4. Direct methods for nonsmooth nonlinear optimal control

Armin Nurkanović

Systems Control and Optimization Laboratory, University of Freiburg, Germany

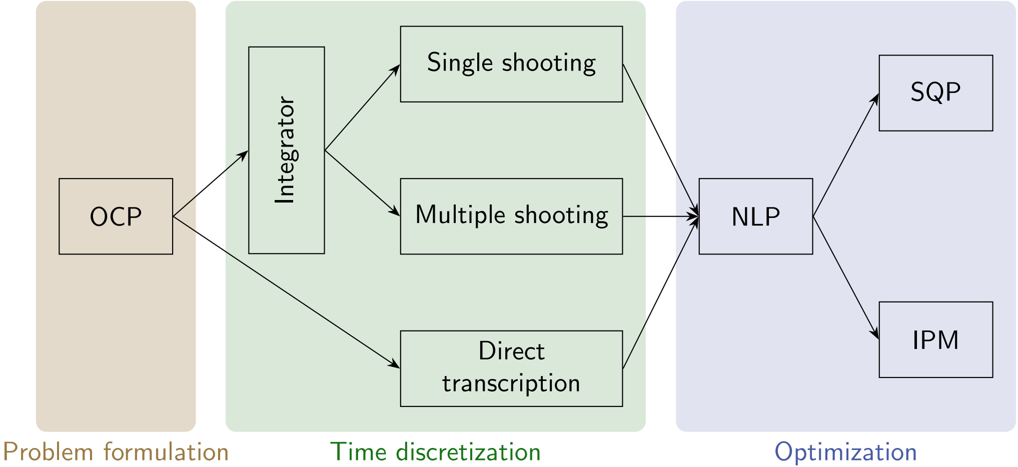
Winter School on Numerical Methods for Optimal Control of Nonsmooth Systems
École des Mines de Paris
February 3-5, 2025, Paris, France

universität freiburg

Outline of the lecture

Work flow in smooth direct optimal control

First discretize, then optimize.



OCP = Optimal Control Problem

NLP = Nonlinear Program

SQP = Sequential Quadratic Programming

IPM = Interior-Point Method

Figure inspired by Lecture 1, Numerical Methods for Optimal Control: Introduction, 2022, by Mario Zanon and Sébastien Gros.

The obvious way to tackle nonsmooth optimal control problems

Let us follow the path that worked so far:

1. Apply fixed-step size integration methods within direct single shooting, direct multiple shooting or direct description to the **nonsmooth** optimal control problem.

The obvious way to tackle nonsmooth optimal control problems

Let us follow the path that worked so far:

1. Apply fixed-step size integration methods within direct single shooting, direct multiple shooting or direct description to the **nonsmooth** optimal control problem.
2. Smooth the nonsmooth model, and apply standard direct methods.

The obvious way to tackle nonsmooth optimal control problems

Let us follow the path that worked so far:

1. Apply fixed-step size integration methods within direct single shooting, direct multiple shooting or direct description to the **nonsmooth** optimal control problem.
2. Smooth the nonsmooth model, and apply standard direct methods.

Due to nonsmooth dynamics, the resulting optimization problem is nonsmooth only in a few points.

What can go wrong?

Tutorial nonsmooth optimal control problem

Tutorial example inspired by [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

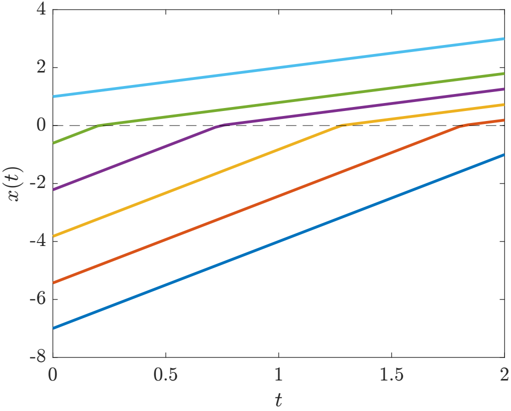
Continuous-time OCP

$$\min_{x(\cdot)\in\mathcal{C}^0([0,2])} \int_0^2 x(t)^2 \mathrm{d}t + (x(2)-5/3)^2$$
$$\text{s.t.} \quad \dot{x}(t) \in 2-\operatorname{sign}(x(t)), \quad t\in[0,2]$$

Free initial value $x(0)$ is the effective degree of freedom.

Equivalent reduced problem

$$\min_{x_0\in\mathbb{R}} V(x_0)$$



Example trajectories.

Tutorial nonsmooth optimal control problem

Tutorial example inspired by [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

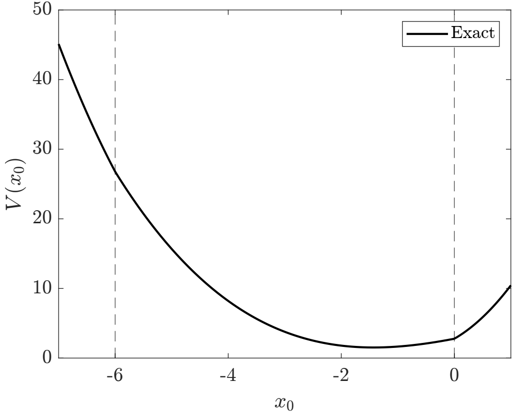
Continuous-time OCP

$$\min_{x(\cdot),\lambda(\cdot),s(\cdot)} \int_0^2 x(t)^2 \mathrm{d}t + (x(2) - 5/3)^2$$
$$\text{s.t.} \quad \dot{x}(t) = 2 - s(t)$$
$$0 \leq \lambda(t) - x(t) \perp 1 + s(t) \geq 0$$
$$0 \leq \lambda(t) \perp 1 - s(t) \geq 0, \ t \in [0, 2]$$

Free initial value $x(0)$ is the effective degree of freedom.

Equivalent reduced problem

$$\min_{x_0 \in \mathbb{R}} V(x_0)$$



Denote by $V(x_0)$ the nonsmooth objective value for the unique feasible trajectory starting at $x(0) = x_0$.

Tutorial nonsmooth optimal control problem

Tutorial example inspired by [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

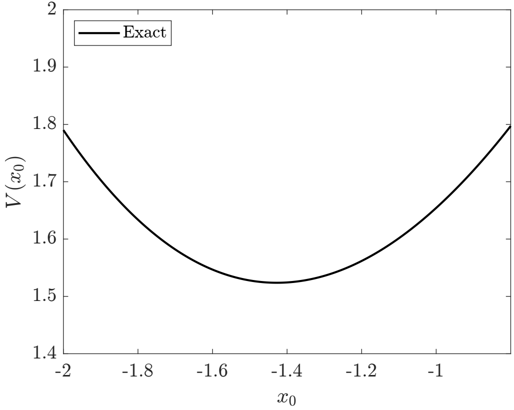
Continuous-time OCP

$$\min_{x(\cdot),\lambda(\cdot),s(\cdot)} \int_0^2 x(t)^2 \mathrm{d}t + (x(2) - 5/3)^2$$
$$\text{s.t.} \quad \dot{x}(t) = 2 - s(t)$$
$$0 \leq \lambda(t) - x(t) \perp 1 + s(t) \geq 0$$
$$0 \leq \lambda(t) \perp 1 - s(t) \geq 0, \; t \in [0,2]$$

Free initial value $x(0)$ is the effective degree of freedom.

Equivalent reduced problem

$$\min_{x_0 \in \mathbb{R}} V(x_0)$$



Denote by $V(x_0)$ the nonsmooth objective value for the unique feasible trajectory starting at $x(0) = x_0$.

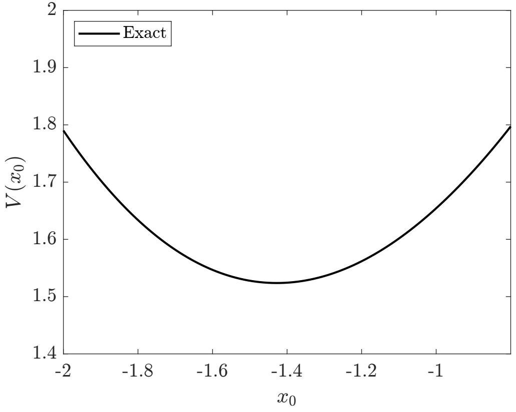
1. Direct optimal control with a time stepping IRK discretization

Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

Continuous-time OCP

$$\begin{aligned} \min_{x(\cdot),\lambda(\cdot),s(\cdot)} \quad & \int_0^2 x(t)^2 \mathrm{d}t + (x(2) - 5/3)^2 \\ \text{s.t.} \quad & \dot{x}(t) = 2 - s(t) \\ & 0 \leq \lambda(t) - x(t) \perp 1 + s(t) \geq 0 \\ & 0 \leq \lambda(t) \perp 1 - s(t) \geq 0, \ t \in [0,2] \end{aligned}$$

- discretize the DCS with fixed step size IRK methods
- e.g., midpoint rule, Gauss-Legendre IRK with $n_{\mathrm{s}} = 1$, accuracy $O(h^2)$



Locally quadratic objective.

1. Direct optimal control with a time stepping IRK discretization

Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

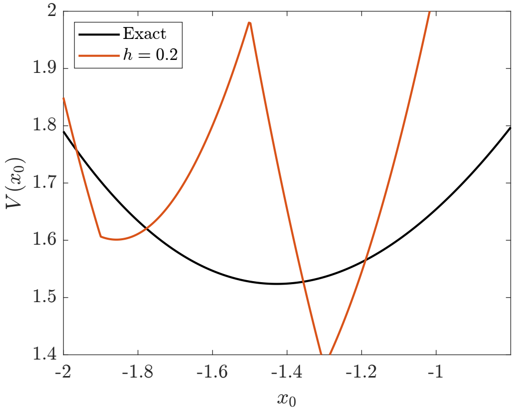
Discrete-time OCP

$$\min_{\mathbf{x},\mathbf{z}} \quad \sum_{n=0}^{N-1} \ell_n(x_n) + (x_N - 5/3)^2$$

$$\text{s.t.} \quad x_{n+1} = \phi_f(x_n, z_n)$$

$$0 = \phi_{\text{int}}(x_n, z_n), \; n = 0, \dots N-1$$

- discretize the DCS with fixed step size IRK methods
- e.g., midpoint rule, Gauss-Legendre IRK with $n_{\mathrm{s}} = 1$, accuracy $O(h^2)$
- step size $h = 0.2$, i.e., $N = 10$ integration steps



Many artificial local minima and wrong derivatives.

1. Direct optimal control with a time stepping IRK discretization

Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

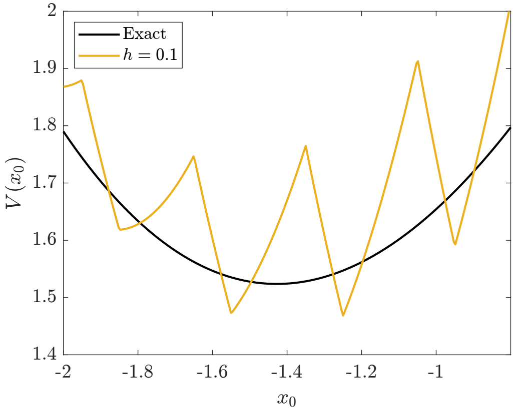
Discrete-time OCP

$$\min_{\mathbf{x},\mathbf{z}} \quad \sum_{n=0}^{N-1} \ell_n(x_n) + (x_N - 5/3)^2$$

$$\text{s.t.} \quad x_{n+1} = \phi_f(x_n, z_n)$$

$$0 = \phi_{\text{int}}(x_n, z_n), \ n = 0, \dots N-1$$

- discretize the DCS with fixed step size IRK methods
- e.g., midpoint rule, Gauss-Legendre IRK with $n_s = 1$, accuracy $O(h^2)$
- step size $h = 0.1$, i.e., $N = 20$ integration steps



Many artificial local minima and wrong derivatives.

1. Direct optimal control with a time stepping IRK discretization

Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

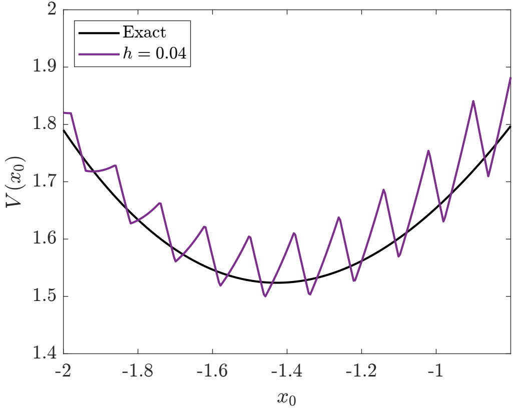
Discrete-time OCP

$$\min_{\mathbf{x},\mathbf{z}} \quad \sum_{n=0}^{N-1} \ell_n(x_n) + (x_N - 5/3)^2$$

$$\text{s.t.} \quad x_{n+1} = \phi_f(x_n, z_n)$$

$$0 = \phi_{\text{int}}(x_n, z_n), \;\; n = 0, \dots N-1$$

- discretize the DCS with fixed step size IRK methods
- e.g., midpoint rule, Gauss-Legendre IRK with $n_{\mathrm{s}} = 1$, accuracy $O(h^2)$
- step size $h = 0.04$, i.e., $N = 50$ integration steps



Many artificial local minima and wrong derivatives.

1. Direct optimal control with a time stepping IRK discretization

Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

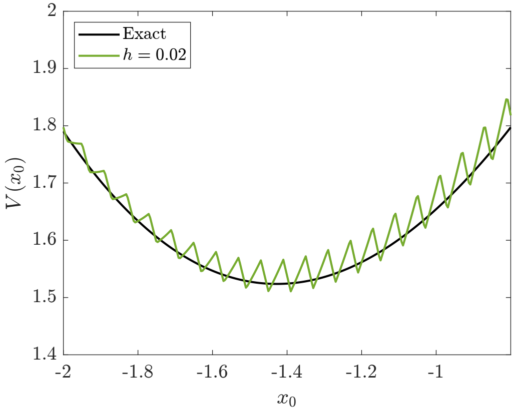
Discrete-time OCP

$$\min_{\mathbf{x},\mathbf{z}} \quad \sum_{n=0}^{N-1} \ell_n(x_n) + (x_N - 5/3)^2$$

$$\text{s.t.} \quad x_{n+1} = \phi_f(x_n, z_n)$$

$$0 = \phi_{\text{int}}(x_n, z_n), \;\; n = 0, \dots N-1$$

- discretize the DCS with fixed step size IRK methods
- e.g., midpoint rule, Gauss-Legendre IRK with $n_{\text{s}} = 1$, accuracy $O(h^2)$
- step size $h = 0.02$, i.e., $N = 100$ integration steps



Many artificial local minima and wrong derivatives.

Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

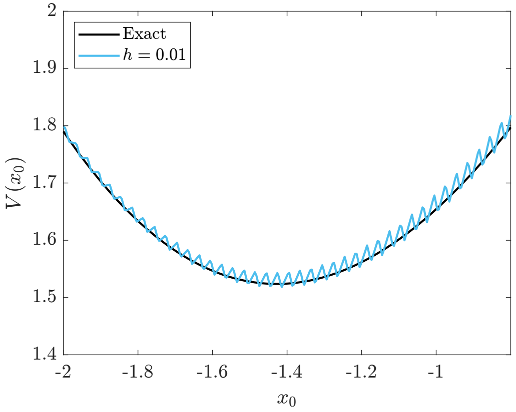
Discrete-time OCP

$$\min_{\mathbf{x},\mathbf{z}} \quad \sum_{n=0}^{N-1} \ell_n(x_n) + (x_N - 5/3)^2$$

$$\text{s.t.} \quad x_{n+1} = \phi_f(x_n, z_n)$$

$$0 = \phi_{\text{int}}(x_n, z_n), \ n = 0, \dots N-1$$

- discretize the DCS with fixed step size IRK methods
- e.g., midpoint rule, Gauss-Legendre IRK with $n_{\text{s}} = 1$, accuracy $O(h^2)$
- step size $h = 0.01$, i.e., $N = 200$ integration steps



Many artificial local minima and wrong derivatives.

1. Direct optimal control with a time stepping IRK discretization

Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

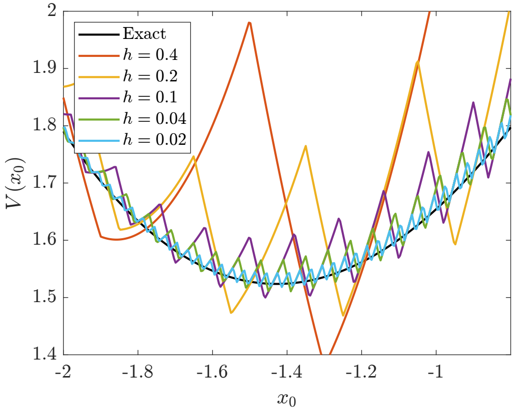
Discrete-time OCP

$$\min_{\mathbf{x},\mathbf{z}} \quad \sum_{n=0}^{N-1} \ell_n(x_n) + (x_N - 5/3)^2$$

$$\text{s.t.} \quad x_{n+1} = \phi_f(x_n, z_n)$$

$$0 = \phi_{\text{int}}(x_n, z_n), \; n = 0, \ldots N-1$$

- discretize the DCS with fixed step size IRK methods
- e.g., midpoint rule, Gauss-Legendre IRK with $n_\text{s} = 1$, accuracy $O(h^2)$
- decreasing the step size might worsen the situation



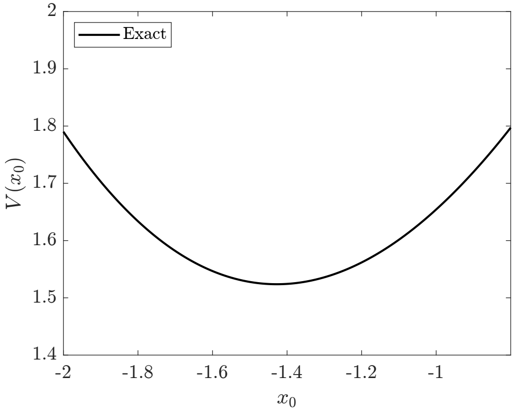
Many artificial local minima and wrong derivatives.

Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

Continuous-time OCP

$$\min_{x(\cdot)\in\mathcal{C}^0([0,2])} \int_0^2 x(t)^2 \mathrm{d}t + (x(2) - 5/3)^2$$
$$\text{s.t.} \quad \dot{x}(t) = 2 - \operatorname{sign}(x(t)), \quad t \in [0,2]$$

- midpoint rule, with $h = 0.05$; $N = 40$



2. Direct optimal control with a standard IRK discretization - smoothing

Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

Smoothed continuous-time OCP

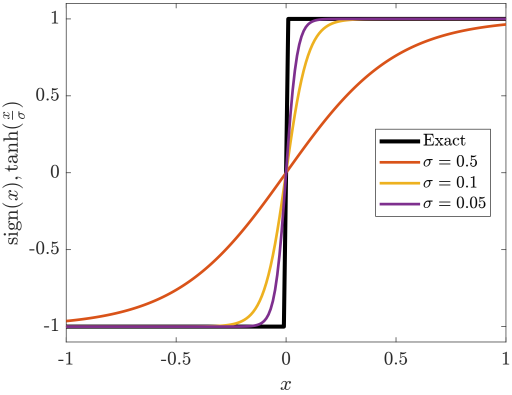
$$\min_{x(\cdot)\in\mathcal{C}^\infty([0,2])} \int_0^2 x(t)^2 \mathrm{d}t + (x(2)-5/3)^2$$

$$\text{s.t.} \quad \dot{x}(t) = 2 - \tanh\left(\frac{x(t)}{\sigma}\right), \quad t \in [0,2]$$

Equivalent reduced problem

$$\min_{x_0\in\mathbb{R}} V_\sigma(x_0)$$

- midpoint rule, with $h = 0.05$; $N = 40$
- solve smoothed OCP for different σ



Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

Smoothed continuous-time OCP

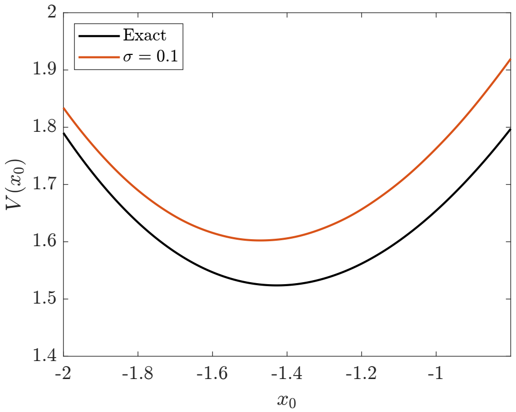
$$\min_{x(\cdot)\in\mathcal{C}^\infty([0,2])} \int_0^2 x(t)^2 \mathrm{d}t + (x(2) - 5/3)^2$$

$$\text{s.t.} \quad \dot{x}(t) = 2 - \tanh\left(\frac{x(t)}{\sigma}\right), \quad t \in [0,2]$$

Equivalent reduced problem

$$\min_{x_0\in\mathbb{R}} V_\sigma(x_0)$$

- midpoint rule, with $h = 0.05;\ N = 40$
- solve smoothed OCP with $\sigma = 0.1$



2. Direct optimal control with a standard IRK discretization - smoothing

Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

Smoothed continuous-time OCP

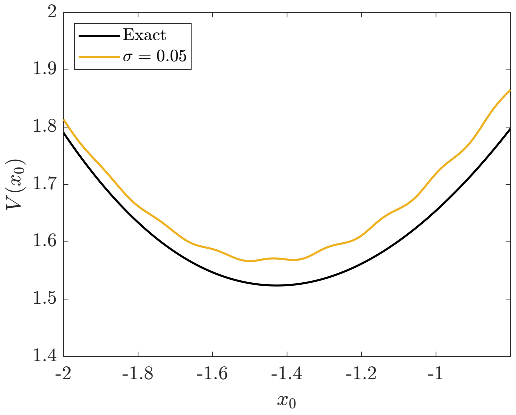
$$\min_{x(\cdot)\in\mathcal{C}^\infty([0,2])} \int_0^2 x(t)^2 \mathrm{d}t + (x(2)-5/3)^2$$

$$\text{s.t.} \quad \dot{x}(t) = 2 - \tanh\left(\frac{x(t)}{\sigma}\right), \quad t \in [0,2]$$

Equivalent reduced problem

$$\min_{x_0\in\mathbb{R}} V_\sigma(x_0)$$

- midpoint rule, with $h = 0.05$; $N = 40$
- solve smoothed OCP with $\sigma = 0.05$



2. Direct optimal control with a standard IRK discretization - smoothing

Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

Smoothed continuous-time OCP

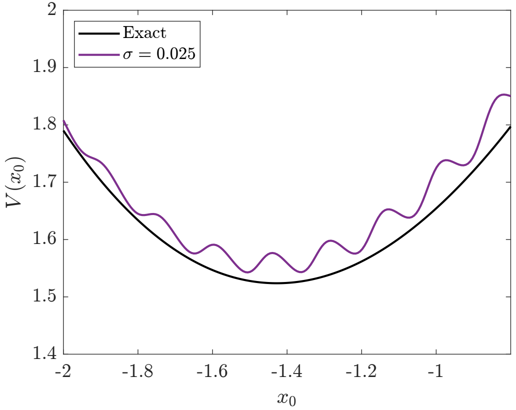
$$\min_{x(\cdot)\in\mathcal{C}^{\infty}([0,2])} \int_0^2 x(t)^2 \mathrm{d}t + (x(2) - 5/3)^2$$

$$\text{s.t.} \quad \dot{x}(t) = 2 - \tanh\left(\frac{x(t)}{\sigma}\right), \quad t \in [0,2]$$

Equivalent reduced problem

$$\min_{x_0 \in \mathbb{R}} V_{\sigma}(x_0)$$

- midpoint rule, with $h = 0.05;\ N = 40$
- solve smoothed OCP with $\sigma = 0.025$



Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

Smoothed continuous-time OCP

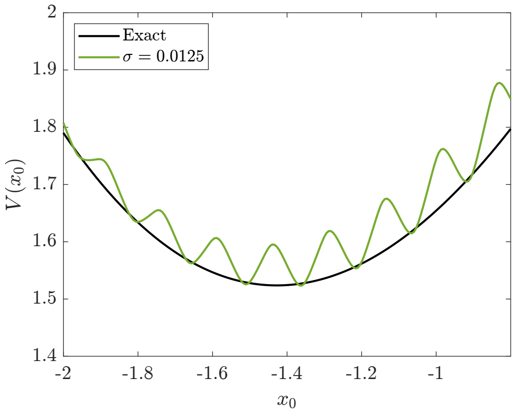
$$\min_{x(\cdot)\in\mathcal{C}^\infty([0,2])} \int_0^2 x(t)^2 \mathrm{d}t + (x(2) - 5/3)^2$$

$$\text{s.t.} \quad \dot{x}(t) = 2 - \tanh\left(\frac{x(t)}{\sigma}\right), \quad t \in [0,2]$$

Equivalent reduced problem

$$\min_{x_0\in\mathbb{R}} V_\sigma(x_0)$$

- midpoint rule, with $h = 0.05$; $N = 40$
- solve smoothed OCP with $\sigma = 0.0125$



Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

Smoothed continuous-time OCP

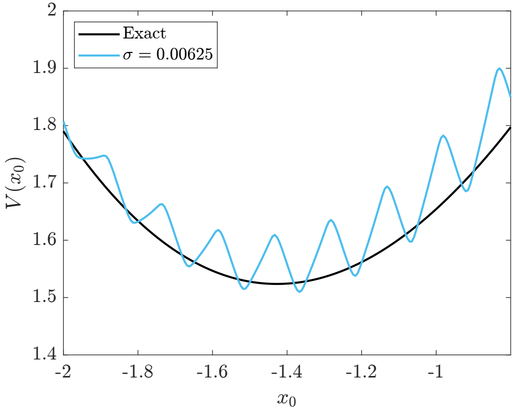
$$\min_{x(\cdot)\in\mathcal{C}^\infty([0,2])} \int_0^2 x(t)^2 \mathrm{d}t + (x(2)-5/3)^2$$

$$\text{s.t.} \quad \dot{x}(t) = 2 - \tanh\left(\frac{x(t)}{\sigma}\right), \quad t \in [0,2]$$

Equivalent reduced problem

$$\min_{x_0\in\mathbb{R}} V_\sigma(x_0)$$

- midpoint rule, with $h = 0.05$; $N = 40$
- solve smoothed OCP with $\sigma = 0.00625$



Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

Smoothed continuous-time OCP

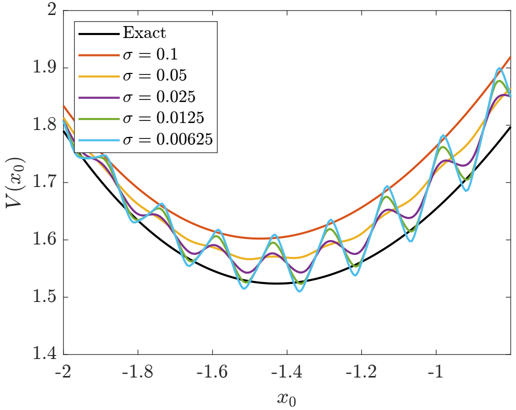
$$\min_{x(\cdot)\in\mathcal{C}^\infty([0,2])} \int_0^2 x(t)^2 \mathrm{d}t + (x(2)-5/3)^2$$

$$\text{s.t.} \quad \dot{x}(t) = 2 - \tanh\left(\frac{x(t)}{\sigma}\right), \quad t \in [0,2]$$

Equivalent reduced problem

$$\min_{x_0 \in \mathbb{R}} V_\sigma(x_0)$$

- midpoint rule, with $h = 0.05;\ N = 40$



If $h \gg \sigma$, then the smooth approximation behaves **the same as the nonsmooth problem!**

Tutorial example inspired [Stewart and Anitescu, 2010]. Further studied in [Nurkanović et al., 2020].

Smoothed continuous-time OCP

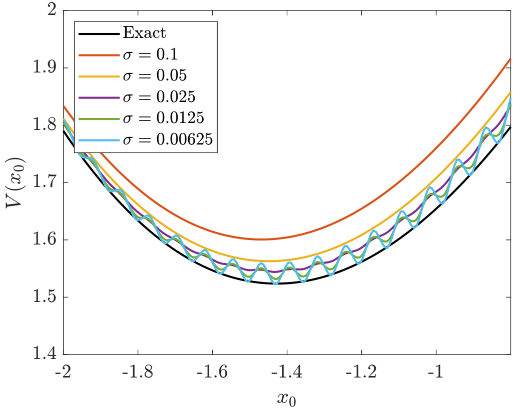
$$\min_{x(\cdot)\in\mathcal{C}^\infty([0,2])} \int_0^2 x(t)^2 \mathrm{d}t + (x(2) - 5/3)^2$$

$$\text{s.t.} \quad \dot{x}(t) = 2 - \tanh\left(\frac{x(t)}{\sigma}\right), \quad t \in [0,2]$$

Equivalent reduced problem

$$\min_{x_0\in\mathbb{R}} V_\sigma(x_0)$$

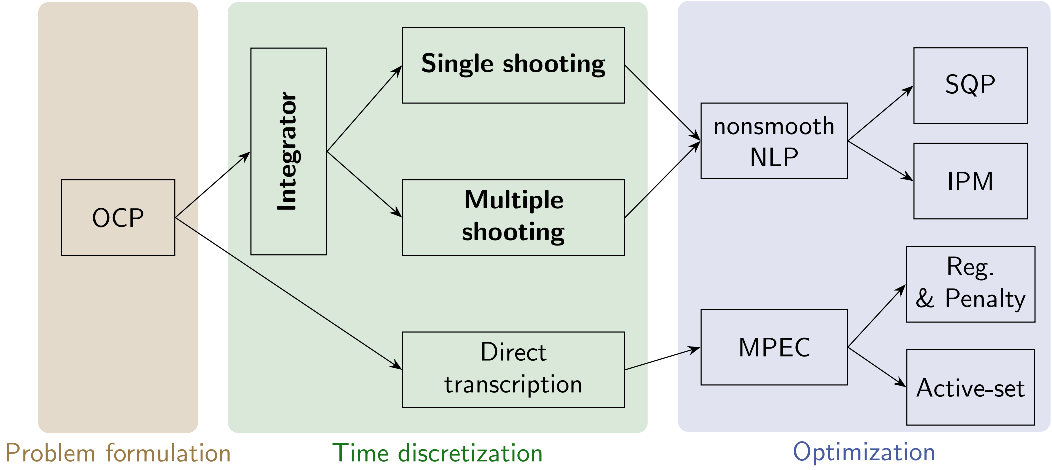
- midpoint rule, with $h = 0.025$; $N = 80$



If $h \gg \sigma$, then the smooth approximation behaves **the same as the nonsmooth problem!**

Work flow in nonsmooth direct optimal control

First discretize, then optimize.



OCP = Optimal Control Problem

NLP = Nonlinear Program

MPEC = Mathematical Program with Equilibrium Constraints

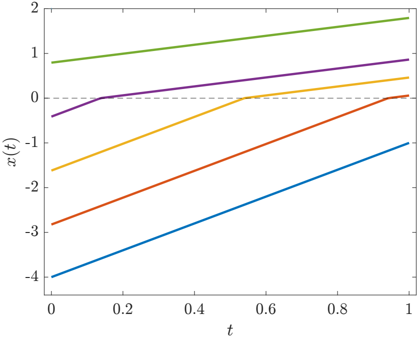
SQP = Sequential Quadratic Programming

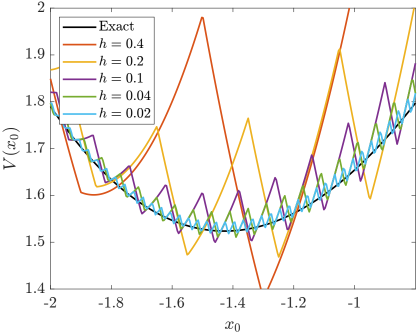
IPM = Interior-Point Method

Reg. = Regularization

We need to:

1. Use a switch detecting integration method: restore accuracy of integration method.
2. Compute derivatives correctly.





Computing derivatives of a discrete time system

Regard an ODE, for simplicity without a control:

$$x(t) = f(x(t)),\ x(t_k) = x_k,\ t \in [t_k, t_{k+1}]$$

In direct optimal control, with the use of an integrator we regard:

$$x_{k+1} = \psi(x_k)$$

Computing derivatives of a discrete time system

Regard an ODE, for simplicity without a control:

$$x(t) = f(x(t)),\ x(t_k) = x_k,\ t \in [t_k, t_{k+1}]$$

In direct optimal control, with the use of an integrator we regard:

$$x_{k+1} = \psi(x_k)$$

In Newton-type optimization we need to linearize this equation, e.g., at a feasible point $(\bar{x}_{k+1}, \bar{x}_k)$:

$$0 = \psi(\bar{x}_k) - \bar{x}_{k+1} + \frac{\partial \psi(\bar{x}_k)}{\partial x}(\hat{x}_k - \bar{x}_k) - (\hat{x}_{k+1} - \bar{x}_{k+1})$$

$$0 = \underbrace{\psi(\bar{x}_k) - \bar{x}_{k+1}}_{=0} + \frac{\partial \psi(\bar{x}_k)}{\partial x}\Delta x_k - \Delta x_{k+1}$$

Computing derivatives of a discrete time system

Regard an ODE, for simplicity without a control:

$$x(t) = f(x(t)),\ x(t_k) = x_k,\ t \in [t_k, t_{k+1}]$$

In direct optimal control, with the use of an integrator we regard:

$$x_{k+1} = \psi(x_k)$$

In Newton-type optimization we need to linearize this equation, e.g., at a feasible point $(\bar{x}_{k+1}, \bar{x}_k)$:

$$0 = \psi(\bar{x}_k) - \bar{x}_{k+1} + \frac{\partial \psi(\bar{x}_k)}{\partial x}(\hat{x}_k - \bar{x}_k) - (\hat{x}_{k+1} - \bar{x}_{k+1})$$

$$0 = \underbrace{\psi(\bar{x}_k) - \bar{x}_{k+1}}_{=0} + \frac{\partial \psi(\bar{x}_k)}{\partial x}\Delta x_k - \Delta x_{k+1}$$

Change in final state by change initial state described by:

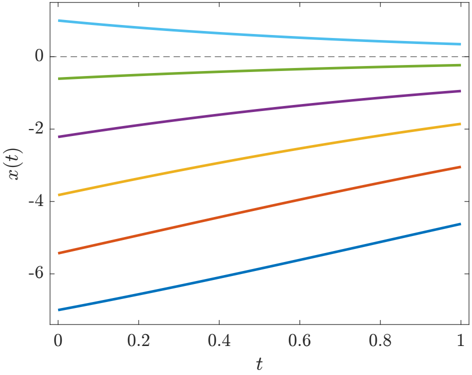
$$\Delta x_{k+1} = \frac{\partial \psi(\bar{x}_k)}{\partial x}\Delta x_k$$

Here $S(t) = \frac{\mathrm{d}x(t;x_k)}{\mathrm{d}x_k}$, i.e., $S(t_{k+1}) = \frac{\mathrm{d}x(t_{k+1};x_k)}{\mathrm{d}x_k} = \frac{\mathrm{d}\phi(x_k)}{\mathrm{d}x_k}$ is the **sensitivity matrix**.

Geometric interpretation of the sensitivity matrix $S(t)$ - smooth case

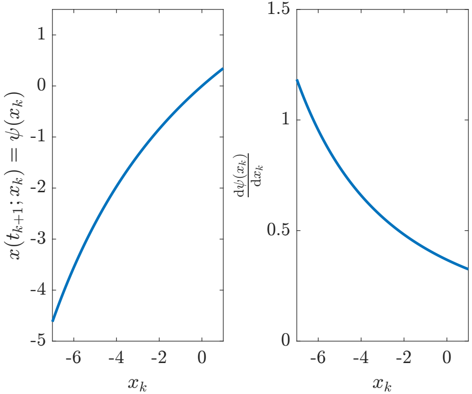
Consider the smooth ODE:

$$\dot{x} = -x - 0.2x^2$$



Trajectory examples.

$$\Delta x_{k+1} = \frac{\partial \psi(\bar{x}_k)}{\partial x} \Delta x_k$$



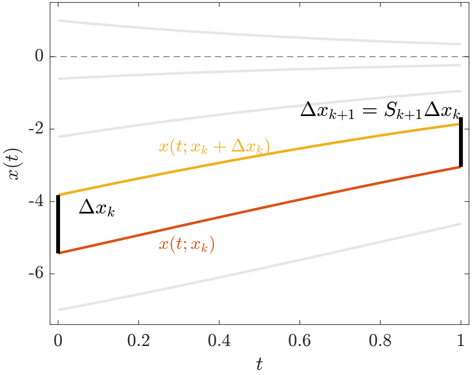
Solution and sensitivity maps.

Geometric interpretation of the sensitivity matrix $S(t)$ - smooth case

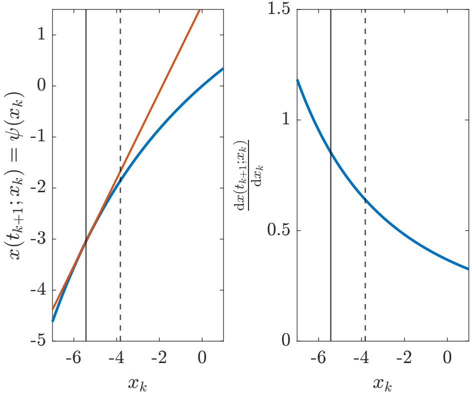
Consider the smooth ODE:

$$\dot{x} = -x - 0.2x^2$$

$$\Delta x_{k+1} = \frac{\partial \psi(\bar{x}_k)}{\partial x} \Delta x_k$$



Linear approximation at t_{k+1}.



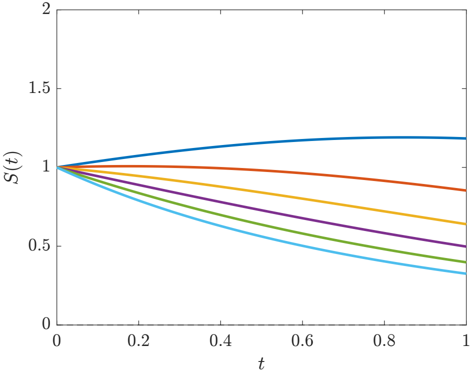
Linear approximation of solution map.

Geometric interpretation of the sensitivity matrix $S(t)$ - smooth case

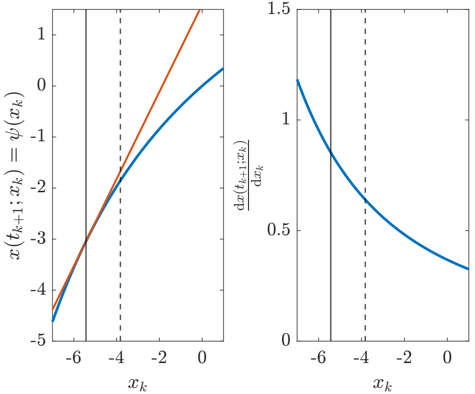
Consider the smooth ODE:

$$\dot{x} = -x - 0.2x^2$$

$$\Delta x_{k+1} = \frac{\partial \psi(\bar{x}_k)}{\partial x} \Delta x_k$$



Sensitivities in time.



Linear approximation of solution map.

Computation of $S(t)$ - smooth case

An excellent reference on sensitivity computation is the PhD Thesis [Quirynen, 2017].

$$S(t) = \frac{\mathrm{d}x(t;x_k)}{\mathrm{d}x_k}, t > t_k, \quad S(t_k) = \frac{\mathrm{d}x(t_k;x_k)}{\mathrm{d}x_k} = \frac{\mathrm{d}x_k}{\mathrm{d}x_k} = I \text{ (initial value)}$$

Computation of $S(t)$ - smooth case

An excellent reference on sensitivity computation is the PhD Thesis [Quirynen, 2017].

$$S(t) = \frac{\mathrm{d}x(t;x_k)}{\mathrm{d}x_k}, t > t_k, \quad S(t_k) = \frac{\mathrm{d}x(t_k;x_k)}{\mathrm{d}x_k} = \frac{\mathrm{d}x_k}{\mathrm{d}x_k} = I \text{ (initial value)}$$

We take the first differentiate, then integrate:

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\mathrm{d}x(t;x_k)}{\mathrm{d}x_k} = \frac{\mathrm{d}\dot{x}(t;x_k)}{\mathrm{d}x_k}$$

$$= \frac{\mathrm{d}f(x(t;x_k))}{\mathrm{d}x_k} = \frac{\partial f(x(t;x_k))}{\partial x_k}\frac{\mathrm{d}x(t;x_k)}{\mathrm{d}x_k} = \frac{\partial f(x(t;x_k))}{\partial x_k}S(t)$$

Computation of $S(t)$ - smooth case

An excellent reference on sensitivity computation is the PhD Thesis [Quirynen, 2017].

$$S(t) = \frac{\mathrm{d}x(t;x_k)}{\mathrm{d}x_k}, t > t_k, \quad S(t_k) = \frac{\mathrm{d}x(t_k;x_k)}{\mathrm{d}x_k} = \frac{\mathrm{d}x_k}{\mathrm{d}x_k} = I \text{ (initial value)}$$

We take the first differentiate, then integrate:

$$\begin{aligned}\frac{\mathrm{d}S(t)}{\mathrm{d}t} &= \frac{\mathrm{d}}{\mathrm{d}t}\frac{\mathrm{d}x(t;x_k)}{\mathrm{d}x_k} = \frac{\mathrm{d}\dot{x}(t;x_k)}{\mathrm{d}x_k} \\ &= \frac{\mathrm{d}f(x(t;x_k))}{\mathrm{d}x_k} = \frac{\partial f(x(t;x_k))}{\partial x_k}\frac{\mathrm{d}x(t;x_k)}{\mathrm{d}x_k} = \frac{\partial f(x(t;x_k))}{\partial x_k}S(t)\end{aligned}$$

Then, jointly integrate:

$$\begin{pmatrix}\dot{x}(t) \\ \dot{S}(t)\end{pmatrix} = \begin{pmatrix} f(x(t)) \\ \frac{\partial f(x(t))}{\partial x}S(t)\end{pmatrix}, \quad x(t_k) = x_k, \; S(t_k) = S_k$$

Computation of $S(t)$ - smooth case

An excellent reference on sensitivity computation is the PhD Thesis [Quirynen, 2017].

$$S(t) = \frac{\mathrm{d}x(t;x_k)}{\mathrm{d}x_k}, t > t_k, \quad S(t_k) = \frac{\mathrm{d}x(t_k;x_k)}{\mathrm{d}x_k} = \frac{\mathrm{d}x_k}{\mathrm{d}x_k} = I \text{ (initial value)}$$

We take the first differentiate, then integrate:

$$\begin{aligned}\frac{\mathrm{d}S(t)}{\mathrm{d}t} &= \frac{\mathrm{d}}{\mathrm{d}t}\frac{\mathrm{d}x(t;x_k)}{\mathrm{d}x_k} = \frac{\mathrm{d}\dot{x}(t;x_k)}{\mathrm{d}x_k} \\ &= \frac{\mathrm{d}f(x(t;x_k))}{\mathrm{d}x_k} = \frac{\partial f(x(t;x_k))}{\partial x_k}\frac{\mathrm{d}x(t;x_k)}{\mathrm{d}x_k} = \frac{\partial f(x(t;x_k))}{\partial x_k} S(t)\end{aligned}$$

Then, jointly integrate:

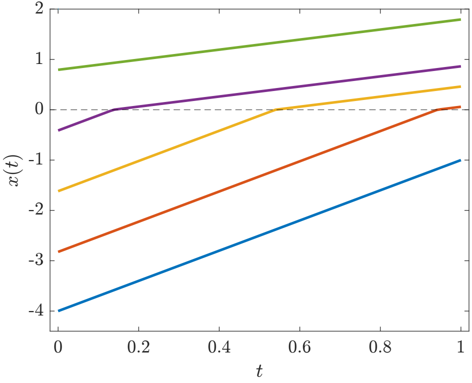
$$\begin{pmatrix}\dot{x}(t) \\ \dot{S}(t)\end{pmatrix} = \begin{pmatrix} f(x(t)) \\ \frac{\partial f(x(t))}{\partial x} S(t)\end{pmatrix}, \quad x(t_k) = x_k,\ S(t_k) = S_k$$

But if the function $f(x)$ is not differentiable in x?

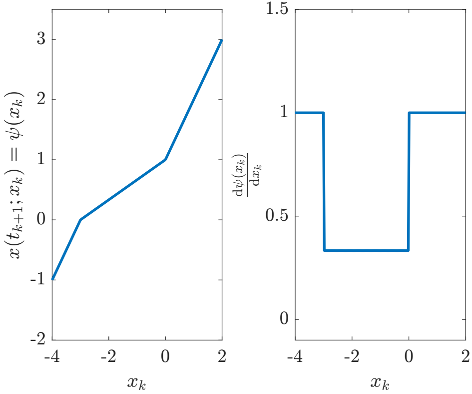
Consider the nonsmooth ODE:

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

$$\Delta x_{k+1} = \frac{\partial \psi(\bar{x}_k)}{\partial x} \Delta x_k$$



Solution has kinks in time.

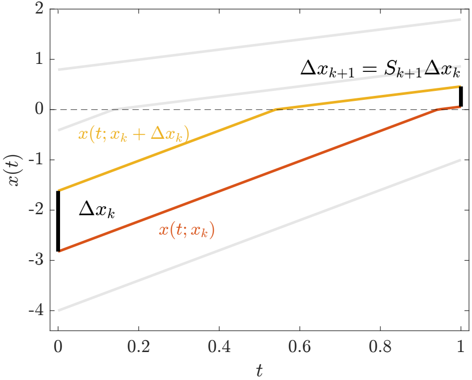


Solution map has kinks.

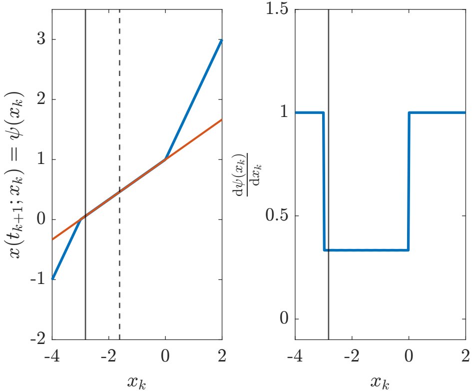
Consider the nonsmooth ODE:

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

$$\boxed{\Delta x_{k+1} = \frac{\partial \psi(\bar{x}_k)}{\partial x} \Delta x_k}$$



Solution has kinks in time.

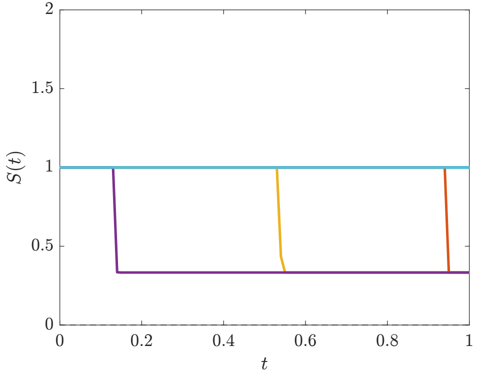


Sensitivity discontinuous.

Geometric interpretation of the sensitivity matrix $S(t)$ - **nonsmooth** case

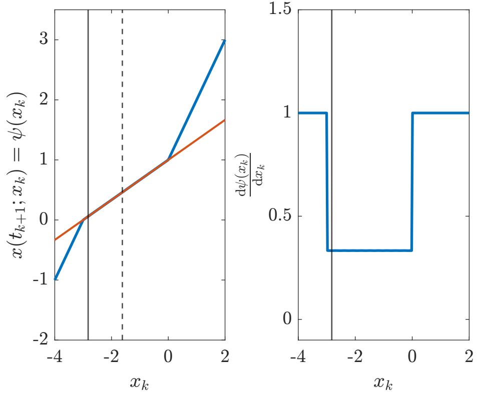
Consider the nonsmooth ODE:

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



Sensitivities jump in time.

$$\Delta x_{k+1} = \frac{\partial \psi(\bar{x}_k)}{\partial x} \Delta x_k$$

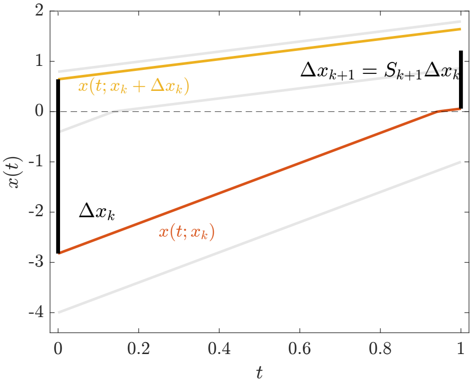


Linearization wrong across switches.

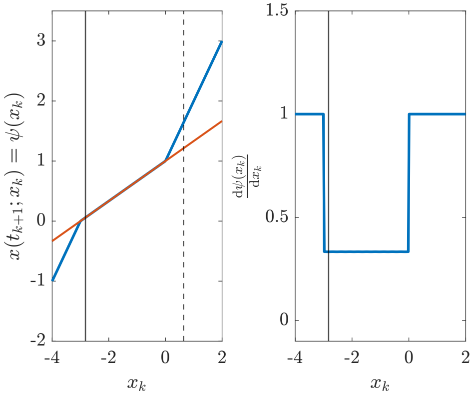
Consider the nonsmooth ODE:

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

$$\Delta x_{k+1} = \frac{\partial \psi(\bar{x}_k)}{\partial x} \Delta x_k$$



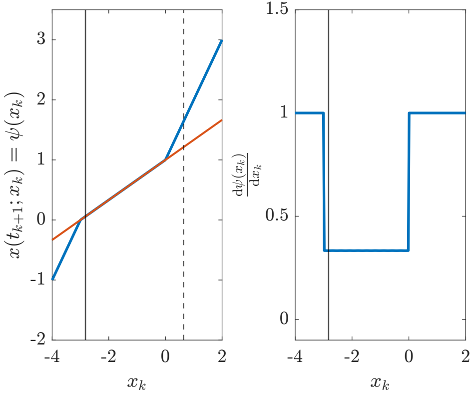
Large error.



Linearization wrong across switches.

Summary on sensitivities of nonsmooth systems

- Solution map $\phi(x_k)$ has kinks, sensitivity $\frac{\mathrm{d}\phi(x_k)}{\mathrm{d}x_k}$ jumps
- The discontinuity of $f(x(t))$ introduces also jumps in $S(t)$ in time
- Linearization can be arbitrarily wrong, if there are changes of switches
- Correct computation of $S(t)$ requires switch detection and updates



Computation of $S(t)$ - nonsmooth case

Regard a bimodal system:

$$\dot{x}(t) = \begin{cases} f_1(x(t)), & \psi(x(t)) < 0, \\ f_2(x(t)), & \psi(x(t)) \geq 0. \end{cases} \tag{1}$$

At some t_{s} trajectory $x(t)$ crosses switching surface $\psi(x) = 0$, e.g.:

- before crossing: $\dot{x} = f_1(x)$ for $t \in [0, t_{\mathrm{s}})$, with solution $x_1(t; x_0)$
- after crossing: t_{s} we have $\dot{x} = f_2(x)$ for $t \in (t_{\mathrm{s}}, T]$ with solution $x_2(t; x_1(t_{\mathrm{s}}; x_0))$

Trajectory pieces $x_1(t)$ and $x_2(t)$ glued together by condition:

$$\psi(x_1(t_{\mathrm{s}}(x_0); x_0)) = 0.$$

Computing sensitivity for:

- $t < t_{\mathrm{s}}$ - just like in the smooth case;
- $t > t_{\mathrm{s}}$ - everything depends implicitly on the switching times $t_{\mathrm{s}}(x_0)$

Computation of $S(t)$ via the Saltation matrix

Before and after the switch the $S(t)$ obey linear variational differential equation (VDE)

$$\dot{S}_i(t) = \frac{\partial f_i(x)}{\partial x} S_i(t), \ \ i = 1, 2$$

The function $S(t)$ obeys smooth VDEs, on both sides of t_s, but exhibits a jump at t_s.

Proposition

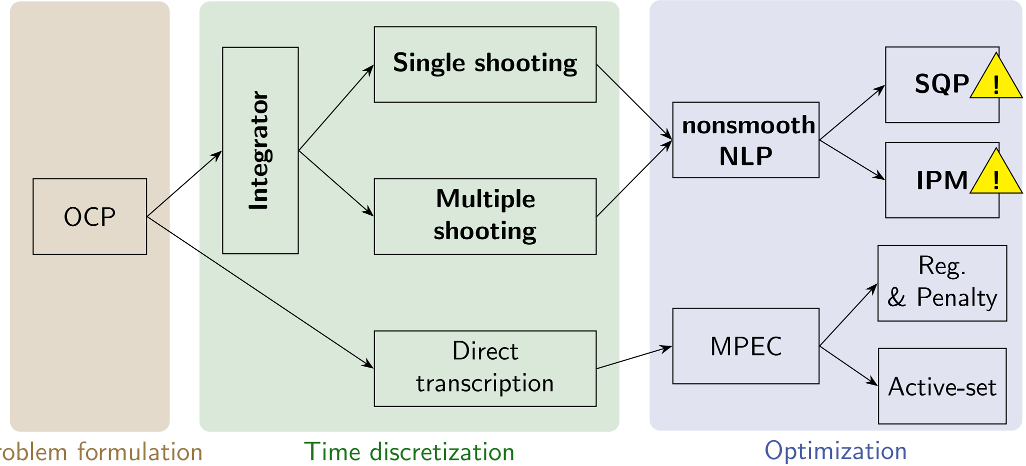
Regard the system (1) *with* $x(0) = x_0 \in R_i$ *on an interval* $[0, T]$ *with a switch at* $t_\mathrm{s} \in (0, T)$. *Assume that the functions* $f_1(x), \ f_2(x), \psi_{i,j}(x)$ *are continuously differentiable along* $x(t), t \in [0, T]$. *Assume the solution* $x(t)$ *reaches the surface of discontinuity transversally, i.e.,* $\nabla \psi(x(t_\mathrm{s}))^\top f_1(x(t_\mathrm{s})) > 0$. *Then the sensitivity* $S(T; 0)$ *of a solution* $x(t; x_0)$ *of the system described by the ODE* (1) *is given by*

$$S(T; 0) = S(T; t_\mathrm{s}^+) J(x(t_\mathrm{s}; x_0)) S(t_\mathrm{s}^-; 0) \ \textit{with}$$

$$J(x(t_\mathrm{s}; x_0)) \coloneqq I + \frac{(f_2(x(t_\mathrm{s}; x_0)) - f_1(x(t_\mathrm{s}; x_0))) \nabla \psi(x(t_\mathrm{s}; x_0))^\top}{\nabla \psi(x(t_\mathrm{s}; x_0))^\top f_1(x(t_\mathrm{s}; x_0))}.$$

Work flow in nonsmooth direct optimal control

First discretize, then optimize.



OCP = Optimal Control Problem

NLP = Nonlinear Program

MPEC = Mathematical Program with Equilibrium Constraints

SQP = Sequential Quadratic Programming

IPM = Interior-Point Method

Reg. = Regularization

Switch detecting integrators in shooting methods is an old idea

Some historical references:

- 1950s - First derivations of the Saltation matrix [Aizerman and Gantmakher, 1958]
- 1980s - Multiple shooting, switch detection, first described in PhD thesis of Hans Georg Bock [Bock, 1987] (multiple shooting introduce by Bock and Plitt [Bock and Plitt, 1984])
- 2000s - Other attempts with multiple shooting and SQP: with single step methods (RK4) [Kirches, 2006], with multi-step methods (BDF) [Brandt-Pollmann, 2004]
- ...
- 2020s: More recent in robotics, hybrid iLQR [Kong et al., 2021], [Kong et al., 2024]

Switch detecting integrators in shooting methods is an old idea

Some historical references:

- 1950s - First derivations of the Saltation matrix [Aizerman and Gantmakher, 1958]
- 1980s - Multiple shooting, switch detection, first described in PhD thesis of Hans Georg Bock [Bock, 1987] (multiple shooting introduce by Bock and Plitt [Bock and Plitt, 1984])
- 2000s - Other attempts with multiple shooting and SQP: with single step methods (RK4) [Kirches, 2006], with multi-step methods (BDF) [Brandt-Pollmann, 2004]
- ...
- 2020s: More recent in robotics, hybrid iLQR [Kong et al., 2021], [Kong et al., 2024]

Can work sometimes quite good, but why not established yet?

Newton's method applied to nonsmooth optimization problems

For simplicity we consider

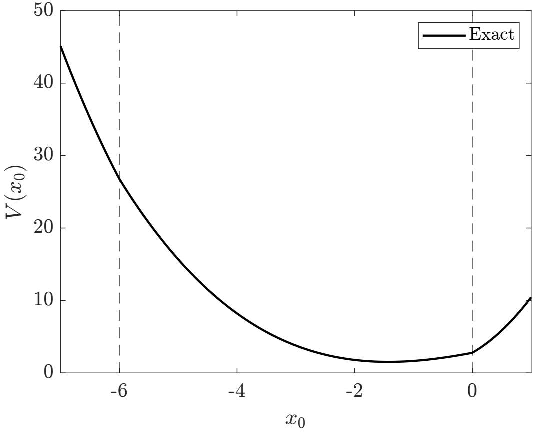
$$\min_{w} F(w)$$

where $F : \mathbb{R}^n \to \mathbb{R}$ is a piecewise smooth function. Without constraints, KKT conditions reduce to

$$\nabla F(w) = 0$$

SQP and IPM reduce to Newton's method and read as

$$w^{k+1} = w^k - [\nabla^2 F(w^k)]^{-1} \nabla F(w^k)$$

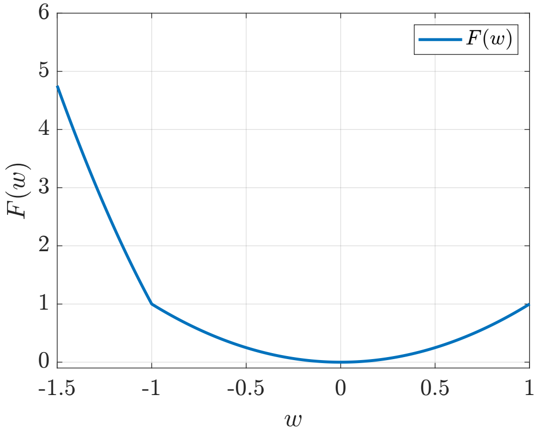


Nonsmooth optimization examples: convex kink not at solution

$$\min_{w} F(w)$$

where

$$F(w) = \begin{cases} 3w^2 - 2, & w < -1 \\ w^2, & w > -1 \end{cases}$$



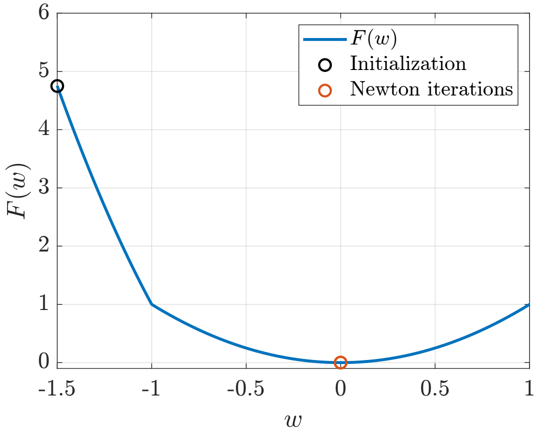
Nonsmooth optimization examples: convex kink not at solution

$$\min_{w} F(w)$$

where

$$F(w) = \begin{cases} 3w^2 - 2, & w < -1 \\ w^2, & w > -1 \end{cases}$$

- Sometimes the Newton steps "skip" over a convex kink.



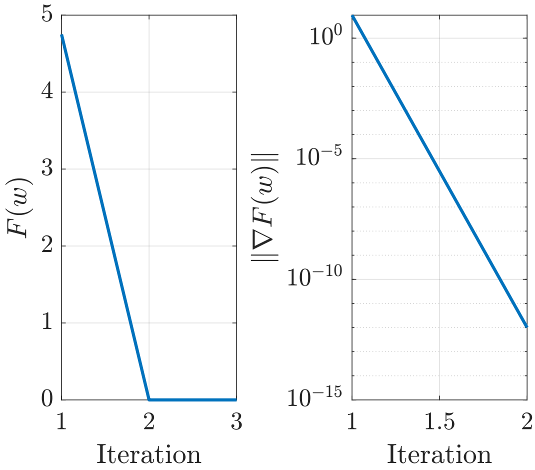
Nonsmooth optimization examples: convex kink not at solution

$$\min_{w} F(w)$$

where

$$F(w) = \begin{cases} 3w^2 - 2, & w < -1 \\ w^2, & w > -1 \end{cases}$$

- Sometimes the Newton steps "skip" over a convex kink.

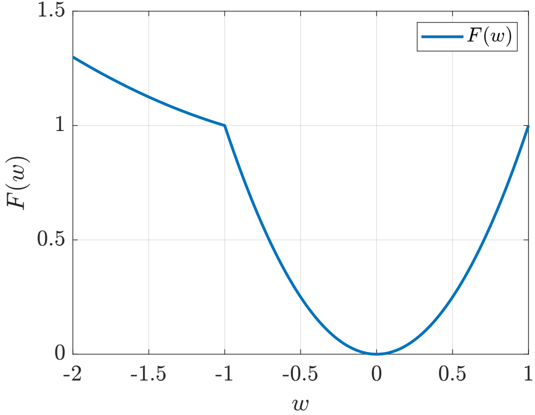


Nonsmooth optimization examples: concave kink not at solution

$$\min_w F(w)$$

where

$$F(w) = \begin{cases} 0.1w^2 - 0.9, & w < -1 \\ w^2, & w > -1 \end{cases}$$



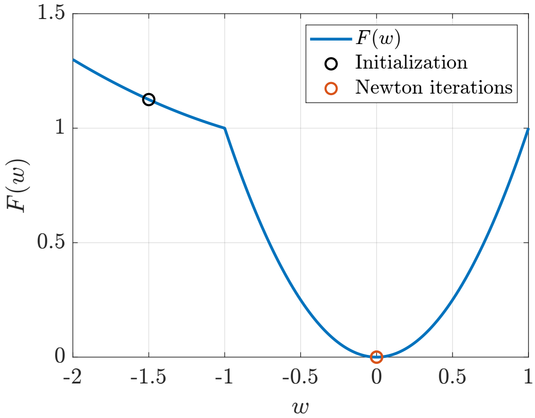
Nonsmooth optimization examples: concave kink not at solution

$$\min_{w} F(w)$$

where

$$F(w) = \begin{cases} 0.1w^2 - 0.9, & w < -1 \\ w^2, & w > -1 \end{cases}$$

- Sometimes the Newton steps “skip” over a concave kink.



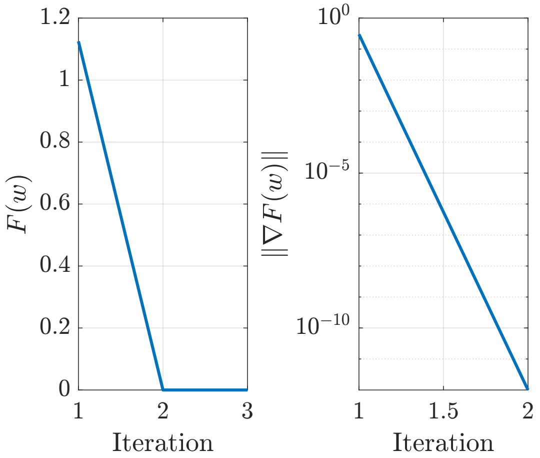
Nonsmooth optimization examples: concave kink not at solution

$$\min_w F(w)$$

where

$$F(w) = \begin{cases} 0.1w^2 - 0.9, & w < -1 \\ w^2, & w > -1 \end{cases}$$

- Sometimes the Newton steps "skip" over a concave kink.

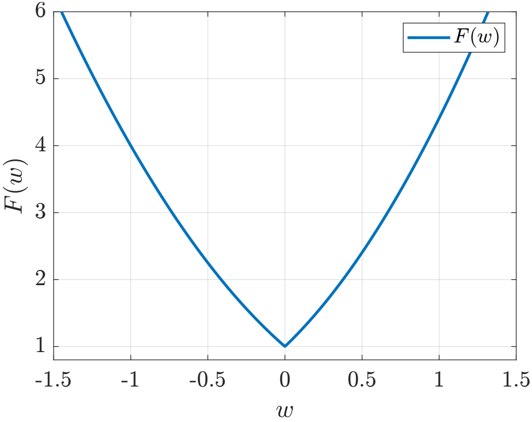


Nonsmooth optimization examples: convex kink at solution

$$\min_{w} F(w)$$

where

$$F(w) = \begin{cases} (-w+1)^2 & w < 0 \\ (1.1w+1)^2, & w > 0 \end{cases}$$



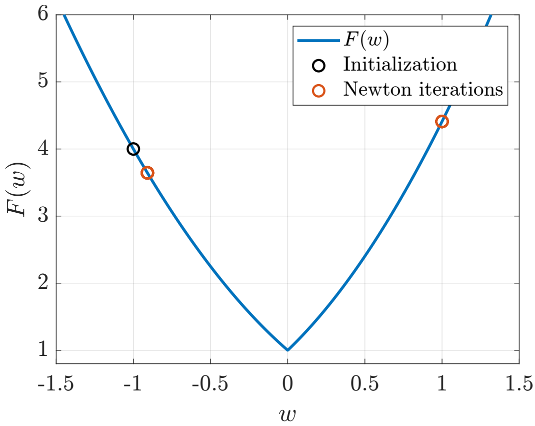
Nonsmooth optimization examples: convex kink at solution

$$\min_{w} F(w)$$

where

$$F(w) = \begin{cases} (-w+1)^2 & w < 0 \\ (1.1w+1)^2, & w > 0 \end{cases}$$

- It may be difficult to converge to a convex kink or to verify it



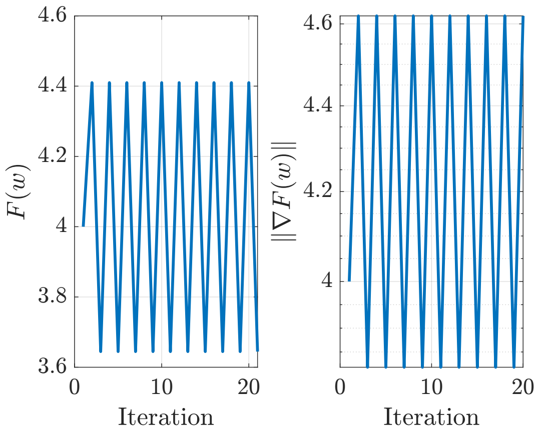
Nonsmooth optimization examples: convex kink at solution

$$\min_{w} F(w)$$

where

$$F(w) = \begin{cases} (-w+1)^2 & w < 0 \\ (1.1w+1)^2, & w > 0 \end{cases}$$

- It may be difficult to converge to a convex kink or to verify it
- At convex kink, we may need to compute a subgradient, and check if $0 \in \partial F(w)$
- More complicated to get "KKT conditions" for nonsmooth problems, may not work at other kinks



line search may help, but still need to verify $0 \in \partial F(w)$

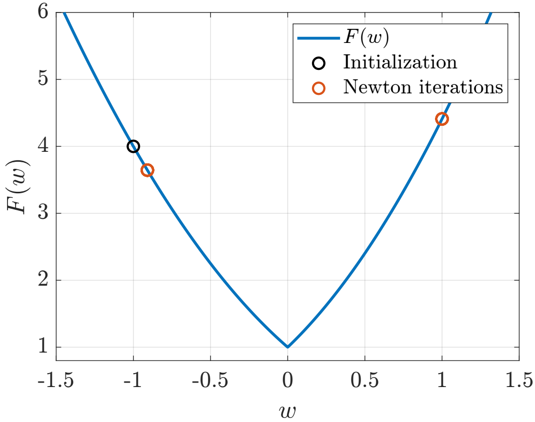
Nonsmooth optimization examples: convex kink at solution

$$\min_{w} F(w)$$

where

$$F(w) = \begin{cases} (-w+1)^2 & w < 0 \\ (1.1w+1)^2, & w > 0 \end{cases}$$

- It may be difficult to converge to a convex kink or to verify it
- At convex kink, we may need to compute a subgradient, and check if $0 \in \partial F(w)$
- More complicated to get "KKT conditions" for nonsmooth problems, may not work at other kinks
- Even more difficult: generic solver that solves these "KKT conditions"



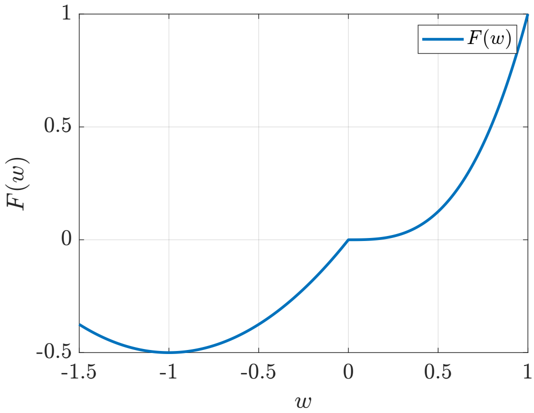
line search may help, but still need to verify $0 \in \partial F(w)$

Stopping despite having a descent direction

$$\min_{w} F(w)$$

where

$$F(w) = \begin{cases} w + 0.5w^2 & w < 0 \\ w^3, & w > 0 \end{cases}$$



Nonsmooth optimization examples: concave kink lead to stalling

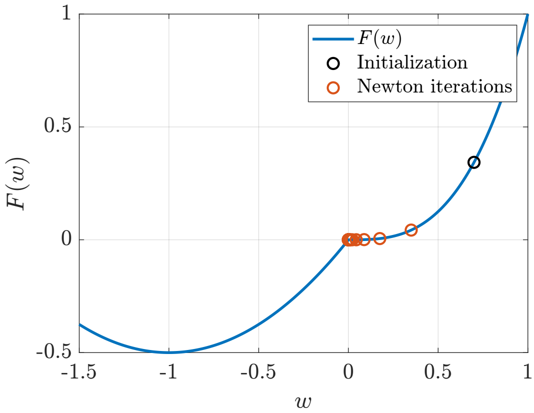
Stopping despite having a descent direction

$$\min_{w} F(w)$$

where

$$F(w) = \begin{cases} w + 0.5w^2 & w < 0 \\ w^3, & w > 0 \end{cases}$$

- Stops where right derivative is $\nabla F(w) = 0$
- Not a "saddle point", left derivative is $\nabla F(w) < 0$



Nonsmooth optimization examples: concave kink lead to stalling

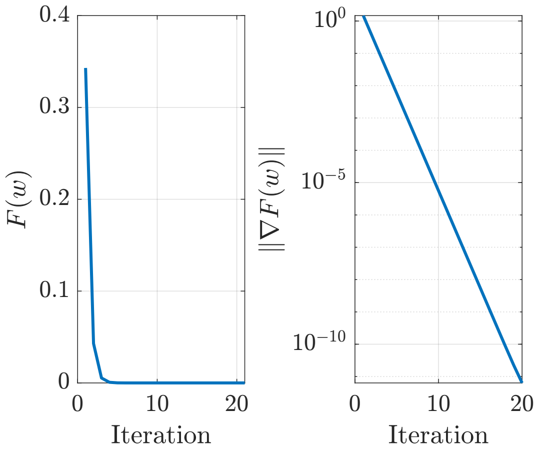
Stopping despite having a descent direction

$$\min_{w} F(w)$$

where

$$F(w) = \begin{cases} w + 0.5w^2 & w < 0 \\ w^3, & w > 0 \end{cases}$$

- Stops where right derivative is $\nabla F(w) = 0$
- Not a "saddle point", left derivative is $\nabla F(w) < 0$
- **Conclusion**: we use the wrong optimality conditions and step computation



Nonsmooth optimization examples: concave kink lead to stalling

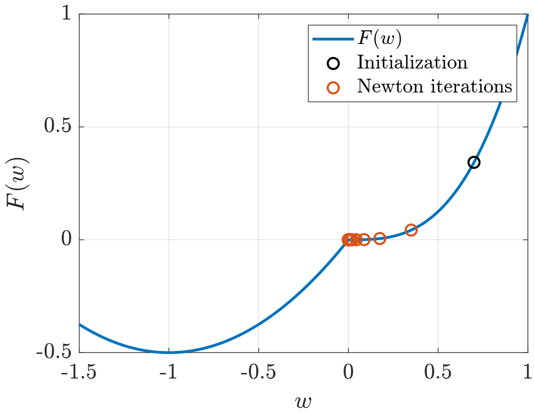
Stopping despite having a descent direction

$$\min_{w} F(w)$$

where

$$F(w) = \begin{cases} w + 0.5w^2 & w < 0 \\ w^3, & w > 0 \end{cases}$$

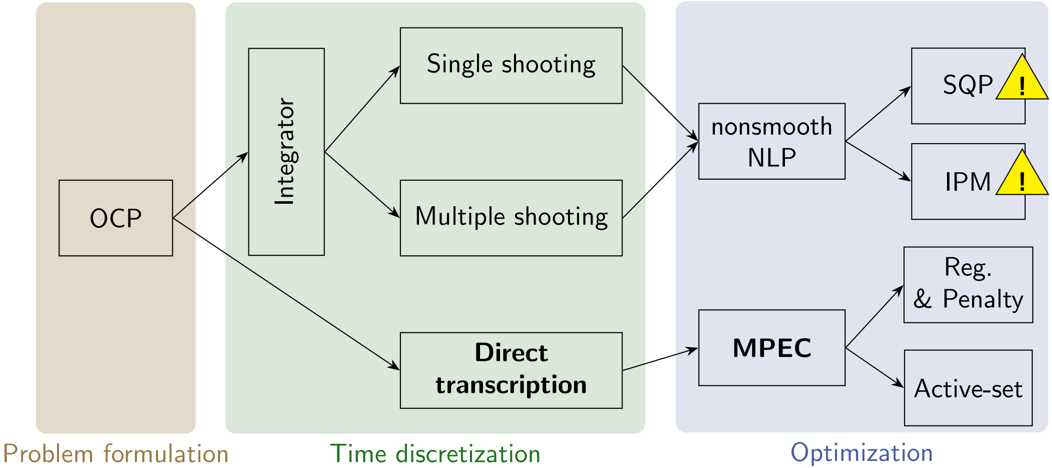
- Stops where right derivative is $\nabla F(w) = 0$
- Not a "saddle point", left derivative is $\nabla F(w) < 0$
- **Conclusion**: we use the wrong optimality conditions and step computation
- We resolve these problems in the Lectures 5 and 6



Outline

Work flow in nonsmooth direct optimal control

First discretize, then optimize.



OCP = Optimal Control Problem

NLP = Nonlinear Program

MPEC = Mathematical Program with Equilibrium Constraints

SQP = Sequential Quadratic Programming

IPM = Interior-Point Method

Reg. = Regularization

Main ideas of FESD

Based on [Baumrucker and Biegler, 2009, Nurkanović et al., 2024, Nurkanović and Diehl, 2022]

FESD overview

1. Transform nonsmooth system into dynamic complementarity system (Lecture 3)

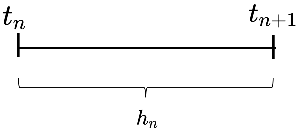
$$\dot{x} \in F_{\mathrm{F}}(x,u) \quad \iff \quad \begin{aligned} \dot{x} &= F(x,u)\,\theta \\ 0 &= g(x) - \lambda - e\mu \\ 0 &\leq \theta \perp \lambda \geq 0 \\ 1 &= e^\top \theta \end{aligned}$$

Main ideas of FESD

Based on [Baumrucker and Biegler, 2009, Nurkanović et al., 2024, Nurkanović and Diehl, 2022]

FESD overview

1. Transform nonsmooth system into dynamic complementarity system (Lecture 3)
2. Consider at least two integration intervals = finite elements

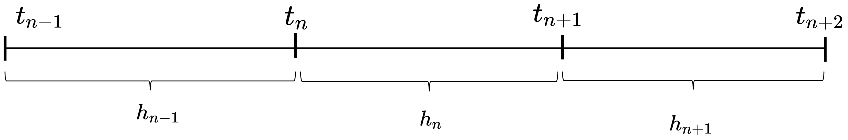


Main ideas of FESD

Based on [Baumrucker and Biegler, 2009, Nurkanović et al., 2024, Nurkanović and Diehl, 2022]

FESD overview

1. Transform nonsmooth system into dynamic complementarity system (Lecture 3)
2. Consider at least two integration intervals = finite elements

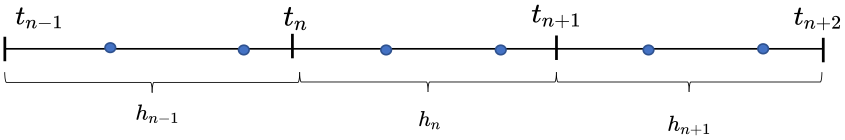


Main ideas of FESD

Based on [Baumrucker and Biegler, 2009, Nurkanović et al., 2024, Nurkanović and Diehl, 2022]

FESD overview

1. Transform nonsmooth system into dynamic complementarity system (Lecture 3)
2. Consider at least two integration intervals = finite elements
3. Use general implicit Runge-Kutta methods (Lectures 2)

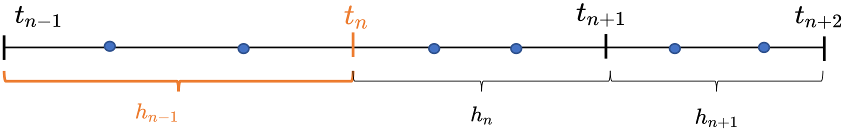


Main ideas of FESD

Based on [Baumrucker and Biegler, 2009, Nurkanović et al., 2024, Nurkanović and Diehl, 2022]

FESD overview

1. Transform nonsmooth system into dynamic complementarity system (Lecture 3)
2. Consider at least two integration intervals = finite elements
3. Use general implicit Runge-Kutta methods (Lectures 2)
4. Let step sizes h_n be degrees of freedom (under-determined system)

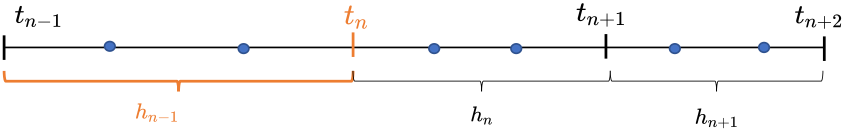


Main ideas of FESD

Based on [Baumrucker and Biegler, 2009, Nurkanović et al., 2024, Nurkanović and Diehl, 2022]

FESD overview

1. Transform nonsmooth system into dynamic complementarity system (Lecture 3)
2. Consider at least two integration intervals = finite elements
3. Use general implicit Runge-Kutta methods (Lectures 2)
4. Let step sizes h_n be degrees of freedom
5. Cross complementarity conditions - adapt h_n for switch detection

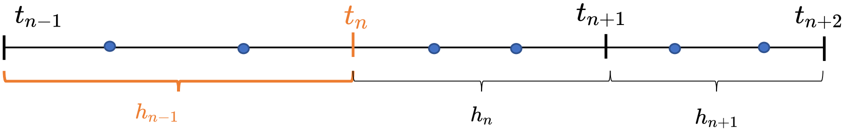


Main ideas of FESD

Based on [Baumrucker and Biegler, 2009, Nurkanović et al., 2024, Nurkanović and Diehl, 2022]

FESD overview

1. Transform nonsmooth system into dynamic complementarity system (Lecture 3)
2. Consider at least two integration intervals = finite elements
3. Use general implicit Runge-Kutta methods (Lectures 2)
4. Let step sizes h_n be degrees of freedom
5. Cross complementarity conditions - adapt h_n for switch detection
6. Step equilibration - remove degrees of freedom if no switch



From Filippov to dynamic complementarity systems

Using the KKT conditions of the parametric LP

LP representation

$$\dot{x} = F(x,u)\ \theta$$

$$\text{with} \quad \theta \in \underset{\tilde{\theta} \in \mathbb{R}^{n_f}}{\operatorname{argmin}} \quad g(x)^\top \tilde{\theta}$$

$$\text{s.t.} \quad 0 \leq \tilde{\theta}$$

$$1 = e^\top \tilde{\theta}$$

where

$$F(x,u) := [f_1(x,u), \dots, f_{n_f}(x,u)] \in \mathbb{R}^{n_x \times n_f}$$

$$g(x) := [g_1(x), \dots, g_{n_f}(x)]^\top \in \mathbb{R}^{n_f}$$

$$e := [1, 1, \dots, 1]^\top \in \mathbb{R}^{n_f}$$

From Filippov to dynamic complementarity systems

Using the KKT conditions of the parametric LP

LP representation

$$\dot{x} = F(x,u)\,\theta$$

$$\text{with} \quad \theta \in \underset{\tilde{\theta}\in\mathbb{R}^{n_f}}{\operatorname{argmin}} \quad g(x)^\top \tilde{\theta}$$
$$\text{s.t.} \quad 0 \le \tilde{\theta}$$
$$1 = e^\top \tilde{\theta}$$

where

$$F(x,u) := [f_1(x,u), \dots, f_{n_f}(x,u)] \in \mathbb{R}^{n_x \times n_f}$$
$$g(x) := [g_1(x), \dots, g_{n_f}(x)]^\top \in \mathbb{R}^{n_f}$$
$$e := [1,1,\dots,1]^\top \in \mathbb{R}^{n_f}$$

Express equivalently by optimality conditions:

Dynamic Complementarity System (DCS)

$$\dot{x} = F(x,u)\,\theta \tag{2a}$$
$$0 = g(x) - \lambda - e\mu \tag{2b}$$
$$0 \le \theta \perp \lambda \ge 0 \tag{2c}$$
$$1 = e^\top \theta \tag{2d}$$

Compact notation

$$\dot{x} = F(x,u)\,\theta$$
$$0 = G_{\mathrm{LP}}(x,\theta,\lambda,\mu),$$

- $\mu \in \mathbb{R}$ and $\lambda \in \mathbb{R}^{n_f}$ are Lagrange multipliers
- (1c) $\Leftrightarrow \min\{\theta,\lambda\} = 0 \in \mathbb{R}^{n_f}$
- Together, (1b), (1c), (1d) determine the $(2n_f+1)$ variables (θ,λ,μ) uniquely

Conventional discretization by Implicit Runge Kutta (IRK) method

Continuous time DCS

$$
\begin{aligned}
x(0) &= \bar{x}_0, \\
\dot{x}(t) &= v(t) \\
v(t) &= F(x(t), u(t))\, \theta(t) \\
0 &= g(x(t)) - \lambda(t) - e\mu(t) \\
0 &\leq \theta(t) \perp \lambda(t) \geq 0 \\
1 &= e^\top \theta(t), \quad t \in [0, T]
\end{aligned}
$$

Conventional discretization by Implicit Runge Kutta (IRK) method

Continuous time DCS

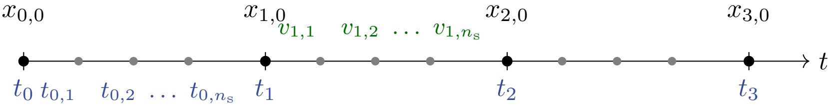
$$
\begin{aligned}
x(0) &= \bar{x}_0, \\
\dot{x}(t) &= v(t) \\
v(t) &= F(x(t), u(t))\,\theta(t) \\
0 &= g(x(t)) - \lambda(t) - e\mu(t) \\
0 &\leq \theta(t) \perp \lambda(t) \geq 0 \\
1 &= e^\top \theta(t), \quad t \in [0, T]
\end{aligned}
$$

Discrete time IRK-DCS equation

$$
\begin{aligned}
x_{0,0} &= \bar{x}_0, \quad x_{n+1,0} = x_{n,0} + h \textstyle\sum_{i=1}^{n_\mathrm{s}} b_i v_{n,i} \\
x_{n,i} &= x_{n,0} + h \textstyle\sum_{j=1}^{n_\mathrm{s}} a_{i,j} v_{n,j} \\
v_{n,i} &= F(x_{n,i}, u_{n,i})\,\theta_{n,i} \\
0 &= g(x_{n,i}) - \lambda_{n,i} - e\mu_{n,i} \\
0 &\leq \theta_{n,i} \perp \lambda_{n,i} \geq 0 \\
1 &= e^\top \theta_{n,i}, \quad i = 1, \ldots, n_\mathrm{s}, \quad n = 0, \ldots, N-1
\end{aligned}
$$

Notation: $x_{n,i} \in \mathbb{R}^{n_x}, \theta_{n,i} \in \mathbb{R}^m$ etc. RK stage values with:

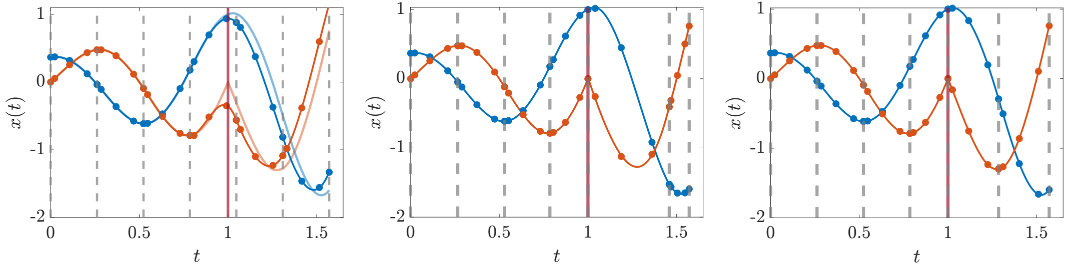
- $n \in \{0, 1, \ldots, N\}$ - index of integration step; step length $h := T/N$
- $i, j \in \{0, 1, \ldots, n_\mathrm{s}\}$ - index of intermediate IRK stage / collocation point
- $a_{i,j}$ and b_i - Butcher tableau entries of Implicit Runge Kutta method



Finite Elements with Switch Detection (FESD)

FESD is a novel DCS discretization method based on three ideas:

- make step sizes h_n free, ensure $\sum_{n=0}^{N-1} h_n = T$ (cf. [Baumrucker and Biegler, 2009])
- allow switches only at element boundaries, enforce via **cross-complementarities**,
- remove spurious degrees of freedom via **step equilibration**.



conventional discretization | variable step sizes and cross-complementarities | FESD discretization with step equilibration

Conventional DCS and FESD discretization without step equilibration

Time-stepping discretization

$$
\begin{aligned}
x_{0,0} &= \bar{x}_0, \quad h = T/N \\
x_{n+1,0} &= x_{n,0} + h \sum_{i=1}^{n_s} b_i v_{n,i} \\
x_{n,i} &= x_{n,0} + h \sum_{j=1}^{n_s} a_{i,j} v_{n,j} \\
v_{n,i} &= F(x_{n,i}, u_{n,i})\, \theta_{n,i} \\
0 &= g(x_{n,i}) - \lambda_{n,i} - e\mu_{n,i} \\
0 &\leq \theta_{n,i} \perp \lambda_{n,i} \geq 0 \\
1 &= e^\top \theta_{n,i}
\end{aligned}
$$

for $i = 1, \dots, n_s$
and $n = 0, \dots, N-1$

FESD discretization without step equilibration

$$
\begin{aligned}
x_{0,0} &= \bar{x}_0, \ \sum_{n=0}^{N-1} h_n = T \\
x_{n+1,0} &= x_{n,0} + h_n \sum_{i=1}^{n_s} b_i v_{n,i} \\
x_{n,i} &= x_{n,0} + h_n \sum_{j=1}^{n_s} a_{i,j} v_{n,j} \\
v_{n,i} &= F(x_{n,i}, u_{n,i})\, \theta_{n,i} \\
0 &= g(x_{n,i'}) - \lambda_{n,i'} - e\mu_{n,i'} \\
0 &\leq \theta_{n,i} \perp \lambda_{n,i'} \geq 0 \quad \text{(cross-complementarities)} \\
1 &= e^\top \theta_{n,i}
\end{aligned}
$$

for $i = 1, \dots, n_s$ and $n = 0, \dots, N-1$
and $i' = 0, 1, \dots, n_s$

- N extra variables $(h_0, \dots, h_{N-1})$ restricted by one extra equality
- Additional multipliers $\lambda_{n,0}, \mu_{n,0}$ are uniquely determined

Conventional DCS and FESD discretization with step equilibration

Time-stepping discretization

$$
\begin{aligned}
x_{0,0} &= \bar{x}_0, \quad h = T/N \\
x_{n+1,0} &= x_{n,0} + h \textstyle\sum_{i=1}^{n_{\mathrm{s}}} b_i v_{n,i} \\
x_{n,i} &= x_{n,0} + h \textstyle\sum_{j=1}^{n_{\mathrm{s}}} a_{i,j} v_{n,j} \\
v_{n,i} &= F(x_{n,i}, u_{n,i})\, \theta_{n,i} \\
0 &= g(x_{n,i}) - \lambda_{n,i} - e\mu_{n,i} \\
0 &\leq \theta_{n,i} \perp \lambda_{n,i} \geq 0 \\
1 &= e^\top \theta_{n,i}
\end{aligned}
$$

for $i = 1, \dots, n_{\mathrm{s}}$
and $n = 0, \dots, N-1$

FESD discretization with step equilibration

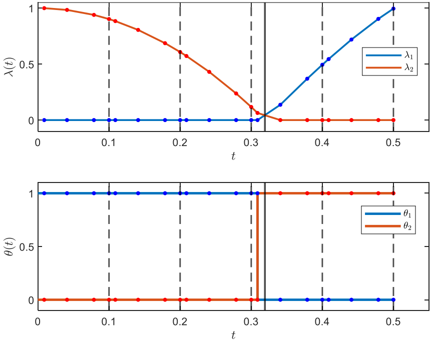
$$
\begin{aligned}
x_{0,0} &= \bar{x}_0, \ \textstyle\sum_{n=0}^{N-1} h_n = T \\
x_{n+1,0} &= x_{n,0} + h_n \textstyle\sum_{i=1}^{n_{\mathrm{s}}} b_i v_{n,i} \\
x_{n,i} &= x_{n,0} + h_n \textstyle\sum_{j=1}^{n_{\mathrm{s}}} a_{i,j} v_{n,j} \\
v_{n,i} &= F(x_{n,i}, u_{n,i})\, \theta_{n,i} \\
0 &= g(x_{n,i'}) - \lambda_{n,ii'} - e\mu_{n,i'} \\
0 &\leq \theta_{n,i} \perp \lambda_{n,i'} \geq 0 \quad \text{(cross-complementarities)} \\
1 &= e^\top \theta_{n,i} \\
0 &= \nu(\theta_{n'}, \theta_{n'+1}, \lambda_{n'}, \lambda_{n'+1}) \cdot (h_{n'} - h_{n'+1})
\end{aligned}
$$

for $i = 1, \dots, n_{\mathrm{s}}$ and $n = 0, \dots, N-1$
and $i' = 0, 1, \dots, n_{\mathrm{s}}$ and $n' = 0, \dots, N-2$

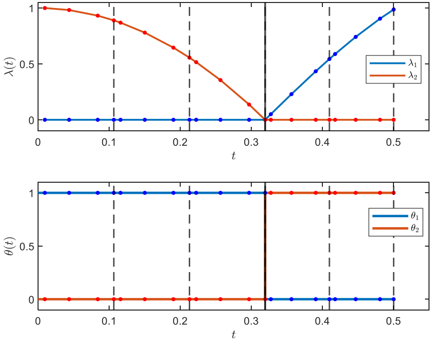
- N extra variables $(h_0, \dots, h_{N-1})$ restricted by one extra equality
- Additional multipliers $\lambda_{n,0}, \mu_{n,0}$ are uniquely determined
- Indicator function $\nu(\theta_{n'}, \theta_{n'+1}, \lambda_{k'}, \lambda_{k'+1})$ only zero if a switch occurs

Multipliers in conventional and FESD discretization

Time stepping discretization:



FESD discretization:



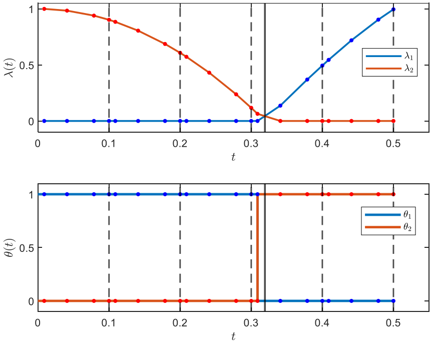
Lemma (Cross complementarity)

If any $\theta_{n,j,i}$ with $j = 1, \ldots, n_{\mathrm{s}}$ is positive, then all $\lambda_{n,j',i}$ with $j' = 0, \ldots, n_{\mathrm{s}}$ must be zero. Conversely, if any $\lambda_{n,j',i}$ is positive, then all $\theta_{n,j,i}$ are zero.

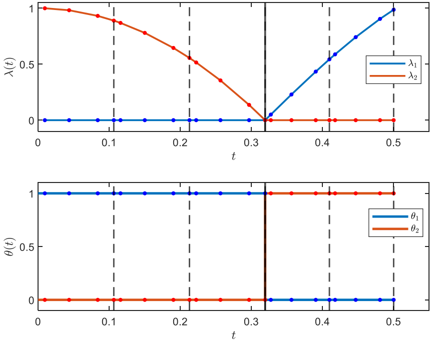
Notation $\lambda_{n,j,i}$ - n - finite element, j - RK stage, i - component of vector

Multipliers in conventional and FESD discretization

Time stepping discretization:



FESD discretization:



FESD's cross-complementarities exploit the fact that the multiplier $\lambda_i(t)$ is continuous in time.
On boundary, $\lambda_i(t_n)$ must be zero if $\theta_i(t) > 0$ for any $t \in [t_{n-1}, t_{n+1}]$ on the adjacent intervals.
This implicitly imposes the constraint $g_i(x_n) - g_j(x_n) = 0$.
$\implies$ h_n adapts for exact switch detection

Step equilibration

- if no switches happen, cross complementarity implied by standard complementarity
- spurious degrees of freedom in h_n: more degrees of freedom than equations

Step equilibration

- if no switches happen, cross complementarity implied by standard complementarity
- spurious degrees of freedom in h_n: more degrees of freedom than equations
- exploit complementarity of θ_n, λ_n to encode switching logic
- define (very complicated) switch indicator function ν (cf. [Nurkanović, 2023]:

$$\nu(\theta_n, \theta_{n+1}, \lambda_n, \lambda_{n+1}) := \begin{cases} \text{positive}, & \text{if no switch at } t_{n+1} \\ 0, & \text{if switch at } t_{n+1} \end{cases}$$

Step equilibration

- if no switches happen, cross complementarity implied by standard complementarity
- spurious degrees of freedom in h_n: more degrees of freedom than equations
- exploit complementarity of θ_n, λ_n to encode switching logic
- define (very complicated) switch indicator function ν (cf. [Nurkanović, 2023]:

$$\nu(\theta_n, \theta_{n+1}, \lambda_n, \lambda_{n+1}) := \begin{cases} \text{positive}, & \text{if no switch at } t_{n+1} \\ 0, & \text{if switch at } t_{n+1} \end{cases}$$

- step equilibration:

$$0 = \nu(\theta_n, \theta_{n+1}, \lambda_n, \lambda_{n+1}) \cdot (h_n - h_{n+1}), \quad n = 0, \dots, N-2$$

Step equilibration

- if no switches happen, cross complementarity implied by standard complementarity
- spurious degrees of freedom in h_n: more degrees of freedom than equations
- exploit complementarity of θ_n, λ_n to encode switching logic
- define (very complicated) switch indicator function ν (cf. [Nurkanović, 2023]:

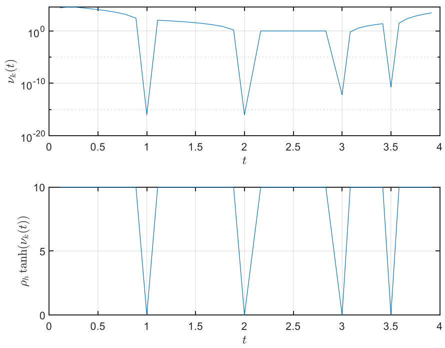
$$\nu(\theta_n, \theta_{n+1}, \lambda_n, \lambda_{n+1}) := \begin{cases} \text{positive}, & \text{if no switch at } t_{n+1} \\ 0, & \text{if switch at } t_{n+1} \end{cases}$$

- step equilibration:

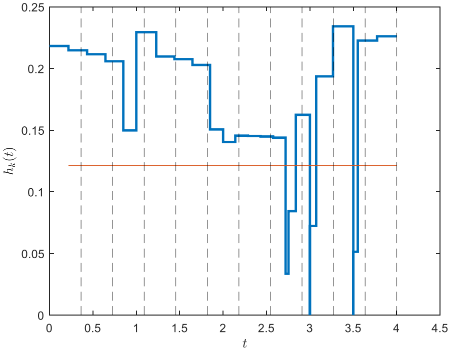
$$0 = \nu(\theta_n, \theta_{n+1}, \lambda_n, \lambda_{n+1}) \cdot (h_n - h_{n+1}), \qquad n = 0, \dots, N-2$$

- Summary:
  - If switch happens, then h_n is determined by cross complementarity.
  - If no switch happens, then h_n is determined by step equilibration.

Indicator function over time:



Step size over time:

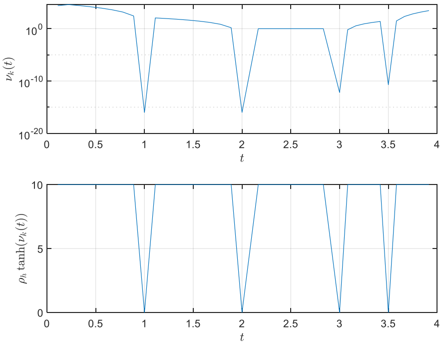


Optimizer varies step size randomly, potentially playing with integration errors.

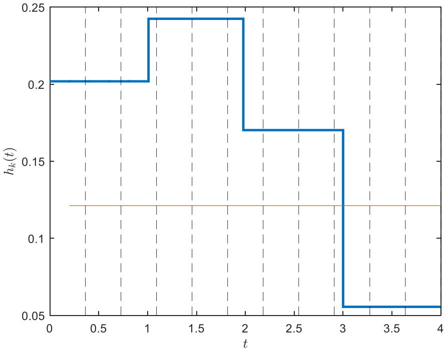
Numerical solution with equilibration

Example with four switches

Indicator function over time:



Step size over time:



Equidistant grid on each "switching stage". Jumps exactly at switching times.

Integration order plots for FESD and IRK time stepping

Revisit example from Lecture 3

Tutorial example

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

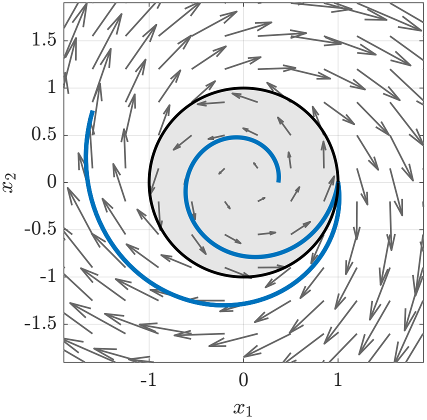
with $A_1 = \begin{bmatrix} 1 & 2\pi \\ -2\pi & 1 \end{bmatrix}$, $A_2 = \begin{bmatrix} 1 & -2\pi \\ 2\pi & 1 \end{bmatrix}$

$x(0) = (e^{-1}, 0)$ for $t \in [0, \frac{\pi}{2}]$.

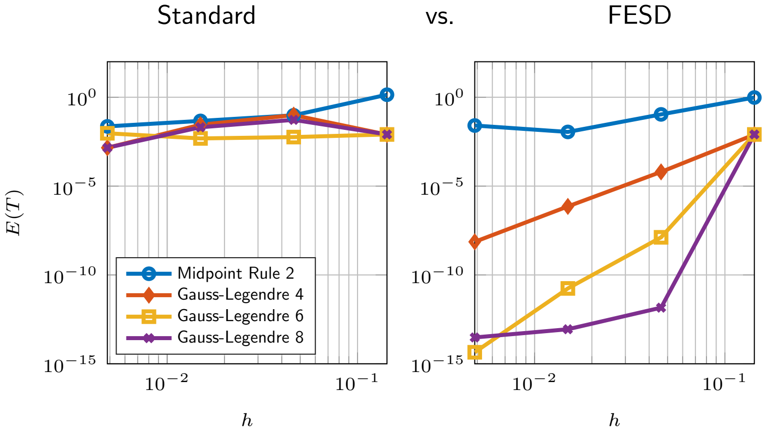
Compute global integration error $E(T)$ using different strategies.

Compute solution approximation:

1. With fixed step size IRK methods (time-stepping).
2. FESD with same underlying IRK methods.



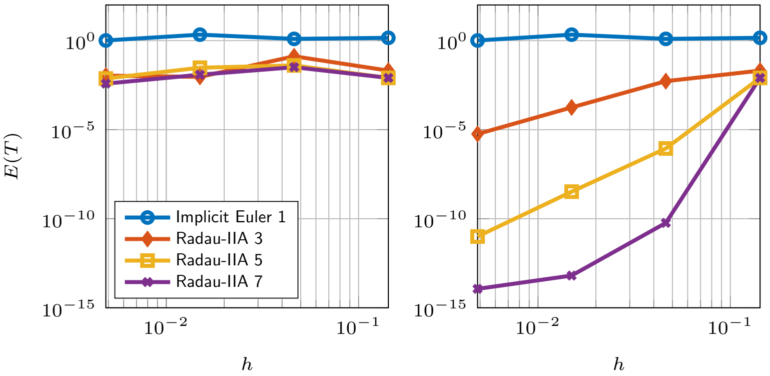
FESD recovers high integration order for switched systems



Integration error $E(T)$ at time $T = \pi/2$ vs. step-size h, for different IRK methods.
FESD discretization recovers high integration order

FESD recovers high integration order for switched systems

Standard vs. FESD



Integration error $E(T)$ at time $T = \pi/2$ vs. step-size h, for different IRK methods.
FESD discretization recovers high integration order

Outline

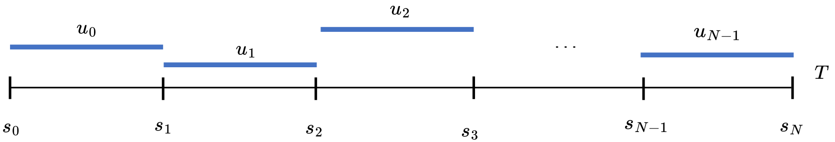
Discretizing optimal control problems with FESD

Discretized optimal control problem

$$\min_{s,z,u} \sum_{k=0}^{N-1} \Phi_L(s_k, z_k, u_k) + E(s_N)$$

$$\begin{aligned} \text{s.t.} \quad s_0 &= \bar{x}_0 \\ s_{k+1} &= \Phi_f(s_k, z_k, u_k) \\ 0 &= \Phi_{\text{int}}(s_k, z_k, u_k) \\ 0 &\geq h(s_k, u_k), \; k = 0, \dots, N-1 \\ 0 &\geq r(s_N) \end{aligned}$$

Control horizon $[0, T]$ with N control stages



- States at control grid points $s = (s_0, \dots, s_N)$
- Piecewise controls $u = (u_0, \dots, u_{N-1})$
- FESD with N_{FE} finite elements applied on every control interval

Discretizing optimal control problems with FESD

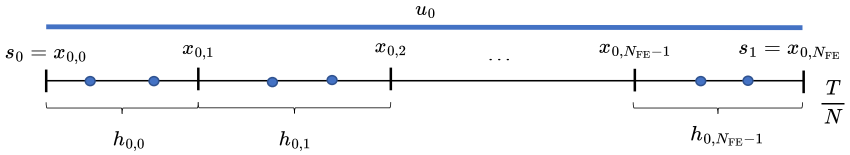
Discretized optimal control problem

$$\min_{s,z,u} \sum_{k=0}^{N-1} \Phi_L(s_k, z_k, u_k) + E(s_N)$$

$$\begin{aligned}
\text{s.t.} \quad s_0 &= \bar{x}_0 \\
s_{k+1} &= \Phi_f(s_k, z_k, u_k) \\
0 &= \Phi_{\text{int}}(s_k, z_k, u_k) \\
0 &\geq h(s_k, u_k), \; k = 0, \dots, N-1 \\
0 &\geq r(s_N)
\end{aligned}$$

- States at control grid points $s = (s_0, \dots, s_N)$
- Piecewise controls $u = (u_0, \dots, u_{N-1})$
- FESD with N_{FE} finite elements applied on every control interval
- Φ_{int} summarizes all internal FESD equations: RK, cross complementarity, step equilibration,...

Control horizon $[0, T]$ with N control stages



Discretizing optimal control problems with FESD

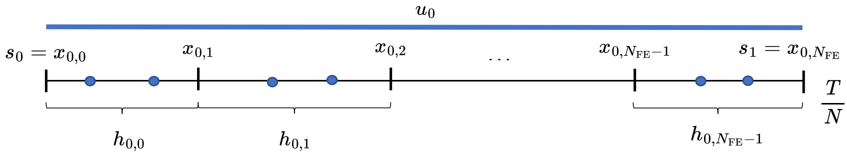
Discretized optimal control problem

$$\min_{s,z,u} \sum_{k=0}^{N-1} \Phi_L(s_k, z_k, u_k) + E(s_N)$$

$$\begin{aligned}
\text{s.t.} \quad s_0 &= \bar{x}_0 \\
s_{k+1} &= \Phi_f(s_k, z_k, u_k) \\
0 &= \Phi_{\text{int}}(s_k, z_k, u_k) \\
0 &\geq h(s_k, u_k), \; k = 0, \dots, N-1 \\
0 &\geq r(s_N)
\end{aligned}$$

- States at control grid points $s = (s_0, \dots, s_N)$
- Piecewise controls $u = (u_0, \dots, u_{N-1})$
- FESD with N_{FE} finite elements applied on every control interval
- Φ_{int} summarizes all internal FESD equations: RK, cross complementarity, step equilibration,...
- $z = (z_0, \dots, z_{N-1})$ - all interval variables: internal states, stage values of states and multipliers, step sizes, ...

Control horizon $[0, T]$ with N control stages



FESD-discretized optimal control problems are MPCC

Discretized optimal control problem

$$\min_{s,z,u} \sum_{k=0}^{N-1} \Phi_L(s_k, z_k, u_k) + E(s_N)$$
$$\text{s.t.} \quad s_0 = \bar{x}_0$$
$$s_{k+1} = \Phi_f(s_k, z_k, u_k)$$
$$0 = \Phi_{\text{int}}(s_k, z_k, u_k)$$
$$0 \geq h(s_k, u_k), \; k = 0, \dots, N-1$$
$$0 \geq r(s_N)$$

Collect $w = (s, z, u) \in \mathbb{R}^{n_w}$
Mathematical programs with complementarity constraints (MPCC) are more difficult than standard NLPs

NLP with Complementarity Constraints

$$\min_{w \in \mathbb{R}^{n_w}} F(w)$$
$$\text{s.t.} \quad 0 = G(w)$$
$$0 \geq H(w)$$
$$0 \leq G_1(w) \perp G_2(w) \geq 0$$

Standard and cross complementarity constraints summarized in

$$0 \leq G_1(w) \perp G_2(w) \geq 0$$

Revisiting the OCP example - now with FESD

Tutorial example inspired by [Stewart & Anitescu, 2010]

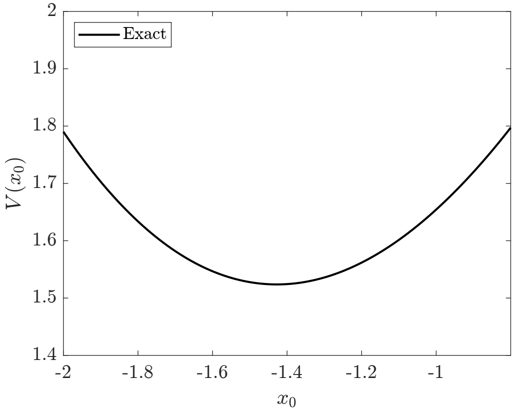
Continuous-time OCP

$$\min_{x(\cdot)\in\mathcal{C}^0([0,2])} \int_0^2 x(t)^2 \mathrm{d}t + (x(2) - 5/3)^2$$
$$\text{s.t.} \quad \dot{x}(t) = 2 - \mathrm{sign}(x(t)), \quad t \in [0,2]$$

Free initial value $x(0)$ is the effective degree of freedom.

Equivalent reduced problem

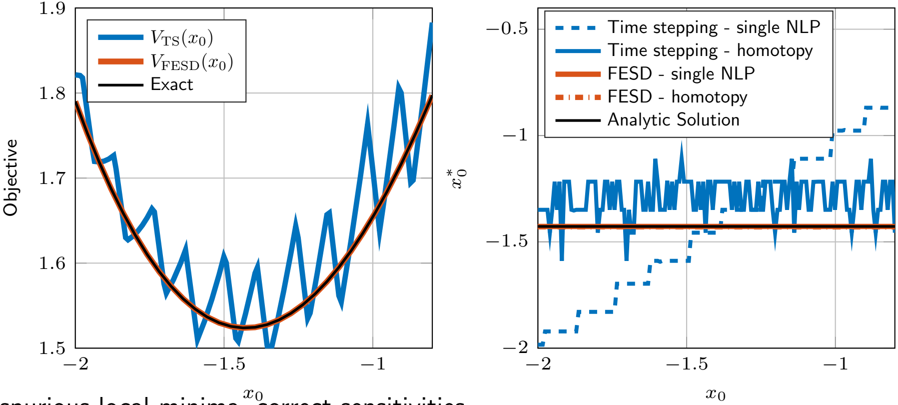
$$\min_{x_0\in\mathbb{R}} V(x_0)$$



- Denote by $V(x_0)$ the nonsmooth objective value for the unique feasible trajectory starting at $x(0) = x_0$.

Revisiting the OCP example - now with FESD

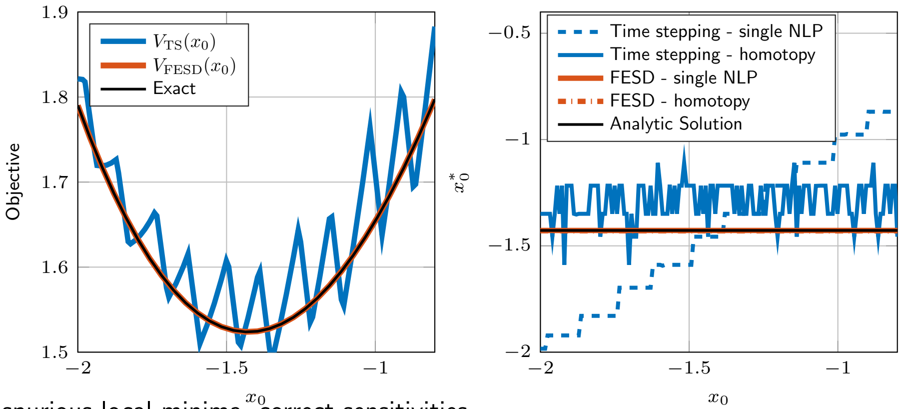
Tutorial example inspired by [Stewart & Anitescu, 2010]



- no spurious local minima, correct sensitivities
- convergence to the "true" local minimum, both with homotopy and without it
- accuracy of order $O(h^p)$, in contrast to standard approach with only $O(h)$

Revisiting the OCP example - now with FESD

Tutorial example inspired by [Stewart & Anitescu, 2010]



- no spurious local minima, correct sensitivities
- convergence to the "true" local minimum, both with homotopy and without it
- accuracy of order $O(h^p)$, in contrast to standard approach with only $O(h)$
- FESD solves the accuracy and convergence issues

nosnoc

An open source tool for optimal control of nonsmooth systems

Overview

- All reformulations are automated, simply provide problem and model data.
- A wide variety of features: DAEs, nonlinear and quadratic costs, general (including complementarity) path constraints, and terminal constraints.
- C++ code generation for embedded MPC
- Developed and Maintained by **Anton Pozharskiy**, Jonathan Frey, and Armin Nurkanović.

github.com/nosnoc/nosnoc

github.com/nosnoc/nosnoc_py

Available Systems

nosnoc supports various systems such as:

- Piecewise Smooth Systems (via Stewart or Heaviside Step reformulations).
- Heaviside Step Differential Inclusions [Nurkanović et al., 2024].
- Complementarity Lagrangian systems (via FESD-J [Nurkanović et al., 2024] or time-freezing [Nurkanović et al., 2023]).
- Projected Dynamical Systems [Pozharskiy et al., 2024].

Conclusions and summary

- Finite Elements with Switch Detection (FESD) allow highly accurate simulation and optimal control for nonsmooth systems of level NSD2
- Following similar lines, FESD can be derived for the Heaviside step reformulation

Conclusions and summary

- Finite Elements with Switch Detection (FESD) allow highly accurate simulation and optimal control for nonsmooth systems of level NSD2
- Following similar lines, FESD can be derived for the Heaviside step reformulation
- Key ideas: make step sizes degrees of freedom and introduce implicit relations that locate the switches
- Switch detection not only essential for high accuracy, **but also for correct sensitivities** (no spurious solutions)

Conclusions and summary

- Finite Elements with Switch Detection (FESD) allow highly accurate simulation and optimal control for nonsmooth systems of level NSD2
- Following similar lines, FESD can be derived for the Heaviside step reformulation
- Key ideas: make step sizes degrees of freedom and introduce implicit relations that locate the switches
- Switch detection not only essential for high accuracy, **but also for correct sensitivities** (no spurious solutions)
- FESD solves many of the issues that standard methods have: integration accuracy, convergence of sensitivities
- Main practical difficulty: solving Mathematical Programs with Complementarity Constraints (MPCC)

Cited references I

Aizerman, M. and Gantmakher, F. (1958).
On the stability of periodic motions.
Journal of Applied Mathematics and Mechanics, 22(6):1065–1078.

Baumrucker, B. T. and Biegler, L. T. (2009).
Mpec strategies for optimization of a class of hybrid dynamic systems.
Journal of Process Control, 19(8):1248–1256.

Bock, H. (1987).
Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen, volume 183 of *Bonner Mathematische Schriften*.
Universität Bonn, Bonn.

Bock, H. G. and Plitt, K. J. (1984).
A multiple shooting algorithm for direct solution of optimal control problems.
In *Proceedings of the IFAC World Congress*, pages 242–247. Pergamon Press.

Cited references II

Brandt-Pollmann, U. (2004).
Numerical solution of optimal control problems with implicitly defined discontinuities with applications in engineering.
PhD thesis, IWR, University of Heidelberg.

Dontchev, A. L. and Rockafellar, R. T. (2014).
Implicit Functions and Solution Mappings: A View from Variational Analysis.
Springer.

Kirches, C. (2006).
A Numerical Method for Nonlinear Robust Optimal Control with Implicit Discontinuities and an Application to Powertrain Oscillations.
Diploma thesis, University of Heidelberg.

Kong, N. J., Council, G., and Johnson, A. M. (2021).
iLQR for piecewise-smooth hybrid dynamical systems.
In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 5374–5381.

Cited references III

Kong, N. J., Payne, J. J., Zhu, J., and Johnson, A. M. (2024).
Saltation matrices: The essential tool for linearizing hybrid dynamical systems.
Proceedings of the IEEE.

Nurkanović, A. (2023).
Numerical Methods for Optimal Control of Nonsmooth Dynamical Systems.
PhD thesis, University of Freiburg.

Nurkanović, A., Albrecht, S., Brogliato, B., and Diehl, M. (2023).
The time-freezing reformulation for numerical optimal control of complementarity lagrangian systems with state jumps.
Automatica, 158:111295.

Nurkanović, A., Albrecht, S., and Diehl, M. (2020).
Limits of MPCC Formulations in Direct Optimal Control with Nonsmooth Differential Equations.
In *2020 European Control Conference (ECC)*, pages 2015–2020.

Cited references IV

Nurkanović, A. and Diehl, M. (2022).
NOSNOC: A software package for numerical optimal control of nonsmooth systems.
IEEE Control Systems Letters, 6:3110–3115.

Nurkanović, A., Frey, J., Pozharskiy, A., and Diehl, M. (2024).
Fesd-j: Finite elements with switch detection for numerical optimal control of rigid bodies with impacts and coulomb friction.
Nonlinear Analysis: Hybrid Systems, 52:101460.

Nurkanović, A., Pozharskiy, A., Frey, J., and Diehl, M. (2024).
Finite elements with switch detection for numerical optimal control of nonsmooth dynamical systems with set-valued Heaviside step functions.
Nonlinear Analysis: Hybrid Systems, 54:101518.

Nurkanović, A., Sperl, M., Albrecht, S., and Diehl, M. (2024).
Finite Elements with Switch Detection for Direct Optimal Control of Nonsmooth Systems.
Numerische Mathematik, pages 1–48.

Cited references V

Pozharskiy, A., Nurkanović, A., and Diehl, M. (2024).
Finite elements with switch detection for numerical optimal control of projected dynamical systems.
arXiv preprint arXiv:2404.05367.

Quirynen, R. (2017).
Numerical Simulation Methods for Embedded Optimization.
PhD thesis, KU Leuven and University of Freiburg.

Stewart, D. E. and Anitescu, M. (2010).
Optimal control of systems with discontinuous differential equations.
Numerische Mathematik, 114(4):653–695.

Derivation of the Saltation matrix

Before and after the switch the $S(t)$ obey linear variational differential equation (VDE)

$$\dot{S}_i(t) = \frac{\partial f_i(x)}{\partial x} S_i(t),\ \ i = 1,2$$

The function $S(t)$ obeys smooth VDEs, on both sides of t_{s}, but exhibits a jump at t_{s}.

Proposition

Regard the system (1) *with* $x(0) = x_0 \in R_i$ *on an interval* $[0,T]$ *with a switch at* $t_{\mathrm{s}} \in (0,T)$*. Assume that the functions* $f_1(x),\ f_2(x), \psi_{i,j}(x)$ *are continuously differentiable along* $x(t), t \in [0,T]$*. Assume the solution* $x(t)$ *reaches the surface of discontinuity transversally, i.e.,* $\nabla\psi(x(t_{\mathrm{s}}))^\top f_1(x(t_{\mathrm{s}})) > 0$*. Then the sensitivity* $S(T;0)$ *of a solution* $x(t;x_0)$ *of the system described by the ODE* (1) *is given by*

$$\begin{aligned} S(T;0) &= S(T;t_{\mathrm{s}}^+)J(x(t_{\mathrm{s}};x_0))S(t_{\mathrm{s}}^-;0) \ \text{with} \\ J(x(t_{\mathrm{s}};x_0)) &\coloneqq I + \frac{(f_2(x(t_{\mathrm{s}};x_0)) - f_1(x(t_{\mathrm{s}};x_0)))\nabla\psi(x(t_{\mathrm{s}};x_0))^\top}{\nabla\psi(x(t_{\mathrm{s}};x_0))^\top f_1(x(t_{\mathrm{s}};x_0))}. \end{aligned} \tag{3}$$

Proof follows similar lines as in [Stewart and Anitescu, 2010]

For $t < t_{\mathrm{s}}$, the solution $x(t; x_0)$ satisfies the ODE $\dot{x} = f_i(x(t; x_0))$ and the sensitivity matrix $S^x(t, 0; x_0) = \frac{\partial x(t;x_0)}{\partial x_0}$ obeys:

$$\dot{S^x}(t) = \frac{\partial f(x)}{\partial x} S^x(t), \;\; S^x(0) = I.$$

At $t = t_{\mathrm{s}}$ the solution reaches the surface of discontinuity:

$$\psi_{i,j}(x(t_{\mathrm{s}}(x_0; x_0))) = 0. \tag{4}$$

For $t > t_{\mathrm{s}}$, one has $y(t) = f_*(y(t; y_0))$ which is related to the solution via

$$y(t; y_0) = x(t + t_{\mathrm{s}}(x_0); x_0), \;\; y_0(x_0) = x(t_{\mathrm{s}}(x_0); x_0). \tag{5}$$

Proof of the proposition (2/3)

Note that $y(t - t_s(x_0); x_0) = x(t; x_0)$. Therefore, the sensitivity, for $t > t_s$ can be computed via

$$\begin{aligned} S^x(t, 0; x_0) = \frac{\partial x(t, x_0)}{\partial x_0} &= \frac{\partial y(t - t_s(x_0)); y_0(x_0))}{\partial x_0} \\ &= \frac{\partial y(t - t_s}{\partial t} \frac{\partial t_s(x_0)}{\partial x_0}^\top + S^y(t - t_s; y_0) \frac{\partial y_0(x_0)}{\partial x_0}, \\ &= -f_*(x(t)) \frac{\partial t_s(x_0)}{\partial x_0}^\top + S^y(t - t_s; y_0) \frac{\partial y_0(x_0)}{\partial x_0}, \end{aligned} \tag{6}$$

We can compute $\frac{\partial y_0(x_0)}{\partial x_0}$ at $t = t_s^-$ using (5)

$$\frac{\partial y_0(x_0)}{\partial x_0} = \frac{\partial x(t_s(x_0); x_0)}{\partial x_0} = f_i(x) \frac{\partial t_s(x_0)}{\partial x_0}^\top + S^x(t_s^-, 0; x_0). \tag{7}$$

Using the implicit function theorem (cf. [Dontchev and Rockafellar, 2014, Theorem 1B.1]) for (4), again at $t = t_s^-$ we obtain

$$\frac{\partial t_s(x_0)}{\partial x_0}^\top = -\frac{\nabla \psi_{i,j}(x(t_s(x_0; x_0)))^\top S^x(t_s^-, 0; x_0)}{\nabla \psi_{i,j}(x(t_s(x_0; x_0)))^\top f_i(x)}. \tag{8}$$

Proof of the proposition (3/3)

Let $t \to t_{\mathrm{s}}^{+}$ in (6), then $S^{y}(t - t_{\mathrm{s}}; y_0) \to I$. Plugging (7) and (8) into (6) for the remaining unknown terms we obtain

$$\begin{aligned} S^{x}(t_{\mathrm{s}}^{+}, 0; x_0) &= f_{*}(x(t)) \frac{\nabla \psi_{i,j}(x(t_{\mathrm{s}}(x_0; x_0)))^{\top} S^{x}(t_{\mathrm{s}}^{-}, 0; x_0)}{\nabla \psi_{i,j}(x(t_{\mathrm{s}}(x_0; x_0)))^{\top} f_i(x)} \\ &- f_i(x) \frac{\nabla \psi_{i,j}(x(t_{\mathrm{s}}(x_0; x_0)))^{\top} S^{x}(t_{\mathrm{s}}^{-}, 0; x_0)}{\nabla \psi_{i,j}(x(t_{\mathrm{s}}(x_0; x_0)))^{\top} f_i(x)} S^{x}(t_{\mathrm{s}}^{-}, 0; x_0) + S^{x}(t_{\mathrm{s}}^{-}, 0; x_0) \end{aligned} \tag{9}$$

Finally, from the chain rule we have $S^{x}(T, 0, x_0) = S^{x}(T, t_{\mathrm{s}}^{+}, x_0) S^{x}(t_{\mathrm{s}}^{+}, 0, x_0)$ and (9) we obtain (3).

Switch detection - example

Suppose that $x(t)$ crosses from R_1 to R_2 and recall that $\mu = \min_j g_j(x)$

Continuous time:

- Before switch: $\theta_1(t) > 0, \lambda_1(t) = 0$, and $\theta_2(t) = 0, \lambda_2 \geq 0$
- After switch: $\theta_1(t) = 0, \lambda_1(t) \geq 0$, and $\theta_2(t) > 0, \lambda_2 = 0$

Switch detection - example

Suppose that $x(t)$ crosses from R_1 to R_2 and recall that $\mu = \min_j g_j(x)$
Discrete time (switch between the n-th and $n+1$-st finite element):

- Before switch: $\theta_{n,j,1} > 0, \lambda_{n,j,1} = 0$, and $\theta_{n,j,2} = 0, \lambda_{n,j,2} \geq 0$
- After switch: $\theta_{n,j,1} = 0, \lambda_{n,j,1} > 0$, and $\theta_{n,j,2} > 0, \lambda_{n,j,2} = 0$

Switch detection - example

Suppose that $x(t)$ crosses from R_1 to R_2 and recall that $\mu = \min_j g_j(x)$
Discrete time (switch between the n-th and $n+1$-st finite element):

- Before switch: $\theta_{n,j,1} > 0, \lambda_{n,j,1} = 0$, and $\theta_{n,j,2} = 0, \lambda_{n,j,2} \geq 0$
- After switch: $\theta_{n,j,1} = 0, \lambda_{n,j,1} > 0$, and $\theta_{n,j,2} > 0, \lambda_{n,j,2} = 0$

From Lemma 1 it follows that $\lambda_{n,n_\mathrm{s},1} = \lambda_{n,n_\mathrm{s},2} = 0$

Switch detection conditions

$$g_1(x_{n+1}) = \lambda_{n,n_\mathrm{s},1} - \mu_{n,n_\mathrm{s}}$$

Switch detection - example

Suppose that $x(t)$ crosses from R_1 to R_2 and recall that $\mu = \min_j g_j(x)$
Discrete time (switch between the n-th and $n+1$-st finite element):

- Before switch: $\theta_{n,j,1} > 0, \lambda_{n,j,1} = 0$, and $\theta_{n,j,2} = 0, \lambda_{n,j,2} \geq 0$
- After switch: $\theta_{n,j,1} = 0, \lambda_{n,j,1} > 0$, and $\theta_{n,j,2} > 0, \lambda_{n,j,2} = 0$

From Lemma 1 it follows that $\lambda_{n,n_\mathrm{s},1} = \lambda_{n,n_\mathrm{s},2} = 0$

Switch detection condition

$$g_1(x_{n+1}) = 0 - g_2(x_{n+1})$$

Switch detection - example

Suppose that $x(t)$ crosses from R_1 to R_2 and recall that $\mu = \min_j g_j(x)$
Discrete time (switch between the n-th and $n+1$-st finite element):

- Before switch: $\theta_{n,j,1} > 0, \lambda_{n,j,1} = 0$, and $\theta_{n,j,2} = 0, \lambda_{n,j,2} \geq 0$
- After switch: $\theta_{n,j,1} = 0, \lambda_{n,j,1} > 0$, and $\theta_{n,j,2} > 0, \lambda_{n,j,2} = 0$

From Lemma 1 it follows that $\lambda_{n,n_{\mathrm{s}},1} = \lambda_{n,n_{\mathrm{s}},2} = 0$

Switch detection conditions

$$0 = g_1(x_{n+1}) - g_2(x_{n+1}) = \psi_{12}(x_{n+1})$$

Implies constraint such that h_n must adapt for exact switch detection!

Summary of FESD theoretical results

1. An FESD problem needs to solve a nonlinear complementarity problem (NCP) to advance the integration. The solutions of these NCP are locally unique.
   - For a given point determine which constraint cross comp. and step eq. are binding, and which implicitly satisfied.
   - Obtain square system and apply implicit function theorem.

2. Convergence of the FESD method to a Filippov solution of the underlying system with accuracy $O(h^p)$ is proven. Here, p is the order of the underlying smooth IRK method.
   - Solution approximation and true solution predict same active set.
   - Switching time accuracy also $O(h^p)$.

Summary of FESD theoretical results

1. An FESD problem needs to solve a nonlinear complementarity problem (NCP) to advance the integration. The solutions of these NCP are locally unique.
    - For a given point determine which constraint cross comp. and step eq. are binding, and which implicitly satisfied.
    - Obtain square system and apply implicit function theorem.

2. Convergence of the FESD method to a Filippov solution of the underlying system with accuracy $O(h^p)$ is proven. Here, p is the order of the underlying smooth IRK method.
    - Solution approximation and true solution predict same active set.
    - Switching time accuracy also $O(h^p)$.

3. Convergence of numerical sensitivities to the true value with $O(h^p)$ is given.
    - Cross. comp. implicitly enforce switching condition and lead to correct sensitivities.
    - The Stewart & Anitescu problem is solved.

Optimal control benchmark with FESD

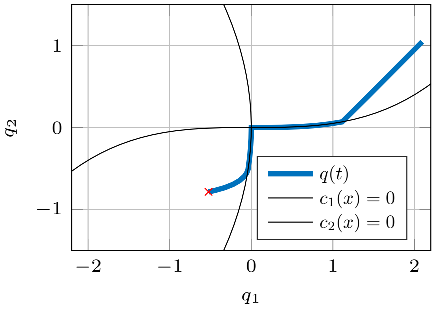
Benchmark example with entering/leaving sliding mode

OCP with sliding modes

$$
\begin{aligned}
\min_{x(\cdot),u(\cdot)} \quad & \int_0^4 u(t)^\top u(t) + v(t)^\top v(t)\,\mathrm{d}t \\
\text{s.t.} \quad & x(0) = \left(\frac{2\pi}{3}, \frac{\pi}{3}, 0, 0\right) \\
& \dot{x}(t) = \begin{bmatrix} -\mathrm{sign}(c(x(t))) + v(t) \\ u(t) \end{bmatrix} \\
& -2e \leq v(t) \leq 2e \\
& -10e \leq u(t) \leq 10e \quad t \in [0,4], \\
& q(T) = \left(-\frac{\pi}{6}, -\frac{\pi}{4}\right)
\end{aligned}
$$

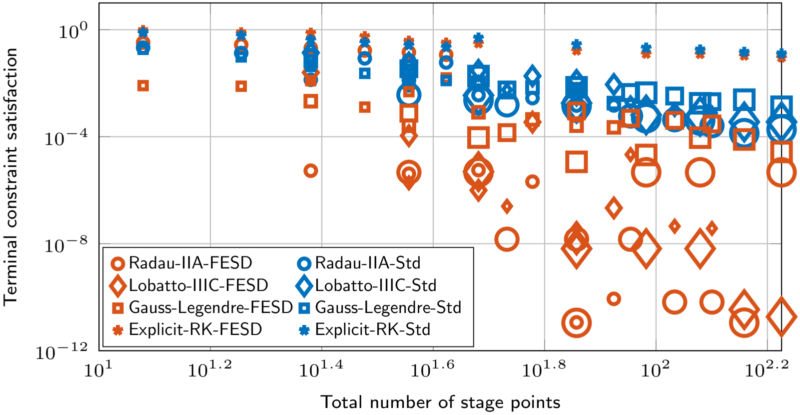
States $q, v \in \mathbb{R}^2$ and control $u \in \mathbb{R}^2$, $x = (q, v)$

Switching functions $c(x) = \begin{bmatrix} q_1 + 0.15q_2^2 \\ 0.05q_1^3 + q_2 \end{bmatrix}$



FESD vs standard IRK - number of function evaluations

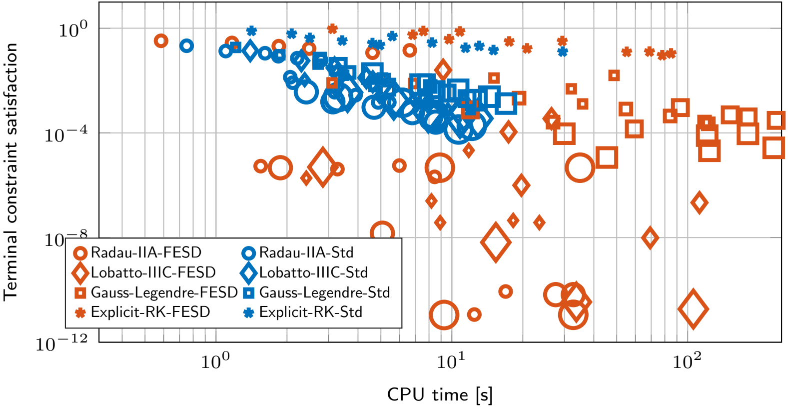
Benchmark on an optimal control problem with nonlinear sliding modes



Terminal constraint satisfaction vs. number of stage points

FESD vs standard IRK - CPU Time

Benchmark on an optimal control problem with nonlinear sliding modes



Terminal constraint satisfaction vs. CPU time

FESD one million times more accurate than Std. for CPU time of ≈ 2 s