

Modeling with mathematical programs with complementarity constraints

Summer School on Nonlinear Optimization with Mixed Integer and Complementarity
Constraints – NOMICC26

Moritz Diehl, Andrea Ghezzi, Sven Leyffer, and Armin Nurkanović

University of Freiburg, Argonne National Laboratory

September 29, 2026

Modeling with mathematical programs with complementarity constraints

1. What are mathematical programs with complementarity constraints?
2. Modeling piecewise smooth functions
3. Advanced complementarity modeling
4. Applications of complementarity constraints
5. Summary

A quick guide to notation across the lectures

Notation changes

Notation may change between lecture blocks, but it is used consistently within each lecture.

Notation in the MINLP lectures

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n, y \in \mathbb{Z}^p} & f(x, y) \\ \text{s.t.} & g(x, y) \leq 0. \end{array}$$

- ▶ x denotes continuous variables.
- ▶ y denotes integer decisions.
- ▶ Inequalities are written as $g(x, y) \leq 0$.
- ▶ Acronyms: **MINLP/MINLO**: mixed-integer nonlinear programming/optimization.

A quick guide to notation across the lectures

Notation changes

Notation may change between lecture blocks, but it is used consistently within each lecture.

Notation in the MINLP lectures

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n, y \in \mathbb{Z}^p} & f(x, y) \\ \text{s.t.} & g(x, y) \leq 0. \end{array}$$

- ▶ x denotes continuous variables.
- ▶ y denotes integer decisions.
- ▶ Inequalities are written as $g(x, y) \leq 0$.
- ▶ Acronyms: **MINLP/MINLO**: mixed-integer nonlinear programming/optimization.

Notation in the NLP/MPCC lectures

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t.} & g(x) = 0, \\ & h(x) \geq 0 \end{array}$$

- ▶ There is no y : NLPs and MPCCs have no integer variables.
- ▶ Equality constraints are kept explicit.
- ▶ Inequalities are written as $h(x) \geq 0$.

A quick guide to notation across the lectures

Notation changes

Notation may change between lecture blocks, but it is used consistently within each lecture.

Notation in the MINLP lectures

$$\begin{aligned} \min_{x \in \mathbb{R}^n, y \in \mathbb{Z}^p} \quad & f(x, y) \\ \text{s.t.} \quad & g(x, y) \leq 0. \end{aligned}$$

- ▶ x denotes continuous variables.
- ▶ y denotes integer decisions.
- ▶ Inequalities are written as $g(x, y) \leq 0$.
- ▶ Acronyms: **MINLP/MINLO**: mixed-integer nonlinear programming/optimization.

Notation in the NLP/MPCC lectures

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0 \end{aligned}$$

- ▶ There is no y : NLPs and MPCCs have no integer variables.
- ▶ Equality constraints are kept explicit.
- ▶ Inequalities are written as $h(x) \geq 0$.

Optimal control and MPC

If we have lot of differential equations, x is reserved for states, w for generic optimization variables.

Mathematical programs with complementarity constraints (MPCC)

MPCC

$$\min_{x \in \mathbb{R}^n} f(x) \quad (1a)$$

$$\text{s.t.} \quad g(x) = 0, \quad (1b)$$

$$h(x) \geq 0, \quad (1c)$$

$$0 \leq G(x) \perp H(x) \geq 0. \quad (1d)$$

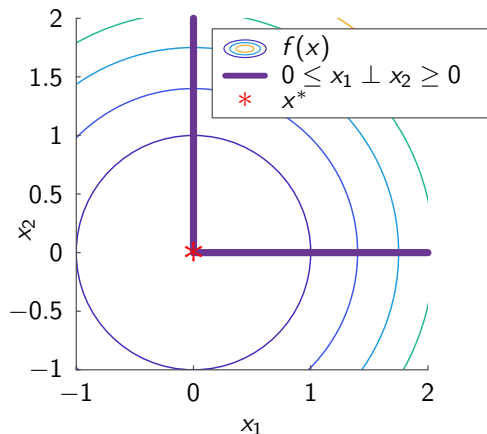
Complementarity constraints

$0 \leq G(x) \perp H(x) \geq 0$ means:

$$G(x) \geq 0, \quad H(x) \geq 0,$$

$$G_i(x) = 0 \quad \text{OR} \quad H_i(x) = 0, \quad \forall i$$

Optimality conditions, game theory, and contact mechanics often lead to complementarity conditions.



$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & x_1^2 + x_2^2 \\ \text{s.t.} \quad & 0 \leq x_1 \perp x_2 \geq 0. \end{aligned}$$

Smooth-looking reformulations of complementarity constraints

The MPCC can be reformulated as an NLP by replacing (1d) with:

1. $G(x) \geq 0, H(x) \geq 0, G_i(x)H_i(x) \leq 0, \quad \forall i$

Smooth-looking reformulations of complementarity constraints

The MPCC can be reformulated as an NLP by replacing (1d) with:

1. $G(x) \geq 0, H(x) \geq 0, G_i(x)H_i(x) \leq 0, \quad \forall i$
2. $G(x) \geq 0, H(x) \geq 0, G(x)^\top H(x) \leq 0,$

Smooth-looking reformulations of complementarity constraints

The MPCC can be reformulated as an NLP by replacing (1d) with:

1. $G(x) \geq 0, H(x) \geq 0, G_i(x)H_i(x) \leq 0, \quad \forall i$
2. $G(x) \geq 0, H(x) \geq 0, G(x)^\top H(x) \leq 0,$
3. $G(x) \geq 0, H(x) \geq 0, G(x)^\top H(x) = 0,$

Smooth-looking reformulations of complementarity constraints

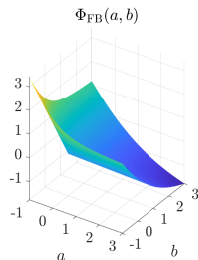
The MPCC can be reformulated as an NLP by replacing (1d) with:

1. $G(x) \geq 0, H(x) \geq 0, G_i(x)H_i(x) \leq 0, \quad \forall i$
2. $G(x) \geq 0, H(x) \geq 0, G(x)^\top H(x) \leq 0,$
3. $G(x) \geq 0, H(x) \geq 0, G(x)^\top H(x) = 0,$
4. $G(x) \geq 0, H(x) \geq 0, G_i(x)H_i(x) = 0, \quad \forall i$

Smooth-looking reformulations of complementarity constraints

The MPCC can be reformulated as an NLP by replacing (1d) with:

1. $G(x) \geq 0, H(x) \geq 0, G_i(x)H_i(x) \leq 0, \quad \forall i$
2. $G(x) \geq 0, H(x) \geq 0, G(x)^\top H(x) \leq 0,$
3. $G(x) \geq 0, H(x) \geq 0, G(x)^\top H(x) = 0,$
4. $G(x) \geq 0, H(x) \geq 0, G_i(x)H_i(x) = 0, \quad \forall i$
5. $\Phi_C(G(x), H(x)) = 0$



$$\Phi_{FB}(a, b) = \sqrt{a^2 + b^2} - a - b$$

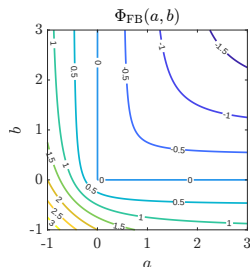
Smooth-looking reformulations do not remove the difficulty

Complementarity constraints need special theoretical and algorithmic treatment (Lectures 5 & 6).

Smooth-looking reformulations of complementarity constraints

The MPCC can be reformulated as an NLP by replacing (1d) with:

1. $G(x) \geq 0, H(x) \geq 0, G_i(x)H_i(x) \leq 0, \quad \forall i$
2. $G(x) \geq 0, H(x) \geq 0, G(x)^\top H(x) \leq 0,$
3. $G(x) \geq 0, H(x) \geq 0, G(x)^\top H(x) = 0,$
4. $G(x) \geq 0, H(x) \geq 0, G_i(x)H_i(x) = 0, \quad \forall i$
5. $\Phi_C(G(x), H(x)) = 0$



$$\Phi_{\text{FB}}(a, b) = \sqrt{a^2 + b^2} - a - b$$

Smooth-looking reformulations do not remove the difficulty

Complementarity constraints need special theoretical and algorithmic treatment (Lectures 5 & 6).

Cone complementarity problems

To learn more about CCPs see [Facchinei and Pang, 2003]

Dual cone

For a closed convex cone $K \subseteq \mathbb{R}^n$, its dual cone is

$$K^* := \{v \in \mathbb{R}^n \mid d^\top v \geq 0 \quad \forall d \in K\}.$$

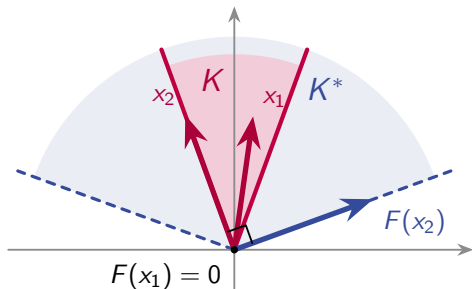
Cone complementarity problem (CCP)

Given $F : K \rightarrow \mathbb{R}^n$, find x such that

$$K \ni x \perp F(x) \in K^*.$$

Equivalently,

$$x \in K, F(x) \in K^*, x^\top F(x) = 0.$$



Cone complementarity problems

To learn more about CCPs see [Facchinei and Pang, 2003]

Dual cone

For a closed convex cone $K \subseteq \mathbb{R}^n$, its dual cone is

$$K^* := \{v \in \mathbb{R}^n \mid d^\top v \geq 0 \quad \forall d \in K\}.$$

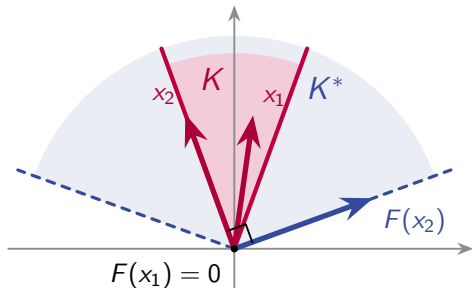
Cone complementarity problem (CCP)

Given $F : K \rightarrow \mathbb{R}^n$, find x such that

$$K \ni x \perp F(x) \in K^*.$$

Equivalently,

$$x \in K, F(x) \in K^*, x^\top F(x) = 0.$$



Special case that we treat: $K = K^* = \mathbb{R}_{\geq 0}^n$,
 $0 \leq x \perp F(x) \geq 0$.

Cone complementarity problems

To learn more about CCPs see [Facchinei and Pang, 2003]

Dual cone

For a closed convex cone $K \subseteq \mathbb{R}^n$, its dual cone is

$$K^* := \{v \in \mathbb{R}^n \mid d^\top v \geq 0 \quad \forall d \in K\}.$$

Cone complementarity problem (CCP)

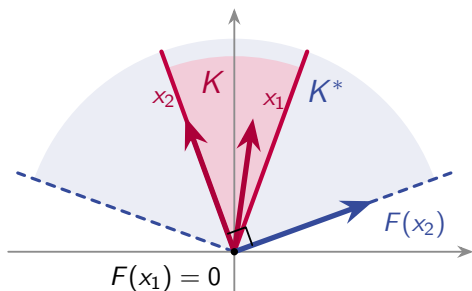
Given $F : K \rightarrow \mathbb{R}^n$, find x such that

$$K \ni x \perp F(x) \in K^*.$$

Equivalently,

$$x \in K, F(x) \in K^*, x^\top F(x) = 0.$$

Most concepts for standard MPCCs extend readily to mathematical programs with cone complementarity constraints [Ding et al., 2014, Ye and Zhou, 2016, Liang et al., 2026].



Special case that we treat: $K = K^* = \mathbb{R}_{\geq 0}^n$,
 $0 \leq x \perp F(x) \geq 0$.

Vertical form of an MPCC

Usually, we keep the complementarity constraints as simple as possible:

Generic MPCC

$$\begin{array}{ll}\min_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t.} & g(x) = 0, \\ & h(x) \geq 0, \\ & 0 \leq G(x) \perp H(x) \geq 0.\end{array}$$

Vertical form of an MPCC

Usually, we keep the complementarity constraints as simple as possible:

Vertical form MPCC

$$\begin{array}{ll}\min_{\substack{x \in \mathbb{R}^n, \\ x_1, x_2 \in \mathbb{R}^m}} & f(x) \\ \text{s.t.} & g(x) = 0, \\ & h(x) \geq 0, \\ & x_1 - G(x) = 0, \\ & x_2 - H(x) = 0, \\ & 0 \leq x_1 \perp x_2 \geq 0.\end{array}$$

Vertical form of an MPCC

Usually, we keep the complementarity constraints as simple as possible:

Vertical form MPCC

$$\begin{array}{ll}\min_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t.} & g(x) = 0, \\ & h(x) \geq 0, \\ & 0 \leq x_1 \perp x_2 \geq 0.\end{array}$$

where $x = (x_0, x_1, x_2) \in \mathbb{R}^n$, $x_0 \in \mathbb{R}^p$, $x_1, x_2 \in \mathbb{R}^m$, $n = p + 2m$.

Attention: some overloading of notation.

An incomplete list of MPCC applications

- ▶ robotics [Wensing et al., 2023]
- ▶ process and chemical engineering [Baumrucker et al., 2008, Biegler, 2010]
- ▶ district heating networks [Krug et al., 2021]
- ▶ optimal control of hybrid/nonsmooth constraints [Baumrucker and Biegler, 2009, Guo and Ye, 2016, Vieira et al., 2019, Nurkanović, 2023]
- ▶ optimization with piecewise smooth functions, abs-normal forms [Hegerhorst-Schultchen et al., 2020]
- ▶ bi-level optimization (if the lower level problem is convex) [Kim et al., 2020]
- ▶ modeling of logical constraints [Pozharskiy et al., 2024]
- ▶ inverse optimization [Albrecht and Ulbrich, 2017, Hu et al., 2012]
- ▶ ...

Some literature on MPECs I would recommend: [Luo et al., 1996, Kim et al., 2020, Hu and Ralph, 2002, Hu et al., 2012, Scheel and Scholtes, 2000, Flegel and Kanzow, 2005, Fletcher and Leyffer, 2004, Nurkanović et al., 2024, Kanzow and Schwartz, 2015, Hoheisel et al., 2013, Biegler, 2010, Hu et al., 2012]

Outline of the lecture

1. What are mathematical programs with complementarity constraints?
2. Modeling piecewise smooth functions
3. Advanced complementarity modeling
4. Applications of complementarity constraints
5. Summary

How complementarity constraints should **NOT** be used!

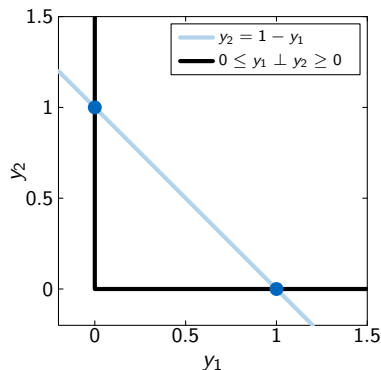
Integrality to complementarity

$$y_1 \in \{0, 1\}$$

can be equivalently written as:

$$y_2 = 1 - y_1$$

$$0 \leq y_1 \perp y_2 \geq 0$$



How complementarity constraints should **NOT** be used!

Integrity to complementarity

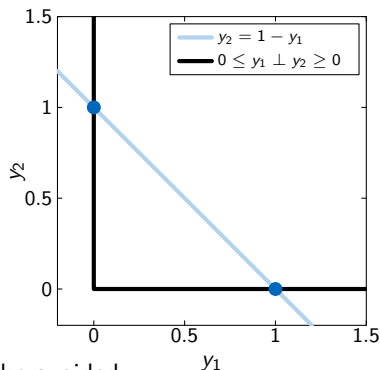
$$y_1 \in \{0, 1\}$$

can be equivalently written as:

$$y_2 = 1 - y_1$$

$$0 \leq y_1 \perp y_2 \geq 0$$

- ▶ Problems with **disconnected feasible regions** should be avoided.
- ▶ May converge only if initialized very close to solution.
- ▶ Bad idea as $\sin(y\pi) = 0$ to obtain $y \in \mathbb{Z}$.



How complementarity constraints should **NOT** be used!

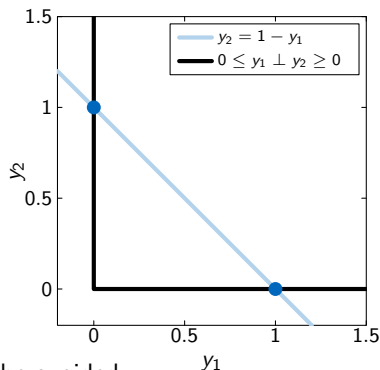
Integrity to complementarity

$$y_1 \in \{0, 1\}$$

can be equivalently written as:

$$y_2 = 1 - y_1$$

$$0 \leq y_1 \perp y_2 \geq 0$$



- ▶ Problems with **disconnected feasible regions** should be avoided.
- ▶ May converge only if initialized very close to solution.
- ▶ Bad idea as $\sin(y\pi) = 0$ to obtain $y \in \mathbb{Z}$.

However, the other direction is for algorithms less troublesome with $y \in \{0, 1\}$:

Complementarity to integrity

$$0 \leq x_1 \perp x_2 \geq 0 \iff \begin{aligned} 0 &\leq x_1 \leq yM, \\ 0 &\leq x_2 \leq (1 - y)M. \end{aligned}$$

Bilevel optimization problems – a rich source of MPCCs

Piecewise-smooth modeling via KKT reformulation

Bilevel optimization problem

$$\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} f(x, z) \quad (2a)$$

$$\text{s.t.} \quad g(x, z) = 0, \quad (2b)$$

$$h(x, z) \geq 0, \quad (2c)$$

$$z \in \arg \min_{v \in \mathbb{R}^m} \{F(x, v) \text{ s.t. } G(x, v) = 0, H(x, v) \geq 0\}. \quad (2d)$$

- The constraints of the *upper-level problem* include another optimization problem (2d), called the *lower-level problem*. Stackelberg games are a prominent example.

Bilevel optimization problems – a rich source of MPCCs

Piecewise-smooth modeling via KKT reformulation

Bilevel optimization problem

$$\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} f(x, z) \quad (2a)$$

$$\text{s.t.} \quad g(x, z) = 0, \quad (2b)$$

$$h(x, z) \geq 0, \quad (2c)$$

$$z \in \arg \min_{v \in \mathbb{R}^m} \{F(x, v) \text{ s.t. } G(x, v) = 0, H(x, v) \geq 0\}. \quad (2d)$$

- ▶ The constraints of the *upper-level problem* include another optimization problem (2d), called the *lower-level problem*. Stackelberg games are a prominent example.
- ▶ The lower-level problem is parameterized by the upper-level variable x .

Bilevel optimization problems – a rich source of MPCCs

Piecewise-smooth modeling via KKT reformulation

Bilevel optimization problem

$$\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} f(x, z) \quad (2a)$$

$$\text{s.t.} \quad g(x, z) = 0, \quad (2b)$$

$$h(x, z) \geq 0, \quad (2c)$$

$$z \in \arg \min_{v \in \mathbb{R}^m} \{F(x, v) \text{ s.t. } G(x, v) = 0, H(x, v) \geq 0\}. \quad (2d)$$

- ▶ The constraints of the *upper-level problem* include another optimization problem (2d), called the *lower-level problem*. Stackelberg games are a prominent example.
- ▶ The lower-level problem is parameterized by the upper-level variable x .
- ▶ Its solution map $x \mapsto z^*(x)$ implicitly defines (under some regularity conditions) piecewise-smooth functions.

Bilevel optimization problems – a rich source of MPCCs

Piecewise-smooth modeling via KKT reformulation

Bilevel optimization problem

$$\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} f(x, z) \quad (2a)$$

$$\text{s.t.} \quad g(x, z) = 0, \quad (2b)$$

$$h(x, z) \geq 0, \quad (2c)$$

$$z \in \arg \min_{v \in \mathbb{R}^m} \{F(x, v) \text{ s.t. } G(x, v) = 0, H(x, v) \geq 0\}. \quad (2d)$$

- ▶ The constraints of the *upper-level problem* include another optimization problem (2d), called the *lower-level problem*. Stackelberg games are a prominent example.
- ▶ The lower-level problem is parameterized by the upper-level variable x .
- ▶ Its solution map $x \mapsto z^*(x)$ implicitly defines (under some regularity conditions) piecewise-smooth functions.
- ▶ If the lower-level problem is convex and satisfies suitable regularity conditions, it can be replaced by its Karush–Kuhn–Tucker (KKT) conditions, yielding an **MPCC**.

Guidelines for good MPCC modeling

Based on Chapter 11 of L. Biegler's textbook on nonlinear programming [Biegler, 2010].

1. To model discrete decisions, piecewise smooth functions, ..., start with a convex *lower level* problem, e.g.,

$$\min_z F(x_0)z \quad \text{s.t.} \quad z_\ell \leq z \leq z_u.$$

Problem is parametric in switching function $F(x_0)$.

Guidelines for good MPCC modeling

Based on Chapter 11 of L. Biegler's textbook on nonlinear programming [Biegler, 2010].

1. To model discrete decisions, piecewise smooth functions, ..., start with a convex *lower level* problem, e.g.,

$$\min_z F(x_0)z \quad \text{s.t.} \quad z_\ell \leq z \leq z_u.$$

Problem is parametric in switching function $F(x_0)$.

2. Whenever possible, formulate the problem such that constraints on upper level variables x_0 do not interfere with lower level constraints – this helps to avoid disconnected feasible sets.

Guidelines for good MPCC modeling

Based on Chapter 11 of L. Biegler's textbook on nonlinear programming [Biegler, 2010].

1. To model discrete decisions, piecewise smooth functions, ..., start with a convex *lower level* problem, e.g.,

$$\min_z F(x_0)z \quad \text{s.t.} \quad z_\ell \leq z \leq z_u.$$

Problem is parametric in switching function $F(x_0)$.

2. Whenever possible, formulate the problem such that constraints on upper level variables x_0 do not interfere with lower level constraints – this helps to avoid disconnected feasible sets.
3. Apply KKT conditions, to reformulate bilevel problem to MPCC:

$$F(x_0) - \lambda_\ell + \lambda_u = 0,$$

$$0 \leq z - z_\ell \perp \lambda_\ell \geq 0,$$

$$0 \leq z_u - z \perp \lambda_u \geq 0.$$

Guidelines for good MPCC modeling

Based on Chapter 11 of L. Biegler's textbook on nonlinear programming [Biegler, 2010].

1. To model discrete decisions, piecewise smooth functions, ..., start with a convex *lower level* problem, e.g.,

$$\min_z F(x_0)z \quad \text{s.t.} \quad z_\ell \leq z \leq z_u.$$

Problem is parametric in switching function $F(x_0)$.

2. Whenever possible, formulate the problem such that constraints on upper level variables x_0 do not interfere with lower level constraints – this helps to avoid disconnected feasible sets.
3. Apply KKT conditions, to reformulate bilevel problem to MPCC:

$$F(x_0) - \lambda_\ell + \lambda_u = 0,$$

$$0 \leq z - z_\ell \perp \lambda_\ell \geq 0,$$

$$0 \leq z_u - z \perp \lambda_u \geq 0.$$

4. If possible, do some variable eliminations in the KKT conditions.

Modeling the Heaviside step and $\text{sign}(\cdot)$ functions

Set-valued Heaviside step function

$$\text{step}(F(x)) = \begin{cases} \{1\}, & F(x) > 0, \\ [0, 1], & F(x) = 0, \\ \{0\}, & F(x) < 0. \end{cases}$$

$$\text{step}(F(x)) = \underset{z \in \mathbb{R}}{\text{argmin}} \quad -F(x)z \quad (3a)$$

$$\text{s.t.} \quad 0 \leq z \leq 1. \quad (3b)$$

Modeling the Heaviside step and $\text{sign}(\cdot)$ functions

Set-valued Heaviside step function

$$\text{step}(F(x)) = \begin{cases} \{1\}, & F(x) > 0, \\ [0, 1], & F(x) = 0, \\ \{0\}, & F(x) < 0. \end{cases}$$

$$\text{step}(F(x)) = \underset{z \in \mathbb{R}}{\text{argmin}} \quad -F(x)z \quad (3a)$$

$$\text{s.t.} \quad 0 \leq z \leq 1. \quad (3b)$$

KKT conditions of the LP (3):

$$\begin{aligned} F(x) &= \lambda_p - \lambda_n, \\ 0 \leq z \perp \lambda_n &\geq 0, \\ 0 \leq 1 - z \perp \lambda_p &\geq 0. \end{aligned}$$

Modeling the Heaviside step and $\text{sign}(\cdot)$ functions

Set-valued Heaviside step function

$$\text{step}(F(x)) = \begin{cases} \{1\}, & F(x) > 0, \\ [0, 1], & F(x) = 0, \\ \{0\}, & F(x) < 0. \end{cases} \quad \begin{aligned} \text{step}(F(x)) = \operatorname{argmin}_{z \in \mathbb{R}} \quad & -F(x)z & (3a) \\ \text{s.t.} \quad & 0 \leq z \leq 1. & (3b) \end{aligned}$$

KKT conditions of the LP (3):

$$\begin{aligned} F(x) &= \lambda_p - \lambda_n, \\ 0 \leq z \perp \lambda_n &\geq 0, \\ 0 \leq 1 - z \perp \lambda_p &\geq 0. \end{aligned}$$

- ▶ For $F(x) > 0 \implies z = 1, \lambda_n = 0$ positive part: $\lambda_p = \max(0, F(x))$.
- ▶ For $F(x) < 0 \implies z = 0, \lambda_p = 0$, negative part: $\lambda_n = \max(0, -F(x))$.
- ▶ $\text{sign}(F(x))$ by setting lower bound to -1 in LP (3), in this case $|F(x)| = F(x)z$.
- ▶ Cannot model discontinuous functions, only their set-valued extensions.

Modeling the `abs(·)` function

Modeling the $\text{abs}(\cdot)$ function

- **Formulation 1:** To model $t = |F(x)|$, use set-valued $z = \text{sign}(F(x))$ function and identity:

$$t = |F(x)| = F(x)z.$$

Modeling the $\text{abs}(\cdot)$ function

- **Formulation 1:** To model $t = |F(x)|$, use set-valued $z = \text{sign}(F(x))$ function and identity:

$$t = |F(x)| = F(x)z.$$

- **Formulation 2:** To model $t = |F(x)|$, use set-valued $z = \text{step}(F(x))$ function and identity:

$$t = |F(x)| = zF(x) + (1 - z)(-F(x)) = (2z - 1)F(x)$$

Modeling the $\text{abs}(\cdot)$ function

- **Formulation 1:** To model $t = |F(x)|$, use set-valued $z = \text{sign}(F(x))$ function and identity:

$$t = |F(x)| = F(x)z.$$

- **Formulation 2:** To model $t = |F(x)|$, use set-valued $z = \text{step}(F(x))$ function and identity:

$$t = |F(x)| = zF(x) + (1 - z)(-F(x)) = (2z - 1)F(x)$$

- **Formulation 3 – THE BEST**, use as little comps. as possible. To model $t = |F(x)|$, use single complementarity constraints:

$$\begin{aligned} F(x) &= \lambda^p - \lambda^n \\ 0 &\leq \lambda^p \perp \lambda^n \geq 0 \end{aligned}$$

Combining the positive and negative part, $t = \lambda^p + \lambda^n$

Practical guide

Good modeling is one complementarity constraint pair per kink.

Modeling max and min of two functions

Using positive and negative parts

Positive–negative decomposition

For any scalar-valued function $r(x)$, introduce λ_p, λ_n through

$$r(x) = \lambda_p - \lambda_n, \quad 0 \leq \lambda_p \perp \lambda_n \leq 0.$$

This uniquely determines

$$\lambda_p = \max(0, r(x)) \text{ (also known as ReLU),} \quad \lambda_n = -\min(0, r(x)).$$

Modeling max and min of two functions

Using positive and negative parts

Positive–negative decomposition

For any scalar-valued function $r(x)$, introduce λ_p, λ_n through

$$r(x) = \lambda_p - \lambda_n, \quad 0 \leq \lambda_p \perp \lambda_n \leq 0.$$

This uniquely determines

$$\lambda_p = \max(0, r(x)) \text{ (also known as ReLU),} \quad \lambda_n = -\min(0, r(x)).$$

Apply this decomposition to

$$r(x) = F_1(x) - F_2(x).$$

Then

$$\begin{aligned} \max(F_1(x), F_2(x)) &= F_2(x) + \lambda_p = F_1(x) + \lambda_n, \\ \min(F_1(x), F_2(x)) &= F_1(x) - \lambda_p = F_2(x) - \lambda_n. \end{aligned}$$

An alternative formulation using the step function

Correct, but unnecessarily indirect

Define z through the parametric LP

$$z \in \arg \min_{0 \leq z \leq 1} (F_2(x) - F_1(x))z.$$

Then

$$\begin{aligned} \max(F_1(x), F_2(x)) &= zF_1(x) + (1 - z)F_2(x), \\ \min(F_1(x), F_2(x)) &= (1 - z)F_1(x) + zF_2(x). \end{aligned}$$

An alternative formulation using the step function

Correct, but unnecessarily indirect

Define z through the parametric LP

$$z \in \arg \min_{0 \leq z \leq 1} (F_2(x) - F_1(x))z.$$

Then

$$\begin{aligned} \max(F_1(x), F_2(x)) &= zF_1(x) + (1 - z)F_2(x), \\ \min(F_1(x), F_2(x)) &= (1 - z)F_1(x) + zF_2(x). \end{aligned}$$

- ▶ If $F_1(x) > F_2(x)$, then $z = 1$.
- ▶ If $F_1(x) < F_2(x)$, then $z = 0$.
- ▶ If $F_1(x) = F_2(x)$, every $z \in [0, 1]$ gives the same value.

An alternative formulation using the step function

Correct, but unnecessarily indirect

Define z through the parametric LP

$$z \in \arg \min_{0 \leq z \leq 1} (F_2(x) - F_1(x))z.$$

Then

$$\begin{aligned} \max(F_1(x), F_2(x)) &= zF_1(x) + (1 - z)F_2(x), \\ \min(F_1(x), F_2(x)) &= (1 - z)F_1(x) + zF_2(x). \end{aligned}$$

- ▶ If $F_1(x) > F_2(x)$, then $z = 1$.
- ▶ If $F_1(x) < F_2(x)$, then $z = 0$.
- ▶ If $F_1(x) = F_2(x)$, every $z \in [0, 1]$ gives the same value.

Why is this formulation less attractive?

The KKT reformulation of the step LP introduces complementarity conditions at both bounds, $z = 0$ and $z = 1$. The direct positive-negative-part formulation uses only one complementarity pair.

Minimum and maximum of several functions

Linear programs over the unit simplex

$$F(x) := (F_1(x), \dots, F_m(x)), \quad e := (1, \dots, 1)^T \in \mathbb{R}^m.$$

Minimum as an LP

$$\begin{array}{ll} \min_{z \in \mathbb{R}^m} & F(x)^T z \\ \text{s.t.} & e^T z = 1, \\ & z \geq 0. \end{array}$$

KKT conditions

$$\begin{aligned} F(x) - \mu e - \lambda &= 0, \\ e^T z &= 1, \\ 0 \leq z \perp \lambda &\geq 0, \end{aligned}$$

where $\mu \in \mathbb{R}$ and $\lambda \in \mathbb{R}^m$.

Minimum and maximum of several functions

Linear programs over the unit simplex

$$F(x) := (F_1(x), \dots, F_m(x)), \quad e := (1, \dots, 1)^T \in \mathbb{R}^m.$$

Minimum as an LP

$$\begin{aligned} \min_{z \in \mathbb{R}^m} \quad & F(x)^T z \\ \text{s.t.} \quad & e^T z = 1, \\ & z \geq 0. \end{aligned}$$

KKT conditions

$$\begin{aligned} F(x) - \mu e - \lambda &= 0, \\ e^T z &= 1, \\ 0 \leq z \perp \lambda &\geq 0, \end{aligned}$$

where $\mu \in \mathbb{R}$ and $\lambda \in \mathbb{R}^m$.

- ▶ Componentwise stationarity gives $F_i(x) = \mu + \lambda_i \geq \mu$
- ▶ $e^T z = 1$ implies $z_j > 0$ for some j .
Complementarity then gives $\lambda_j = 0$, and hence $\mu = F_j(x) = \min_i F_i(x)$.
- ▶ The maximum follows analogously:
 $\max_{i \in \{1, \dots, m\}} F_i(x) = \max_z z^T F(x) \text{ s.t. } e^T z = 1, z \geq 0.$

Minimum and maximum of several functions

Linear programs over the unit simplex

$$F(x) := (F_1(x), \dots, F_m(x)), \quad e := (1, \dots, 1)^T \in \mathbb{R}^m.$$

Minimum as an LP

$$\begin{aligned} \min_{z \in \mathbb{R}^m} \quad & F(x)^T z \\ \text{s.t.} \quad & e^T z = 1, \\ & z \geq 0. \end{aligned}$$

KKT conditions

$$\begin{aligned} F(x) - \mu e - \lambda &= 0, \\ e^T z &= 1, \\ 0 \leq z \perp \lambda &\geq 0, \end{aligned}$$

where $\mu \in \mathbb{R}$ and $\lambda \in \mathbb{R}^m$.

- ▶ Componentwise stationarity gives $F_i(x) = \mu + \lambda_i \geq \mu$
- ▶ $e^T z = 1$ implies $z_j > 0$ for some j .
Complementarity then gives $\lambda_j = 0$, and hence $\mu = F_j(x) = \min_i F_i(x)$.
- ▶ The maximum follows analogously:
 $\max_{i \in \{1, \dots, m\}} F_i(x) = \max_z z^T F(x) \text{ s.t. } e^T z = 1, z \geq 0.$

Question

What is the worst-case complexity of solving these particular LPs?

The ℓ_1 -norm via complementarity constraints

Positive and negative parts

For each component $F_i(x)$, introduce its positive and negative parts $\lambda_{p,i}$ and $\lambda_{n,i}$:

$$\begin{aligned} F_i(x) &= \lambda_{p,i} - \lambda_{n,i}, \quad i = 1, \dots, m, \\ 0 &\leq \lambda_{p,i} \perp \lambda_{n,i} \geq 0, \quad i = 1, \dots, m. \end{aligned}$$

We already have learned that the complementarity condition ensures

$$\begin{aligned} \lambda_{p,i} &= \max(0, F_i(x)), \\ \lambda_{n,i} &= \max(0, -F_i(x)), \end{aligned}$$

and therefore

$$\|F(x)\|_1 = \sum_{i=1}^m (\lambda_{p,i} + \lambda_{n,i}).$$

Question

How does a complementarity reformulation of $\|F(x)\|_\infty$ look like?

Exercise: Reformulating nonsmooth optimization problems

Find the “most convenient” equivalent formulation.

Hint: Be careful where the nonsmooth function appears.

Problem 1

$$\begin{aligned} \min_{x \in \mathbb{R}^3} \quad & \|x\|_1 \\ \text{s.t.} \quad & x_1 + 2x_2 + 3x_3 = 1, \\ & x \geq 0. \end{aligned}$$

Problem 2

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & \|x\|_1 \\ \text{s.t.} \quad & x_1 - 2x_2 = 5, \\ & x_2 \leq 3 \end{aligned}$$

Problem 3

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & x^\top x + g^\top x \\ \text{s.t.} \quad & \|x\|_1 = 10. \end{aligned}$$

Here, $g \in \mathbb{R}^n$ is given.

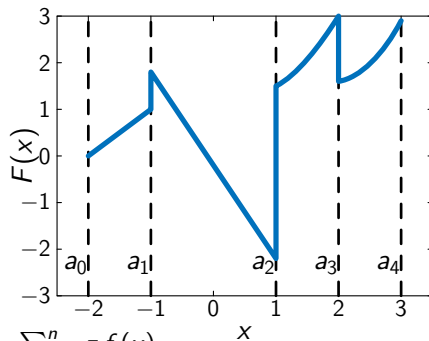
Piecewise smooth functions and look-up tables

Introduced in [Ragunathan and Biegler, 2003], generalizations follow later.

Model a piecewise smooth function $F : \mathbb{R} \rightarrow \mathbb{R}$ over intervals defined by grid points

$$a_0 < a_1 < \dots < a_n.$$

$$\begin{aligned} \min_{z \in \mathbb{R}^n} \quad & \sum_{i=1}^n (x - a_{i-1})(x - a_i) z_i \\ \text{s.t.} \quad & \sum_{i=1}^n z_i = 1, \\ & z_i \geq 0. \end{aligned}$$



- ▶ Compute the selector variables z_i , and set $F(x) = \sum_{i=1}^n z_i f_i(x)$.
- ▶ If $x \in [a_{i-1}, a_i]$, then $(x - a_{i-1})(x - a_i) \leq 0$; otherwise, it is positive.

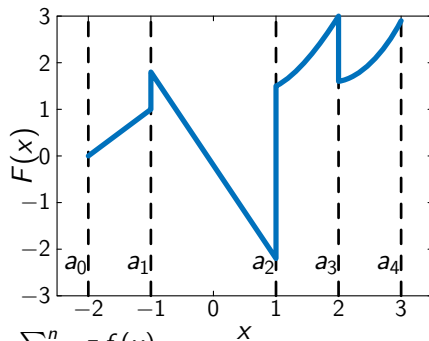
Piecewise smooth functions and look-up tables

Introduced in [Raghunathan and Biegler, 2003], generalizations follow later.

Model a piecewise smooth function $F : \mathbb{R} \rightarrow \mathbb{R}$ over intervals defined by grid points

$$a_0 < a_1 < \dots < a_n.$$

$$\begin{aligned} \min_{z \in \mathbb{R}^n} \quad & \sum_{i=1}^n (x - a_{i-1})(x - a_i) z_i \\ \text{s.t.} \quad & \sum_{i=1}^n z_i = 1, \\ & z \geq 0. \end{aligned}$$



- ▶ Compute the selector variables z_i , and set $F(x) = \sum_{i=1}^n z_i f_i(x)$.
- ▶ If $x \in [a_{i-1}, a_i]$, then $(x - a_{i-1})(x - a_i) \leq 0$; otherwise, it is positive.
- ▶ If the function is discontinuous, complementarity constraints allow only **set-valued extensions**.

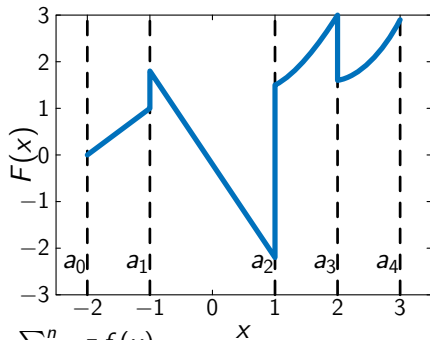
Piecewise smooth functions and look-up tables

Introduced in [Raghunathan and Biegler, 2003], generalizations follow later.

Model a piecewise smooth function $F : \mathbb{R} \rightarrow \mathbb{R}$ over intervals defined by grid points

$$a_0 < a_1 < \dots < a_n.$$

$$\begin{aligned} \min_{z \in \mathbb{R}^n} \quad & \sum_{i=1}^n (x - a_{i-1})(x - a_i) z_i \\ \text{s.t.} \quad & \sum_{i=1}^n z_i = 1, \\ & z \geq 0. \end{aligned}$$



- ▶ Compute the selector variables z_i , and set $F(x) = \sum_{i=1}^n z_i f_i(x)$.
- ▶ If $x \in [a_{i-1}, a_i]$, then $(x - a_{i-1})(x - a_i) \leq 0$; otherwise, it is positive.
- ▶ If the function is discontinuous, complementarity constraints allow only **set-valued extensions**.
- ▶ The approach can also be used for functions such as $\text{round}(\cdot)$.
- ▶ Generalizations to multidimensional input and output spaces via Stewart's LP follow later.

Outline of the lecture

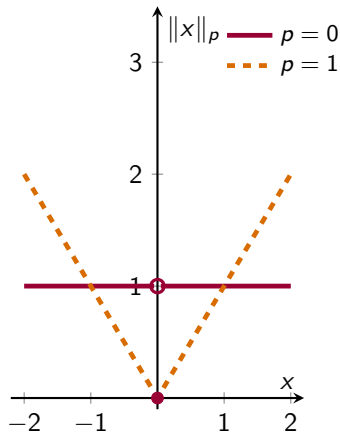
1. What are mathematical programs with complementarity constraints?
2. Modeling piecewise smooth functions
3. Advanced complementarity modeling
4. Applications of complementarity constraints
5. Summary

Sparsity optimization

The ℓ_0 pseudo-norm

$$\|x\|_0 := \text{card}(\{i \mid x_i \neq 0\}).$$

- ▶ True norms satisfy absolute homogeneity:
 $\|\alpha x\| = |\alpha| \|x\|,$
- ▶ Here, $\|\alpha x\|_0 = \|x\|_0$, $\alpha \neq 0$
and absolute homogeneity holds only for $\alpha = \pm 1$.



$$\|x\|_0 = \begin{cases} 0, & x = 0, \\ 1, & x \neq 0. \end{cases}$$

Sparsity optimization

The ℓ_0 pseudo-norm

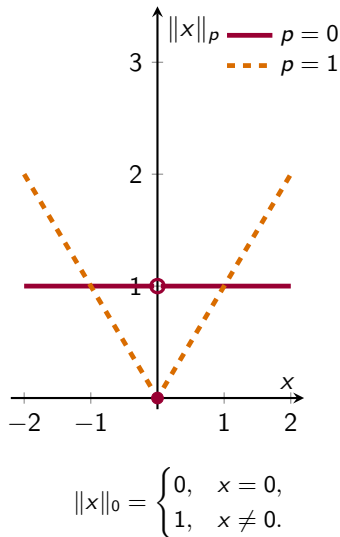
$$\|x\|_0 := \text{card}(\{i \mid x_i \neq 0\}).$$

- ▶ True norms satisfy absolute homogeneity:
 $\|\alpha x\| = |\alpha| \|x\|,$
- ▶ Here, $\|\alpha x\|_0 = \|x\|_0$, $\alpha \neq 0$
and absolute homogeneity holds only for $\alpha = \pm 1$.

Sparsity optimization (SPO)

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n} & f(x) + \rho \|x\|_0 \\ \text{s.t.} & g(x) = 0, \\ & h(x) \geq 0. \end{array}$$

$$\rho > 0$$



Sparsity optimization

The ℓ_0 pseudo-norm

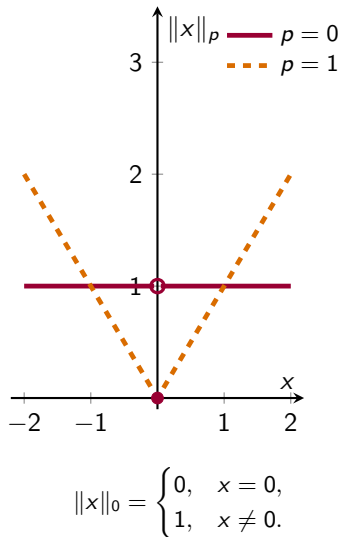
$$\|x\|_0 := \text{card}(\{i \mid x_i \neq 0\}).$$

- ▶ True norms satisfy absolute homogeneity:
 $\|\alpha x\| = |\alpha| \|x\|,$
- ▶ Here, $\|\alpha x\|_0 = \|x\|_0$, $\alpha \neq 0$
and absolute homogeneity holds only for $\alpha = \pm 1$.

Sparsity optimization (SPO)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) + \rho \|x\|_0 \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0. \end{aligned}$$

$$\rho > 0$$



Coding exercise:  [nomicc26/exercises/ell0-portfolio](https://github.com/nomicc26/exercises/ell0-portfolio)

Sparsity optimization with full complementarity constraints

Derived in [Feng et al., 2018]

Sparsity optimization

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) + \rho \|x\|_0 \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0. \end{aligned}$$

Full complementarity SPO [Feng et al., 2018]

$$\begin{aligned} \min_{x, x^\pm, z \in \mathbb{R}^n} \quad & f(x) + \rho \sum_{i=1}^n (1 - z_i) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0, \\ & x = x^+ - x^-, \\ & 0 \leq z \perp x^+ + x^- \leq 0, \\ & 0 \leq x^+ \perp x^- \leq 0, \\ & z \leq e. \end{aligned}$$

- ▶ At an optimal solution: if $x_i \neq 0$, then $z_i = 0$. If $x_i = 0$, then $z_i = 1$.
- ▶ The same reformulations can be used to handle cardinality constraints $\|x\|_0 \leq \kappa$, $\kappa \in \mathbb{N}$, but may have additional local minima [Kanzow et al., 2024].

Vanishing (inequality) constraints for $H(x) \geq 0$

Studied in detail in [Achtziger and Kanzow, 2008]

Vanishing constraint

If $H_i(x) > 0$ then $G_i(x) \leq 0$, else if $H_i(x) = 0$ then $G_i(x) \in \mathbb{R}$ ($G_i(x) \leq 0$ vanishes)

Vanishing (inequality) constraints for $H(x) \geq 0$

Studied in detail in [Achtziger and Kanzow, 2008]

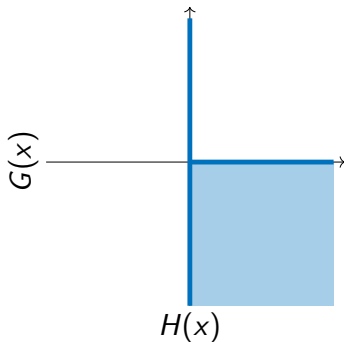
Vanishing constraint

If $H_i(x) > 0$ then $G_i(x) \leq 0$, else if $H_i(x) = 0$ then $G_i(x) \in \mathbb{R}$ ($G_i(x) \leq 0$ vanishes)

► Logical formulation:

$$H(x) \geq 0,$$

$$H_i(x) > 0 \implies G_i(x) \leq 0, \quad i = 1, \dots, m.$$



Vanishing (inequality) constraints for $H(x) \geq 0$

Studied in detail in [Achtziger and Kanzow, 2008]

Vanishing constraint

If $H_i(x) > 0$ then $G_i(x) \leq 0$, else if $H_i(x) = 0$ then $G_i(x) \in \mathbb{R}$ ($G_i(x) \leq 0$ vanishes)

► Logical formulation:

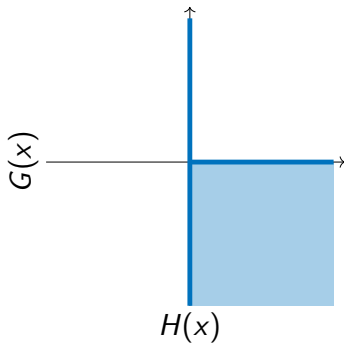
$$H(x) \geq 0,$$

$$H_i(x) > 0 \implies G_i(x) \leq 0, \quad i = 1, \dots, m.$$

► Nonlinear programming formulation:

$$H(x) \geq 0,$$

$$H_i(x)G_i(x) \leq 0, \quad i = 1, \dots, m.$$



Vanishing (inequality) constraints for $H(x) \geq 0$

Studied in detail in [Achtziger and Kanzow, 2008]

Vanishing constraint

If $H_i(x) > 0$ then $G_i(x) \leq 0$, else if $H_i(x) = 0$ then $G_i(x) \in \mathbb{R}$ ($G_i(x) \leq 0$ vanishes)

- Logical formulation:

$$H(x) \geq 0,$$

$$H_i(x) > 0 \implies G_i(x) \leq 0, \quad i = 1, \dots, m.$$

- Nonlinear programming formulation:

$$H(x) \geq 0,$$

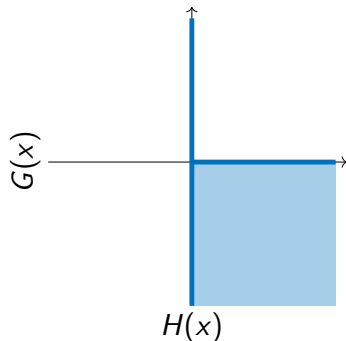
$$H_i(x)G_i(x) \leq 0, \quad i = 1, \dots, m.$$

- Lifted complementarity formulation:

$$G(x) - z \leq 0,$$

$$0 \leq z \perp H(x) \geq 0.$$

For $H_i(x) > 0$: $z = 0$, for $H_i(x) = 0$: $z \geq \max(G_i(x), 0)$.



Refresher on binary modeling

Let $0 \leq x \leq M$, where $M > 0$ is a known upper bound.

Implication of positivity

To model

$$x > 0 \implies y = 1,$$

use

$$x \leq My, \quad y \in \{0, 1\}.$$

- ▶ If $y = 0$, then $x = 0$.
- ▶ Consequently, $x > 0$ forces $y = 1$.
- ▶ The converse is not enforced: $y = 1$ does not imply $x > 0$.

Refresher on binary modeling

Let $0 \leq x \leq M$, where $M > 0$ is a known upper bound.

Implication of positivity

To model

$$x > 0 \implies y = 1,$$

use

$$x \leq My, \quad y \in \{0, 1\}.$$

- ▶ If $y = 0$, then $x = 0$.
- ▶ Consequently, $x > 0$ forces $y = 1$.
- ▶ The converse is not enforced: $y = 1$ does not imply $x > 0$.

Need valid upper bound $H_i(x) \leq M$.

Indicator variable

To impose a constraint only when $y_i = 1$,

$$y_i = 1 \implies H_i(x) \leq 0,$$

use big-M and impose

$$H_i(x) \leq M(1 - y_i), \quad y_i \in \{0, 1\}.$$

- ▶ If $y_i = 1$, then $H_i(x) \leq 0$ is enforced.
- ▶ If $y_i = 0$, only the known bound $H_i(x) \leq M$ remains.

Vanishing constraints with binary variables

Combine implication of positivity and indicator variable

$$H(x) > 0 \implies G(x) \leq 0.$$

Use a binary variable $y \in \{0, 1\}$ and a sufficiently large $M > 0$:

$$0 \leq H(x) \leq My, \quad G(x) \leq M(1 - y).$$

Vanishing constraints with binary variables

Combine implication of positivity and indicator variable

$$H(x) > 0 \implies G(x) \leq 0.$$

Use a binary variable $y \in \{0, 1\}$ and a sufficiently large $M > 0$:

$$0 \leq H(x) \leq My, \quad G(x) \leq M(1 - y).$$

Interpretation:

- Implication of positivity: if $H(x) > 0$, then $y = 1$ is forced by

$$H(x) \leq My.$$

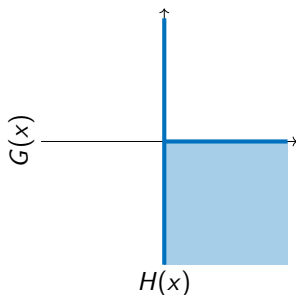
Indicator variables:

$$G(x) \leq M(1 - y) = 0.$$

- If $H(x) = 0$, then $y = 0$ or $y = 1$ is possible. For $y = 0$,

$$G(x) \leq M,$$

so the constraint $G(x) \leq 0$ is relaxed, it vanishes.



State-triggered constraints (vanishing equality constraints)

Introduced in [Szmuk et al., 2020]

State triggered constraints

If $H_i(x) < 0$ then $G_i(x) = 0$, otherwise if $H_i(x) \geq 0$ then $G_i(x) \in \mathbb{R}$

State-triggered constraints (vanishing equality constraints)

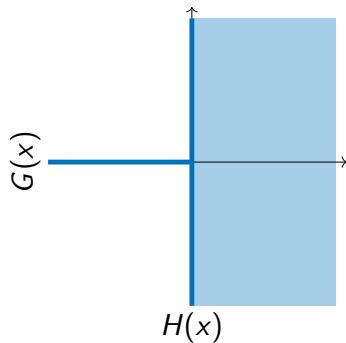
Introduced in [Szmuk et al., 2020]

State triggered constraints

If $H_i(x) < 0$ then $G_i(x) = 0$, otherwise if $H_i(x) \geq 0$ then $G_i(x) \in \mathbb{R}$

► Logical formulation:

$$H_i(x) < 0 \implies G_i(x) = 0, \quad i = 1, \dots, m.$$



State-triggered constraints (vanishing equality constraints)

Introduced in [Szmuk et al., 2020]

State triggered constraints

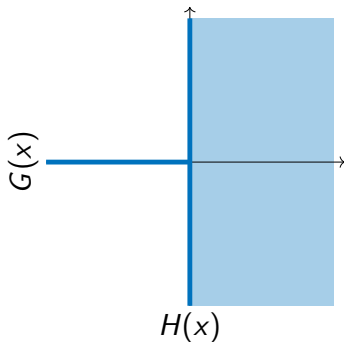
If $H_i(x) < 0$ then $G_i(x) = 0$, otherwise if $H_i(x) \geq 0$ then $G_i(x) \in \mathbb{R}$

- Logical formulation:

$$H_i(x) < 0 \implies G_i(x) = 0, \quad i = 1, \dots, m.$$

- Complementarity formulation:

$$\begin{aligned} z &\perp G(x), \\ 0 &\leq H(x) + z \perp z \geq 0. \end{aligned}$$



State-triggered constraints (vanishing equality constraints)

Introduced in [Szmuk et al., 2020]

State triggered constraints

If $H_i(x) < 0$ then $G_i(x) = 0$, otherwise if $H_i(x) \geq 0$ then $G_i(x) \in \mathbb{R}$

- ▶ Logical formulation:

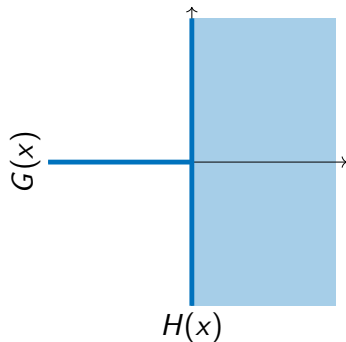
$$H_i(x) < 0 \implies G_i(x) = 0, \quad i = 1, \dots, m.$$

- ▶ Complementarity formulation:

$$\begin{aligned} z &\perp G(x), \\ 0 &\leq H(x) + z \perp z \geq 0. \end{aligned}$$

- ▶ Interpretation:

- ▶ $H_i(x) < 0 \implies z_i = -H_i(x) > 0 \implies G_i(x) = 0$,
- ▶ $H_i(x) > 0 \implies z_i = 0 \implies G_i(x) \in \mathbb{R}$,
- ▶ $H_i(x) = 0 \implies z_i = 0 \implies G_i(x) \in \mathbb{R}$.



State-triggered constraints with binaries

Goal

$$H_i(x) < 0 \implies G_i(x) = 0, \quad H_i(x) \geq 0 \implies G_i(x) \in \mathbb{R}.$$

Introduce $y_i \in \{0, 1\}$, where $y_i = 1$ means that $G_i(x) = 0$ is active.

Step 1: encode the sign of $H_i(x)$

$$H_i(x) \geq -M_i^H y_i, \quad H_i(x) \leq M_i^H (1 - y_i).$$

- ▶ $H_i(x) < 0 \Rightarrow y_i = 1$,
- ▶ $H_i(x) > 0 \Rightarrow y_i = 0$,
- ▶ $H_i(x) = 0 \Rightarrow y_i \in \{0, 1\}$.

Step 2: activate the equality constraint

$$-M_i^G (1 - y_i) \leq G_i(x) \leq M_i^G (1 - y_i).$$

$$y_i = 1 \Rightarrow G_i(x) = 0, \quad y_i = 0 \Rightarrow -M_i^G \leq G_i(x) \leq M_i^G.$$

State-triggered constraints with binaries

Goal

$$H_i(x) < 0 \implies G_i(x) = 0, \quad H_i(x) \geq 0 \implies G_i(x) \in \mathbb{R}.$$

Introduce $y_i \in \{0, 1\}$, where $y_i = 1$ means that $G_i(x) = 0$ is active.

Step 1: encode the sign of $H_i(x)$

$$H_i(x) \geq -M_i^H y_i, \quad H_i(x) \leq M_i^H (1 - y_i).$$

- ▶ $H_i(x) < 0 \Rightarrow y_i = 1$,
- ▶ $H_i(x) > 0 \Rightarrow y_i = 0$,
- ▶ $H_i(x) = 0 \Rightarrow y_i \in \{0, 1\}$.

Step 2: activate the equality constraint

$$-M_i^G (1 - y_i) \leq G_i(x) \leq M_i^G (1 - y_i).$$

$$y_i = 1 \Rightarrow G_i(x) = 0, \quad y_i = 0 \Rightarrow -M_i^G \leq G_i(x) \leq M_i^G.$$

State-triggered constraints with binaries

Goal

$$H_i(x) < 0 \implies G_i(x) = 0, \quad H_i(x) \geq 0 \implies G_i(x) \in \mathbb{R}.$$

Introduce $y_i \in \{0, 1\}$, where $y_i = 1$ means that $G_i(x) = 0$ is active.

Step 1: encode the sign of $H_i(x)$

$$H_i(x) \geq -M_i^H y_i, \quad H_i(x) \leq M_i^H (1 - y_i).$$

- ▶ $H_i(x) < 0 \Rightarrow y_i = 1$,
- ▶ $H_i(x) > 0 \Rightarrow y_i = 0$,
- ▶ $H_i(x) = 0 \Rightarrow y_i \in \{0, 1\}$.

Step 2: activate the equality constraint

$$-M_i^G (1 - y_i) \leq G_i(x) \leq M_i^G (1 - y_i).$$

$$y_i = 1 \Rightarrow G_i(x) = 0, \quad y_i = 0 \Rightarrow -M_i^G \leq G_i(x) \leq M_i^G.$$

Final formulation

$$y_i \in \{0, 1\}, \quad H_i(x) \geq -M_i^H y_i, \quad H_i(x) \leq M_i^H (1 - y_i), \\ -M_i^G (1 - y_i) \leq G_i(x) \leq M_i^G (1 - y_i).$$

State-triggered inequality constraints

For related formulation see the exercise and [Ghezzi et al., 2026]

State-triggered constraints

If $H_i(x) > 0$, enforce $G_i(x) \leq 0$. Otherwise, $G_i(x)$ is unrestricted.

State-triggered inequality constraints

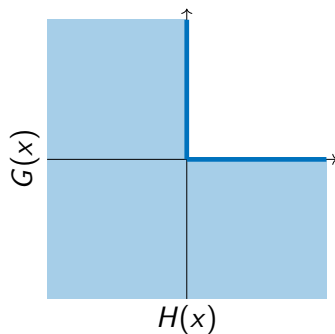
For related formulation see the exercise and [Ghezzi et al., 2026]

State-triggered constraints

If $H_i(x) > 0$, enforce $G_i(x) \leq 0$. Otherwise, $G_i(x)$ is unrestricted.

► Logical formulation:

$$H_i(x) > 0 \implies G_i(x) \leq 0, \quad i = 1, \dots, m.$$



State-triggered inequality constraints

For related formulation see the exercise and [Ghezzi et al., 2026]

State-triggered constraints

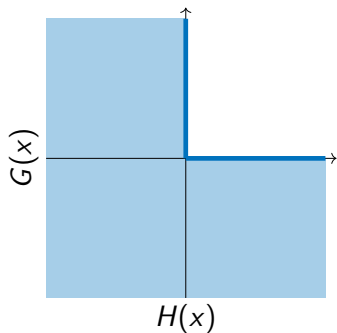
If $H_i(x) > 0$, enforce $G_i(x) \leq 0$. Otherwise, $G_i(x)$ is unrestricted.

- Logical formulation:

$$H_i(x) > 0 \implies G_i(x) \leq 0, \quad i = 1, \dots, m.$$

- Complementarity formulation:

$$\begin{aligned} H_i(x) &= u_i - z_i, \\ 0 &\leq u_i \perp z_i \geq 0, \\ u_i G_i(x) &\leq 0, \quad i = 1, \dots, m. \end{aligned}$$



State-triggered inequality constraints

For related formulation see the exercise and [Ghezzi et al., 2026]

State-triggered constraints

If $H_i(x) > 0$, enforce $G_i(x) \leq 0$. Otherwise, $G_i(x)$ is unrestricted.

- ▶ Logical formulation:

$$H_i(x) > 0 \implies G_i(x) \leq 0, \quad i = 1, \dots, m.$$

- ▶ Complementarity formulation:

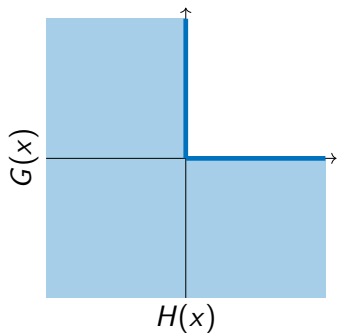
$$H_i(x) = u_i - z_i,$$

$$0 \leq u_i \perp z_i \geq 0,$$

$$u_i G_i(x) \leq 0, \quad i = 1, \dots, m.$$

- ▶ Interpretation:

- ▶ $H_i(x) > 0$ implies $u_i = H_i(x) > 0$, so $G_i(x) \leq 0$.
- ▶ $H_i(x) < 0$ implies $u_i = 0$ and $z_i = -H_i(x) > 0$, so $G_i(x)$ is unrestricted.
- ▶ $H_i(x) = 0$ implies $u_i = z_i = 0$, so $G_i(x)$ is unrestricted.



Summary on logical constraints

- ▶ Complementarity constraints can be used to model logically triggered equality or inequality constraints.
- ▶ Lead to connected feasible sets, hence they are nice for continuous optimization formulations.
- ▶ There exist equivalent (non)linear integer formulations for the same conditions.
- ▶ Do not use integer here, unless, you already have a binary variable, which takes the role of the indicator.
- ▶ Try to avoid mixing complementarity and integer optimization.

Summary on logical constraints

- ▶ Complementarity constraints can be used to model logically triggered equality or inequality constraints.
- ▶ Lead to connected feasible sets, hence they are nice for continuous optimization formulations.
- ▶ There exist equivalent (non)linear integer formulations for the same conditions.
- ▶ Do not use integer here, unless, you already have a binary variable, which takes the role of the indicator.
- ▶ Try to avoid mixing complementarity and integer optimization.

You will use what you have learned in:



`nomicc26/exercises/soft-docking`

Outline of the lecture

1. What are mathematical programs with complementarity constraints?
2. Modeling piecewise smooth functions
3. Advanced complementarity modeling
4. Applications of complementarity constraints
5. Summary

Piecewise smooth systems (PSS)

Regard **discontinuous** right-hand side, **piecewise smooth** on disjoint open regions $R_i \subset \mathbb{R}^{n_x}$

Discontinuous ODE

$$\dot{x} = f_i(x, u), \text{ if } x \in R_i, \quad i = 1, \dots, n_f$$

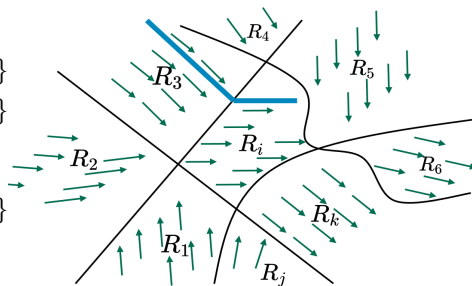
$$R_1 = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) > 0, \psi_2(x) > 0, \dots, \psi_{n_\psi}(x) > 0\}$$

$$R_2 = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) > 0, \psi_2(x) > 0, \dots, \psi_{n_\psi}(x) < 0\}$$

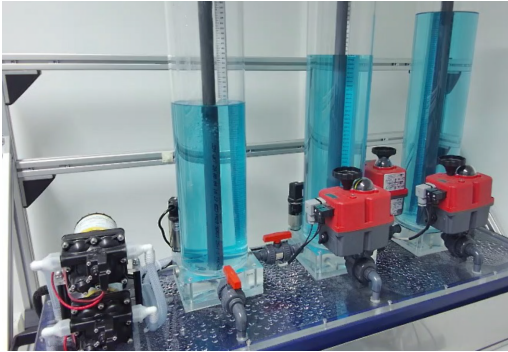
$\vdots$

$$R_{n_f} = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) < 0, \psi_2(x) < 0, \dots, \psi_{n_\psi}(x) < 0\}$$

- ▶ zero level sets of $\psi_i(x) = 0$ - region boundaries
- ▶ n_ψ smooth scalar switching functions define
 $n_f = 2^{n_\psi}$ nonempty regions



Examples of piecewise smooth systems

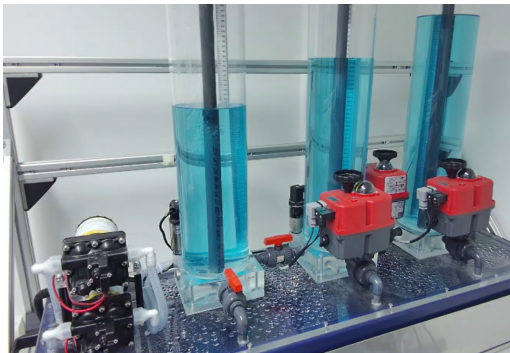


Tank systems: water can flow out from a specific level: discrete change in dynamics [Alsmeier et al., 2026].



Seasonal ice storage, phase changes of freezing and melting [Kollmar et al., 2026, in preparation].

Examples of piecewise smooth systems



Tank systems: water can flow out from a specific level: discrete change in dynamics [Alsmeier et al., 2026].



Seasonal ice storage, phase changes of freezing and melting [Kollmar et al., 2026, in preparation].

You can also optimally control and use MPC for PSS with nosnoc!

 github.com/nosnoc/nosnoc (🔥)

 github.com/nosnoc/nosnoc_py (🐍)

Filippov convexification for piecewise smooth systems

Aleksei F. Filippov (1923-2006) was a Russian mathematician.

The “structured” discontinuous right-hand side in PSS enables to define convex multipliers θ_i to define the convex set $F_F(x, u)$

Filippov Differential Inclusion

$$\dot{x} \in F_F(x, u) := \left\{ \sum_{i=1}^{n_f} f_i(x, u) \theta_i \mid \sum_{i=1}^{n_f} \theta_i = 1, \theta_i \geq 0, i = 1, \dots, n_f; \theta_i = 0, \text{ if } x \notin \overline{R_i} \right\}$$

Filippov convexification for piecewise smooth systems

Aleksei F. Filippov (1923-2006) was a Russian mathematician.

The “structured” discontinuous right-hand side in PSS enables to define convex multipliers θ_i to define the convex set $F_F(x, u)$

Filippov Differential Inclusion

$$\dot{x} \in F_F(x, u) := \left\{ \sum_{i=1}^{n_f} f_i(x, u) \theta_i \mid \sum_{i=1}^{n_f} \theta_i = 1, \theta_i \geq 0, i = 1, \dots, n_f; \theta_i = 0, \text{ if } x \notin \overline{R_i} \right\}$$

- ▶ for interior points $x \in R_i$ nothing changes: $F_F(x, u) = \{f_i(x, u)\}$
- ▶ Provides meaningful generalization on region boundaries
E.g. on $\overline{R_1} \cap \overline{R_2}$ both θ_1 and θ_2 can be nonzero

Stewart's representation

Introduced in [Stewart, 1990], used in [Nurkanović et al., 2024]

Assume sets R_i given by $R_i = \{x \in \mathbb{R}^{n_x} \mid g_i(x) < \min_{j \neq i} g_j(x)\}$

- ▶ How to obtain it from $R_i = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) > 0, \psi_2(x) > 0, \dots, \psi_{n_\psi}(x) > 0\}$?
- ▶ How to find the functions $g_i(x)$?

Stewart's representation

Introduced in [Stewart, 1990], used in [Nurkanović et al., 2024]

Assume sets R_i given by $R_i = \{x \in \mathbb{R}^{n_x} \mid g_i(x) < \min_{j \neq i} g_j(x)\}$

- ▶ How to obtain it from $R_i = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) > 0, \psi_2(x) > 0, \dots, \psi_{n_\psi}(x) > 0\}$?
- ▶ How to find the functions $g_i(x)$?

Definition of regions via switching functions

$$R_1 = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) > 0, \psi_2(x) > 0, \dots, \psi_{n_\psi}(x) > 0\}$$

$$R_2 = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) > 0, \psi_2(x) > 0, \dots, \psi_{n_\psi}(x) < 0\}$$

$$\vdots$$

$$R_{n_f} = \{x \in \mathbb{R}^{n_x} \mid \psi_1(x) < 0, \psi_2(x) < 0, \dots, \psi_{n_\psi}(x) < 0\}$$

$$\psi(x) := \begin{bmatrix} \psi_1(x) & \psi_2(x) & \dots & \psi_{n_\psi}(x) \end{bmatrix}^\top \in \mathbb{R}^{n_\psi}$$

$$g(x) = -S\psi(x)$$

Sign matrix

$$S = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \dots & -1 \end{bmatrix}$$

Definition via i -th row $S_{i,\bullet}$:

$$R_i = \{x \in \mathbb{R}^{n_x} \mid S_{i,\bullet}\psi(x) > 0\}$$

Filippov's convexification

Switched ODE not well-defined on region boundaries ∂R_i .

Replace ODE by differential inclusion, using convex combination of neighboring vector fields.

Filippov differential inclusion

$$\dot{x} \in F_F(x, u) := \left\{ \sum_{i=1}^{n_f} f_i(x, u) \theta_i \mid \sum_{i=1}^{n_f} \theta_i = 1, \theta_i \geq 0, i = 1, \dots, n_f, \theta_i = 0, \text{ if } x \notin R_i \cup \partial R_i \right\}$$

Filippov's convexification

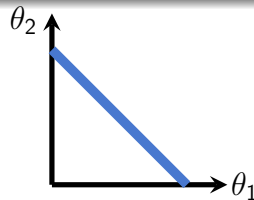
Switched ODE not well-defined on region boundaries ∂R_i .

Replace ODE by differential inclusion, using convex combination of neighboring vector fields.

Filippov differential inclusion

$$\dot{x} \in F_F(x, u) := \left\{ \sum_{i=1}^{n_f} f_i(x, u) \theta_i \mid \sum_{i=1}^{n_f} \theta_i = 1, \theta_i \geq 0, i = 1, \dots, n_f, \theta_i = 0, \text{ if } x \notin R_i \cup \partial R_i \right\}$$

- ▶ For interior points $x \in R_i$ nothing changes:
 $F_F(x, u) = \{f_i(x, u)\}$.
- ▶ Provides meaningful generalization on region boundaries.
E.g. on $\partial R_1 \cap \partial R_2$ both θ_1 and θ_2 can be nonzero.



The unit simplex.

From Filippov to dynamic complementarity systems

Using Stewart's reformulation [Stewart, 1990] and the KKT conditions of the parametric linear program.

$$R_i = \{x \in \mathbb{R}^n \mid g_i(x) < \min_{j \neq i} g_j(x)\}.$$

Linear programming representation

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

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

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

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

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

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

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

From Filippov to dynamic complementarity systems

Using Stewart's reformulation [Stewart, 1990] and the KKT conditions of the parametric linear program.

$$R_i = \{x \in \mathbb{R}^n \mid g_i(x) < \min_{j \neq i} g_j(x)\}.$$

Linear programming representation

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

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

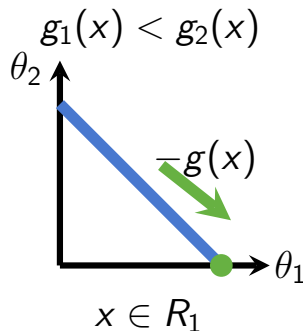
$$\text{s.t. } 0 \leq \tilde{\theta}$$

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

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

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

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



From Filippov to dynamic complementarity systems

Using Stewart's reformulation [Stewart, 1990] and the KKT conditions of the parametric linear program.

$$R_i = \{x \in \mathbb{R}^n \mid g_i(x) < \min_{j \neq i} g_j(x)\}.$$

Linear programming representation

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

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

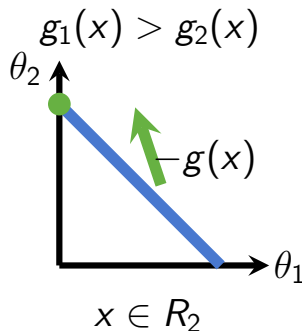
$$\text{s.t. } 0 \leq \tilde{\theta}$$

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

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

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

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



From Filippov to dynamic complementarity systems

Using Stewart's reformulation [Stewart, 1990] and the KKT conditions of the parametric linear program.

$$R_i = \{x \in \mathbb{R}^n \mid g_i(x) < \min_{j \neq i} g_j(x)\}.$$

Linear programming representation

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

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

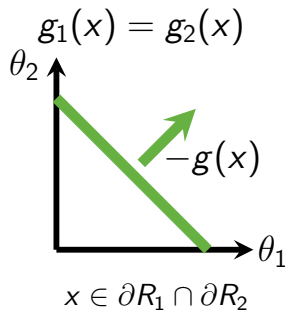
$$\text{s.t. } 0 \leq \tilde{\theta}$$

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

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

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

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



From Filippov to dynamic complementarity systems

Using Stewart's reformulation [Stewart, 1990] and the KKT conditions of the parametric linear program.

$$R_i = \{x \in \mathbb{R}^n \mid g_i(x) < \min_{j \neq i} g_j(x)\}.$$

Linear programming representation

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

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

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

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

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

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

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

Replace the LP by its optimality conditions.

Dynamic complementarity system (DCS)

$$\dot{x} = F(x, u) \theta \quad (4a)$$

$$0 = g(x) - \lambda - e\mu \quad (4b)$$

$$0 \leq \theta \perp \lambda \geq 0 \quad (4c)$$

$$1 = e^\top \theta \quad (4d)$$

- ▶ $\mu \in \mathbb{R}$ and $\lambda \in \mathbb{R}^{n_f}$ Lagrange multipliers.
- ▶ $(4c) \Leftrightarrow \min\{\theta, \lambda\} = 0 \in \mathbb{R}^{n_f}$
- ▶ Together, (4b)-(4d) determine the $(2n_f + 1)$ variables (θ, λ, μ) uniquely.

Controlled complementarity Lagrangian systems

Controlled CLS

$$\dot{q} = v$$

$$M(q)\dot{v} = f_v(q, v) + B_u(q)u$$

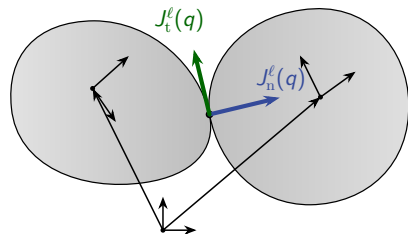
Controlled complementarity Lagrangian systems

Controlled CLS

$$\dot{q} = v$$

$$M(q)\dot{v} = f_v(q, v) + B_u(q)u + \sum_{\ell=1}^{n_c} (J_n^\ell(q)\lambda_n^\ell) \quad)$$

$$0 \leq \lambda_n^\ell \perp f_c^\ell(q) \geq 0, \quad \forall \ell \in \mathcal{C}$$



- ▶ $\mathcal{C} = \{1, \dots, n_c\}$ – index set of contact, $J_n^\ell(q)$ – contact normal, ϵ_r^ℓ – coeff. of restitution.
- ▶ **blue terms:** impact model $f_c^\ell(q) = 0$ becomes active, triggers state jump.

Controlled complementarity Lagrangian systems

Controlled CLS

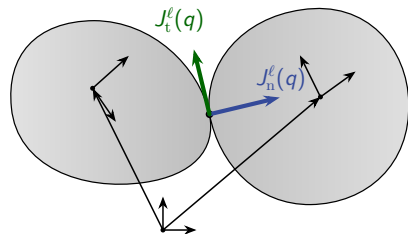
$$\dot{q} = v$$

$$M(q)\dot{v} = f_v(q, v) + B_u(q)u + \sum_{\ell=1}^{n_c} (J_n^\ell(q) \lambda_n^\ell) \quad)$$

$$0 \leq \lambda_n^\ell \perp f_c^\ell(q) \geq 0, \quad \forall \ell \in \mathcal{C}$$

if $(f_c^\ell(q(\tilde{s})) \leq 0$ then

$$J_n^\ell(q(t_s))^T v(t_s^+) \geq -\epsilon_r^\ell J_n^\ell(q(t_s))^T v(t_s^-), \quad \forall \ell \in \mathcal{C}$$



- ▶ $\mathcal{C} = \{1, \dots, n_c\}$ – index set of contact, $J_n^\ell(q)$ – contact normal, ϵ_r^ℓ – coeff. of restitution.
- ▶ **blue terms:** impact model $f_c^\ell(q) = 0$ becomes active, triggers state jump.

Controlled complementarity Lagrangian systems

Controlled CLS

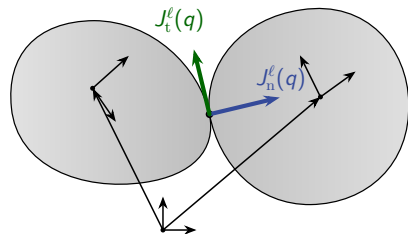
$$\dot{q} = v$$

$$M(q)\dot{v} = f_v(q, v) + B_u(q)u + \sum_{\ell=1}^{n_c} (J_n^\ell(q)\lambda_n^\ell) \quad)$$

$$0 \leq \lambda_n^\ell \perp f_c^\ell(q) \geq 0, \quad \forall \ell \in \mathcal{C}$$

if $f_c^\ell(q(\tilde{s})) \leq 0$ then

$$J_n^\ell(q(t_s))^\top v(t_s^+) \geq 0, \quad \forall \ell \in \mathcal{C}$$



- ▶ $\mathcal{C} = \{1, \dots, n_c\}$ – index set of contact, $J_n^\ell(q)$ – contact normal, ϵ_r^ℓ – coeff. of restitution.
- ▶ **blue terms:** impact model $f_c^\ell(q) = 0$ becomes active, triggers state jump.
- ▶ **green terms:** Coulomb's friction model (maximum dissipation principle).
- ▶ $J_t^\ell(q) \in \mathbb{R}^{n_q \times n_t}$, $n_t \in \{1, 2\}$ spans the tangent plane.

Controlled complementarity Lagrangian systems

Controlled CLS

$$\dot{q} = v$$

$$M(q)\dot{v} = f_v(q, v) + B_u(q)u + \sum_{\ell=1}^{n_c} (J_n^\ell(q)\lambda_n^\ell + J_t^\ell(q)\lambda_t^\ell)$$

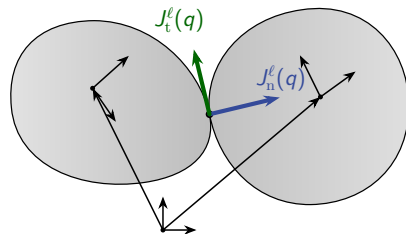
$$0 \leq \lambda_n^\ell \perp f_c^\ell(q) \geq 0, \quad \forall \ell \in \mathcal{C}$$

if $f_c^\ell(q(\tilde{s})) \leq 0$ then

$$J_n^\ell(q(t_s))^T v(t_s^+) \geq 0, \quad \forall \ell \in \mathcal{C}$$

$$\lambda_t^\ell \in \arg \max_{\tilde{\lambda}_t^\ell \in \mathbb{R}^{n_t}} -v^T J_t^\ell(q) \tilde{\lambda}_t^\ell$$

$$\text{s.t.} \quad \|\tilde{\lambda}_t^\ell\|_2 \leq \mu^\ell \lambda_n^\ell, \quad \forall \ell \in \mathcal{C}$$



- ▶ $\mathcal{C} = \{1, \dots, n_c\}$ – index set of contact, $J_n^\ell(q)$ – contact normal, ϵ_r^ℓ – coeff. of restitution.
- ▶ **blue terms:** impact model $f_c^\ell(q) = 0$ becomes active, triggers state jump.
- ▶ **green terms:** Coulomb's friction model (maximum dissipation principle).
- ▶ $J_t^\ell(q) \in \mathbb{R}^{n_q \times n_t}$, $n_t \in \{1, 2\}$ spans the tangent plane.

Controlled complementarity Lagrangian systems – single contact

Controlled CLS

$$\dot{q} = v$$

$$M(q)\dot{v} = f_v(q, v) + B_u(q)u + J_n(q)\lambda_n + J_t(q)\lambda_t$$

$$0 \leq \lambda_n \perp f_c(q) \geq 0$$

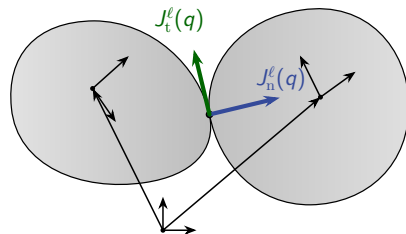
if $f_c(q(\tilde{s})) \leq 0$ then

$$J_n(q(t_s))^T v(t_s^+) \geq 0$$

$$\lambda_t \in \arg \max_{\tilde{\lambda}_t \in \mathbb{R}^{n_t}} -v^T J_t(q) \tilde{\lambda}_t$$

$$\text{s.t. } \|\tilde{\lambda}_t\|_2 \leq \mu \lambda_n$$

- ▶ $J_n(q)$ - contact normal
- ▶ **blue terms**: impact model $f_c(q) = 0$ becomes active, triggers state jump
- ▶ **green terms**: Coulomb's friction model (maximum dissipation principle)
- ▶ $J_t(q) \in \mathbb{R}^{n_q \times n_t}$, $n_t \in \{1, 2\}$ spans the tangent plane



The friction cone

$$M(q)\dot{v} = f_v(q, v) + B_u(q)u + J_n(q)\lambda_n + J_t(q)\lambda_t$$

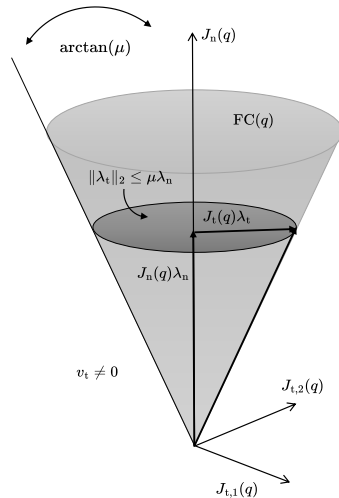
Solution map of friction model

$$\lambda_t \in \begin{cases} \{-\mu\lambda_n \frac{v_t}{\|v_t\|_2}\}, & \text{if } \|v_t\|_2 > 0 \end{cases}$$

- Set of all possible contact forces

$$FC(q) = \{J_n(q)\lambda_n + J_t(q)\lambda_t \mid \lambda_n \geq 0, \|\lambda_t\|_2 \leq \mu\lambda_n\}.$$

- Tangential velocity defined as $v_t := J_t(q)^T v$.
- In 2D solution map reduces to $\lambda_t \in -\mu\lambda_n \cdot \text{sign}(v_t)$



The friction cone

$$M(q)\dot{v} = f_v(q, v) + B_u(q)u + J_n(q)\lambda_n + J_t(q)\lambda_t$$

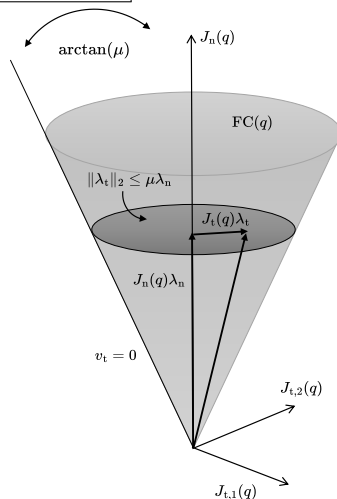
Solution map of friction model

$$\lambda_t \in \begin{cases} \{-\mu\lambda_n \frac{v_t}{\|v_t\|_2}\}, & \text{if } \|v_t\|_2 > 0 \\ \{\tilde{\lambda}_t \mid \|\tilde{\lambda}_t\|_2 \leq \mu\lambda_n\}, & \text{if } \|v_t\|_2 = 0 \end{cases}$$

- Set of all possible contact forces

$$FC(q) = \{J_n(q)\lambda_n + J_t(q)\lambda_t \mid \lambda_n \geq 0, \|\lambda_t\|_2 \leq \mu\lambda_n\}.$$

- Tangential velocity defined as $v_t := J_t(q)^T v$.
- In 2D solution map reduces to $\lambda_t \in -\mu\lambda_n \cdot \text{sign}(v_t)$



The friction cone

$$M(q)\dot{v} = f_v(q, v) + B_u(q)u + J_n(q)\lambda_n + J_t(q)\lambda_t$$

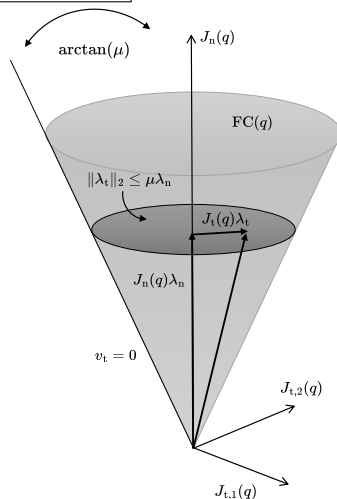
Solution map of friction model

$$\lambda_t \in \begin{cases} \{-\mu\lambda_n \frac{v_t}{\|v_t\|_2}\}, & \text{if } \|v_t\|_2 > 0 \\ \{\tilde{\lambda}_t \mid \|\tilde{\lambda}_t\|_2 \leq \mu\lambda_n\}, & \text{if } \|v_t\|_2 = 0 \end{cases}$$

- Set of all possible contact forces

$$FC(q) = \{J_n(q)\lambda_n + J_t(q)\lambda_t \mid \lambda_n \geq 0, \|\lambda_t\|_2 \leq \mu\lambda_n\}.$$

- Tangential velocity defined as $v_t := J_t(q)^T v$.
- In 2D solution map reduces to $\lambda_t \in -\mu\lambda_n \cdot \text{sign}(v_t)$



Exercise example:  [nomicc26/exercises/contacts-example](https://github.com/nomicc26/exercises/contacts-example)

MPCC modeling: key takeaways

- ▶ Complementarity constraints arise naturally in game theory, mathematical programming, economics, contact mechanics, hybrid systems, and many other fields.

MPCC modeling: key takeaways

- ▶ Complementarity constraints arise naturally in game theory, mathematical programming, economics, contact mechanics, hybrid systems, and many other fields.
- ▶ They can model *continuous* piecewise-smooth functions, set-valued extensions of discontinuous functions, logical relations, and sparsity measures such as ℓ_0 .

MPCC modeling: key takeaways

- ▶ Complementarity constraints arise naturally in game theory, mathematical programming, economics, contact mechanics, hybrid systems, and many other fields.
- ▶ They can model *continuous* piecewise-smooth functions, set-valued extensions of discontinuous functions, logical relations, and sparsity measures such as ℓ_0 .
- ▶ MPCCs require specialized theory and numerical methods. We will see tomorrow why.

MPCC modeling: key takeaways

- ▶ Complementarity constraints arise naturally in game theory, mathematical programming, economics, contact mechanics, hybrid systems, and many other fields.
- ▶ They can model *continuous* piecewise-smooth functions, set-valued extensions of discontinuous functions, logical relations, and sparsity measures such as ℓ_0 .
- ▶ MPCCs require specialized theory and numerical methods. We will see tomorrow why.
- ▶ Encoding integrality through complementarity is generally a bad idea. Encoding complementarity with integer variables is often safer, but may produce weak relaxations and slow MINLP solves.

MPCC modeling: key takeaways

- ▶ Complementarity constraints arise naturally in game theory, mathematical programming, economics, contact mechanics, hybrid systems, and many other fields.
- ▶ They can model *continuous* piecewise-smooth functions, set-valued extensions of discontinuous functions, logical relations, and sparsity measures such as ℓ_0 .
- ▶ MPCCs require specialized theory and numerical methods. We will see tomorrow why.
- ▶ Encoding integrality through complementarity is generally a bad idea. Encoding complementarity with integer variables is often safer, but may produce weak relaxations and slow MINLP solves.
- ▶ Avoid complementarity encodings of discrete decisions that create disconnected feasible sets. When possible, also avoid mixing integer and complementarity constraints in the same problem.

MPCC modeling: key takeaways

- ▶ Complementarity constraints arise naturally in game theory, mathematical programming, economics, contact mechanics, hybrid systems, and many other fields.
- ▶ They can model *continuous* piecewise-smooth functions, set-valued extensions of discontinuous functions, logical relations, and sparsity measures such as ℓ_0 .
- ▶ MPCCs require specialized theory and numerical methods. We will see tomorrow why.
- ▶ Encoding integrality through complementarity is generally a bad idea. Encoding complementarity with integer variables is often safer, but may produce weak relaxations and slow MINLP solves.
- ▶ Avoid complementarity encodings of discrete decisions that create disconnected feasible sets. When possible, also avoid mixing integer and complementarity constraints in the same problem.
- ▶ Logical conditions can often be expressed using either binary variables or complementarity constraints: **know your tools, your solver, and your application.**

Cited references I

-  [Achtziger, W. and Kanzow, C. \(2008\).](#)
Mathematical programs with vanishing constraints: optimality conditions and constraint qualifications.
Mathematical Programming, 114:69–99.
-  [Albrecht, S. and Ulbrich, M. \(2017\).](#)
Mathematical programs with complementarity constraints in the context of inverse optimal control for locomotion.
Optimization Methods and Software, 32(4):670–698.
-  [Alsmeier, H., Häusser, F., Knödler, A., Nurkanović, A., Pozharskiy, A., Diehl, M., and Findeisen, R. \(2026\).](#)
Real-time non-smooth mpc for switching systems: Application to a three-tank process.
In *2026 European Control Conference (ECC)*, pages 3352–3357.
-  [Baumrucker, B., Renfro, J. G., and Biegler, L. T. \(2008\).](#)
Mpec problem formulations and solution strategies with chemical engineering applications.
Computers & Chemical Engineering, 32(12):2903–2913.

Cited references II

-  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.
-  Biegler, L. T. (2010).
Nonlinear Programming.
MOS-SIAM Series on Optimization. SIAM.
-  Ding, C., Sun, D., and Ye, J. J. (2014).
First order optimality conditions for mathematical programs with semidefinite cone complementarity constraints.
Mathematical Programming, 147(1):539–579.
-  Facchinei, F. and Pang, J.-S. (2003).
Finite-dimensional variational inequalities and complementarity problems, volume 1-2.
Springer-Verlag.
-  Feng, M., Mitchell, J. J., Pang, J. S., Shen, X., and Wächter, A. (2018).
Complementarity formulations of ℓ_0 -norm optimization.
Pacific Journal of Optimization, 14(2):273–305.

Cited references III

-  Flegel, M. L. and Kanzow, C. (2005).
Abadie-type constraint qualification for mathematical programs with equilibrium constraints.
Journal of Optimization Theory and Applications, 124(3):595–614.
-  Fletcher, R. and Leyffer, S. (2004).
Solving mathematical programs with complementarity constraints as nonlinear programs.
Optimization Methods and Software, 19(1):15–40.
-  Ghezzi, A., Nurkanović, A., Weiss, A., Diehl, M., and Di Cairano, S. (2026).
A comparative study of minlp and mpvc formulations for solving complex nonlinear decision-making problems in aerospace applications.
Optimal Control Applications and Methods, 47(1):4–23.
-  Guo, L. and Ye, J. J. (2016).
Necessary optimality conditions for optimal control problems with equilibrium constraints.
SIAM Journal on Control and Optimization, 54(5):2710–2733.
-  Hegerhorst-Schultchen, L. C., Kirches, C., and Steinbach, M. C. (2020).
On the relation between mpecs and optimization problems in abs-normal form.
Optimization Methods and Software, 35(3):560–575.

Cited references IV



Hoheisel, T., Kanzow, C., and Schwartz, A. (2013).

Theoretical and numerical comparison of relaxation methods for mathematical programs with complementarity constraints.

Mathematical Programming, 137(1):257–288.



Hu, J., Mitchell, J. E., Pang, J.-S., and Yu, B. (2012).

On linear programs with linear complementarity constraints.

Journal of Global Optimization, 53:29–51.



Hu, X. and Ralph, D. (2002).

A note on sensitivity of value functions of mathematical programs with complementarity constraints.

Mathematical Programming, 93(2):265–279.



Kanzow, C. and Schwartz, A. (2015).

The price of inexactness: convergence properties of relaxation methods for mathematical programs with complementarity constraints revisited.

Mathematics of Operations Research, 40(2):253–275.

Cited references V

-  Kanzow, C., Schwartz, A., and Weiß, F. (2024).
The sparse (st) optimization problem: Reformulations, optimality, stationarity, and numerical results.
Computational Optimization and Applications, pages 1–36.
-  Kim, Y., Leyffer, S., and Munson, T. (2020).
Mpec methods for bilevel optimization problems.
In *Bilevel Optimization*, pages 335–360. Springer.
-  Krug, R., Mehrmann, V., and Schmidt, M. (2021).
Nonlinear optimization of district heating networks.
Optimization and Engineering, 22(2):783–819.
-  Liang, Y.-C., Xia, E.-H., Lin, G.-H., and Zhu, X. (2026).
Smoothing methods for mathematical programs with second-order cone complementarity constraints.
Journal of Global Optimization, pages 1–28.
-  Luo, Z., Pang, J., and Ralph, D. (1996).
Mathematical Programs with Equilibrium Constraints.
Cambridge University Press, Cambridge.

Cited references VI

-  Nurkanović, A. (2023).
Numerical Methods for Optimal Control of Nonsmooth Dynamical Systems.
PhD thesis, University of Freiburg.
-  Nurkanović, A., Pozharskiy, A., and Diehl, M. (2024).
Solving mathematical programs with complementarity constraints arising in nonsmooth optimal control.
Vietnam Journal of Mathematics, pages 1–39.
-  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.
-  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.
-  Raghunathan, A. and Biegler, L. (2003).
Mathematical programs with equilibrium constraints (MPECs) in process engineering.
Computers and Chemical Engineering, 27:1381–1392.

Cited references VII

-  Scheel, H. and Scholtes, S. (2000).
Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity.
Math. Oper. Res., 25:1–22.
-  Stewart, D. (1990).
A high accuracy method for solving odes with discontinuous right-hand side.
Numerische Mathematik, 58(1):299–328.
-  Szmuk, M., Reynolds, T. P., and Açıkmeşe, B. (2020).
Successive convexification for real-time six-degree-of-freedom powered descent guidance with state-triggered constraints.
Journal of Guidance, Control, and Dynamics, 43(8):1399–1413.
-  Vieira, A., Brogliato, B., and Prieur, C. (2019).
Quadratic optimal control of linear complementarity systems: First-order necessary conditions and numerical analysis.
IEEE Transactions on Automatic Control, 65(6):2743–2750.

Cited references VIII

-  Wensing, P. M., Posa, M., Hu, Y., Escande, A., Mansard, N., and Del Prete, A. (2023). Optimization-based control for dynamic legged robots.
IEEE Transactions on Robotics.
-  Ye, J. J. and Zhou, J. (2016). First-order optimality conditions for mathematical programs with second-order cone complementarity constraints.
SIAM Journal on Optimization, 26(4):2820–2846.

min, max, and the ℓ_1 -norm via binary variables

Let $M > 0$ be large enough such that all constraints below are feasible.

Maximum

The equality $t = \max_i F_i(x)$ is modeled by

$$F_i(x) \leq t \leq F_i(x) + M(1 - y_i), \quad i=1, \dots, m,$$

$$\sum_{i=1}^m y_i = 1, \quad y \in \{0, 1\}^m.$$

If $y_i = 1$, then $t = F_i(x)$.

Minimum

The equality $t = \min_i F_i(x)$ is modeled by

$$F_i(x) - M(1 - y_i) \leq t \leq F_i(x), \quad i=1, \dots, m,$$

$$\sum_{i=1}^m y_i = 1, \quad y \in \{0, 1\}^m.$$

If $y_i = 1$, then $t = F_i(x)$.

Exact ℓ_1 -norm

Introduce $\lambda_p, \lambda_n \in \mathbb{R}^m$ and $y \in \{0, 1\}^m$:

$$\begin{aligned} F_i(x) &= \lambda_{p,i} - \lambda_{n,i}, & i &= 1, \dots, m, \\ 0 &\leq \lambda_{p,i} \leq My_i, & i &= 1, \dots, m, \\ 0 &\leq \lambda_{n,i} \leq M(1 - y_i), & i &= 1, \dots, m, \\ t &= \sum_{i=1}^m (\lambda_{p,i} + \lambda_{n,i}). \end{aligned}$$

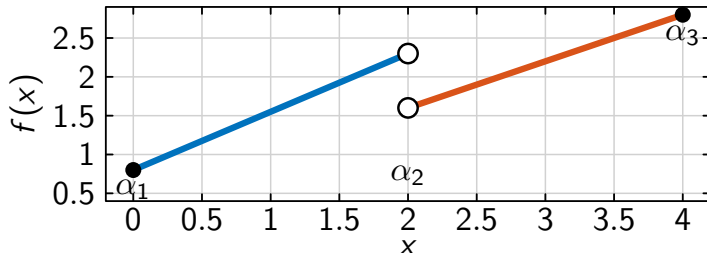
Hence $\lambda_{p,i}\lambda_{n,i} = 0$ and $t = \|F(x)\|_1 = \sum_{i=1}^m |F_i(x)|$.

Lower semi-continuous piecewise-linear functions

Complementarities cannot model discontinuous functions, integer constraints can

At a jump point α_2 , distinguish

$$f_-(\alpha_2), \quad f(\alpha_2), \quad f_+(\alpha_2).$$



$$x = \lambda_1 \alpha_1 + (\lambda_2 + \lambda_3 + \lambda_4) \alpha_2 + \lambda_5 \alpha_3,$$

$$f(x) = \lambda_1 f(\alpha_1) + \lambda_2 f_-(\alpha_2) + \lambda_3 f(\alpha_2) + \lambda_4 f_+(\alpha_2) + \lambda_5 f(\alpha_3),$$

$$\lambda_1 + \lambda_2 \leq y_1, \quad \lambda_3 \leq y_2, \quad \lambda_4 + \lambda_5 \leq y_3,$$

$$\lambda \geq 0, \quad \sum_{i=1}^5 \lambda_i = 1, \quad \sum_{i=1}^3 y_i = 1, \quad y \in \{0, 1\}^3.$$

We have indicator variables for sets: $[\alpha_1, \alpha_2)$, $\{\alpha_2\}$, $(\alpha_2, \alpha_3]$.