) at $x_0 \in X$ if for every $\epsilon > 0$ there exists a $\delta > 0$ such that $F(x) \subset F(x_0) + \epsilon\mathcal{B}(0)$ (resp. $F(x_0) \subset F(x) + \epsilon\mathcal{B}(0)$) for all $x \in x_0 + \delta\mathcal{B}(0)$. It is called continuous at x_0 if it both OSC and ISC at this point.



Existence of solutions to differential inclusions

Theorem (Existence of solution, Theorem 4, p. 101 in Aubin, J. P., and Cellina, A., 1994)

Regard the initial value problem related to the DI (1) *with the initial value* $x(0) = x_0$. *Suppose that the function* $F : [0, T] \times \mathbb{R}^{n_x} \to \mathcal{P}(\mathbb{R}^{n_x})$ *satisfies the following conditions:.*

i) $\|y\| \leq C(t)(1 + \|x\|)$ *for all* x *and* $y \in F(t, x)$, *where* $C(\cdot)$ *is an integrable function,*

ii) $F(t, \cdot)$ *is outer semi-continuous for all* t,

iii) *the set* $F(t, x)$ *is nonempty and closed convex set for all* t *and* x,

Then there exists an absolutely continuous solution $x(\cdot)$ *to this initial value problem.*

Variational inequalities

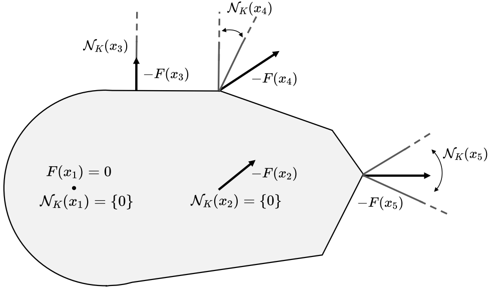
Definition

Let $K \subseteq \mathbb{R}^n$ be a closed convex set and $F : \mathbb{R}^n \to \mathbb{R}^n$. A variational inequality, denoted by $\mathrm{VI}(K, F)$, is the problem of finding $x \in \mathbb{R}^n$ such that

$$x \in K, \; F(x)^\top (y - x) \geq 0, \quad \text{for all } y \in K.$$

The set of solutions to this problem is denoted by $\mathrm{SOL}(K, F)$.

- $x \in K$ is a solution of $\mathrm{VI}(K, F)$ iff either $F(x) = 0$ or $F(x)$ forms a non-obtuse angle with every vector $y - x$ for all $y \in K$
- $\mathcal{N}_K(x) = \{v \in \mathbb{R}^n \mid v^\top (y - x) \leq 0, \text{ for all } y \in K\}$, $\mathrm{VI}(K, F)$ is the same as: $0 \ni F(x) + \mathcal{N}_K(x)$



x_1, x_3 and x_5 are solutions, x_2 and x_4 are not

Differential variational inequalities

Definition (Differential variational inequalities)

Given an initial value $x(0) = x_0$, a Differential Variational Inequality (DVI) is the problem of finding functions $x : [0, T] \to \mathbb{R}^{n_x}$ and $z : [0, T] \to \mathbb{R}^{n_z}$ such that

$$\dot{x}(t) = f(t, x(t), z(t)), \tag{2a}$$

$$z(t) \in K, \text{ for almost all } t, \tag{2b}$$

$$0 \leq (\hat{z} - z(t))^\top F(t, x(t), z(t)), \text{ for all } \hat{z} \in K \text{ and for almost all } t. \tag{2c}$$

- DVI can be easily cast into differential inclusions
- Denote the set of all solutions, parameterized by $x(t)$, of the VI (2c) by $\mathrm{SOL}(F(t, x(t), \cdot), K)$.

$$\dot{x}(t) \in f(t, x(t), \mathrm{SOL}(F(t, x(t), \cdot), K)),\ x(0) = x_0.$$

Dynamic complementarity systems

Definition (Dynamic complementarity systems)

Given an initial value $x(0) = x_0$, a dynamic complementarity system is the problem of finding functions $x : [0,T] \to \mathbb{R}^{n_x}$ and $z : [0,T] \to \mathbb{R}^{n_z}$ such that

$$\begin{aligned} \dot{x}(t) &= f(t, x(t), z(t)),\ x(0) = x_0, \\ 0 \leq z(t) &\perp F(t, x(t), z(t)) \geq 0,\ \text{for almost all } t, \end{aligned}$$

- Discrete-time counterpart: nonlinear complementarity problems (e.g. KKT conditions of an NLP)
- Computationally very useful as NCPs can often be solved efficiently
- Found in nonsmooth mechanics: complementarity between gap function and normal contact forces
- Filippov systems can be casted into DCS (next lecture)
- $\text{DI} \supset \text{DVI} \supset \text{DCS} \supset \text{ODE}.$

Discontinuous ODEs and Filippov systems

Regard an ODE with a discontinuous right-hand side and study the following IVP

$$\dot{x} = f(t, x(t)),\ x(0) = x_0.$$

Consider the following initial value problem:

$$\dot{x} = \begin{cases} 1, & x < 0, \\ -1, & x \geq 0, \end{cases}, \quad x(0) = x_0.$$

- For $x_0 > 0$, there exist the solution $x(t) = x(0) - t$ for $t \in [0, x_0)$
- For $x_0 < 0$, there exist the solution $x(t) = x(0) + t$ for $t \in [0, -x_0)$.
- As t beyond $|x_0|$ in both cases, each solution reaches the point $x(t) = 0$ and cannot leave it
- Since $\dot{x} = 0 \neq -1$, we have no solution in the classical or Carathéodory sense.
- To resolve this, we can use Filippov's solution concept - **next lecture**.

Summary

- External state depended switches and jumps are qualitatively different from integer controls - different numerical treatment.
- Nonsmooth systems exhibit rich behavior not seen in smooth systems.
- Accurate smooth approximation kill the performance of smooth solvers and,
- ... behave numerically as nonsmooth systems, but with smoothing ignore exploitable structure.
- Different classes of numerical methods for time discretization.
- There are many mathematical formalism to treat nonsmoothness.
- Depending on the formalism and *degree of nonsmoothness*, numerical methods must be adapted.

References

- Stewart, D. E. Dynamics with Inequalities: impacts and hard constraints. SIAM, 2011.
- Aubin, J. P., and Cellina, A. Differential Inclusions: Set-Valued Maps and Viability Theory. Springer-Verlag, 1984.
- Brogliato, B., and Tanwani, A. Dynamical systems coupled with monotone set-valued operators: Formalisms, applications, well-posedness, and stability. Siam Review 62, 1 (2020), 3–129.
- Acary, V., and Brogliato, B. Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics. Springer, Science and Business Media, 2008
- Smirnov, G. V. Introduction to the Theory of Differential Inclusions, vol. 41. American Mathematical Soc., 2002.
- Facchinei, F., and Pang, J.-S. Finite-dimensional variational inequalities and complementarity problems, vol. 1-2. Springer-Verlag, 2003.
- Nurkanović. A., Numerical Methods for Optimal Control of Nonsmooth Dynamical Systems, *to appear end of 2023*

Relating CCP to VIs

Proposition (Proposition 1.1.3. in Facchinei and Pang 2003)

Let K be a closed convex cone. A vector $x \in \mathbb{R}^n$ is a solution to $\mathrm{VI}(K, F)$ if and only if it is a solution to the cone complementarity problem:

$$K \ni x \perp F(x) \in K^*, \tag{3}$$

where this compact notation means that $x \in K, F(x) \in K^*$ and $F(x)^\top x = 0$.

Proof. Let x be a solution to the $\mathrm{VI}(K, F)$. On one hand, since K is a cone, setting $y = 0 \in K$ we have from $x \in K,\ F(x)^\top(y - x) \geq 0, \ \text{ for all } y \in K$, that $F(x)^\top x \leq 0$. On the other hand, from the definition of a cone $x \in K$ it follows that $2x \in K$. Again, from the VI and setting $y = 2x$ we obtain that $F(x)^\top x \geq 0$. Therefore, $F(x)^\top x = 0$. We further exploit that $F(x)^\top x \geq 0$, i.e., we can see that $F(x)^\top(y - x) \geq 0$ implies that $F(x)^\top y \geq 0$ for all $y \in K$, which is equivalent to $F(x) \in K^*$. Thus we have proven that x solves also (3). Conversely, if x solves (3), we have from the definition that $F(x)^\top y \geq 0$ for all $y \in K$ and $F(x)^\top x = 0$. Subtracting these relations we obtain that the VI holds. □