

Model Predictive Control and Reinforcement Learning

– Lecture 02: Numerical Optimization –

Armin Nurkanović

University of Freiburg

Slides jointly developed with Moritz Diehl, Andrea Ghezzi

School on Model Predictive Control and Reinforcement Learning
Freiburg, 7-16 October 2026

universität freiburg



NTNU

Norwegian University of
Science and Technology



Basic definitions

Some classifications of optimization problems

Optimality conditions for unconstrained optimization

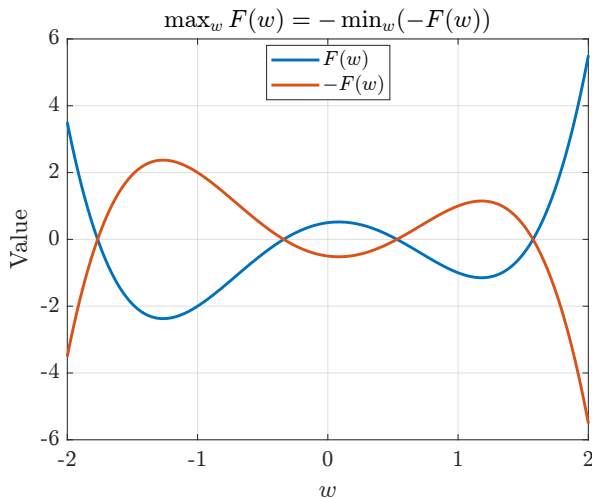
Newton's method for unconstrained optimization

Optimality conditions for constrained optimization

Optimization algorithms for constrained optimization

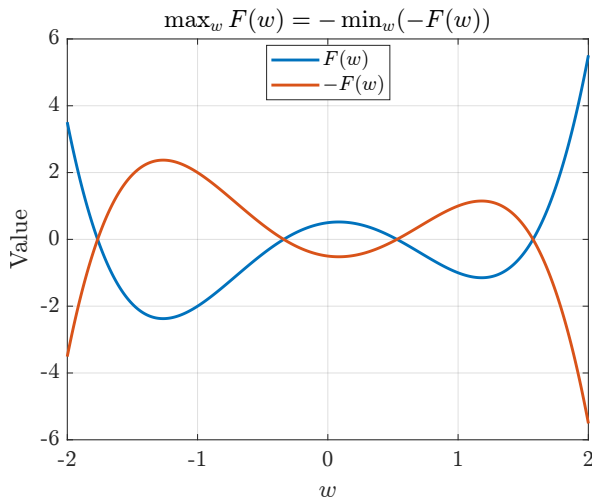
Minimize or maximize?

$$\max_{w \in \mathbb{R}^n} F(w) = - \min_{w \in \mathbb{R}^n} -F(w)$$



Minimize or maximize?

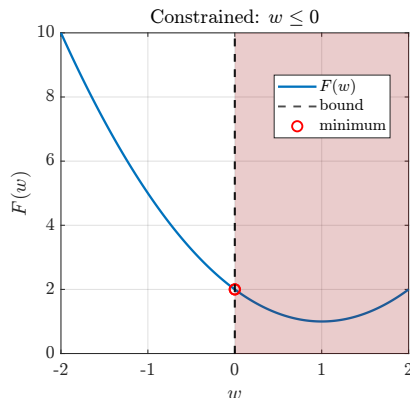
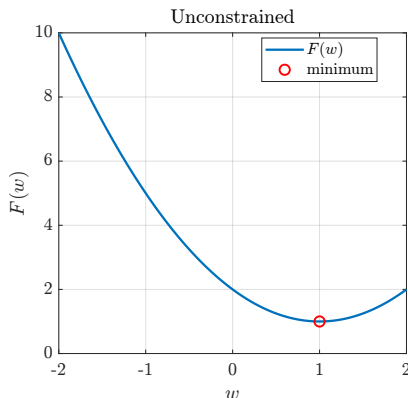
$$\max_{w \in \mathbb{R}^n} F(w) = -\min_{w \in \mathbb{R}^n} -F(w)$$



Without loss of generality – we will only minimize!

Constrained vs. unconstrained

- ▶ In unconstrained optimization we search in $\mathbb{R}^n$, $\nabla F(w) = 0$ is necessary for optimality.
- ▶ In constrained optimization we search in $\Omega \subset \mathbb{R}^n$, we need more elaborate necessary conditions.
- ▶ Constraints come from dynamics, actuator limits, model limits, design requirements, safety, ...



What is an optimization problem?

Minimize (or maximize) an objective function $F(w)$ depending on decision variables w subject to equality and/or inequality constraints

What is an optimization problem?

Minimize (or maximize) an objective function $F(w)$ depending on decision variables w subject to equality and/or inequality constraints

An optimization problem

$$\min_w F(w) \quad (1a)$$

$$\text{s.t. } G(w) = 0 \quad (1b)$$

$$H(w) \geq 0 \quad (1c)$$

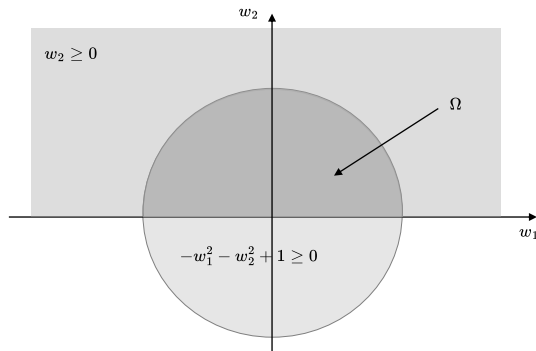
Terminology

- ▶ w : decision variable
- ▶ F : objective/cost function
- ▶ G, H : equality and inequality constraint functions

- ▶ Optimization is a powerful tool used in all quantitative sciences
- ▶ Only in few special cases a closed form solution exist
- ▶ Use an iterative algorithm to find solution
- ▶ The optimization problem may be parametric, and all functions depend on a fixed parameter p (e.g. MPC & RL)

Definition

The feasible set of the optimization problem (1) is defined as $\Omega = \{w \in \mathbb{R}^n \mid G(w) = 0, H(w) \geq 0\}$. A point $w \in \Omega$ is called a feasible point.

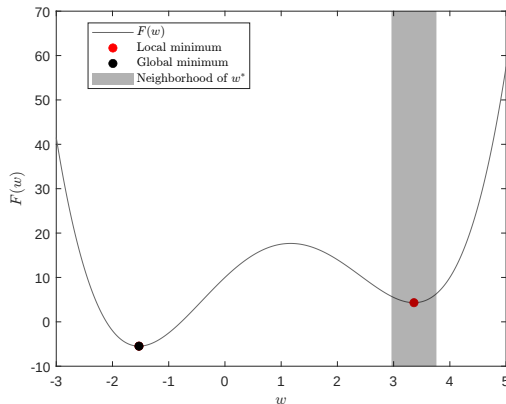


The feasible set is the intersection of the two grey areas (halfspace and circle)

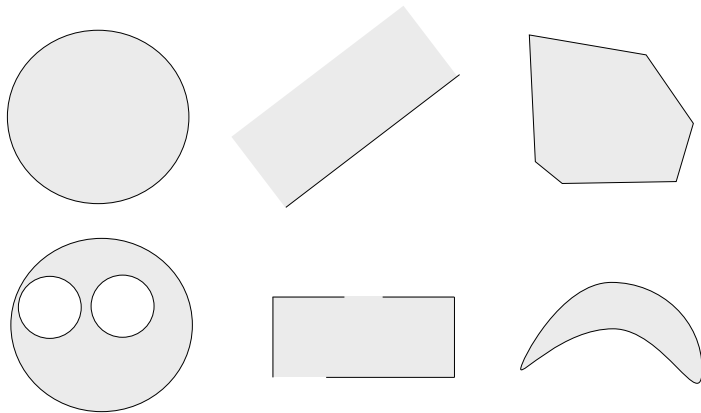
Definition

- ▶ A point $w^* \in \Omega$ is called a **local minimizer** of the NLP (1) if there exists an open ball $\mathcal{B}_\epsilon(w^*)$ with $\epsilon > 0$, such that for all $w \in \mathcal{B}_\epsilon(w^*) \cap \Omega$ it holds that $F(w) \geq F(w^*)$.
- ▶ A point $w^* \in \Omega$ is called a **global minimizer** of the NLP (1) if for all $w \in \Omega$ it holds that $F(w) \geq F(w^*)$.

The value $F(w^*)$ at a local/global minimizer w^* is called local/global minimum.



$$F(w) = \frac{1}{2}w^4 - 2w^3 - 3w^2 + 12w + 10$$



A set Ω is said to be convex if for any w_1, w_2 and any $\theta \in [0, 1]$ it holds $\theta w_1 + (1 - \theta)w_2 \in \Omega$

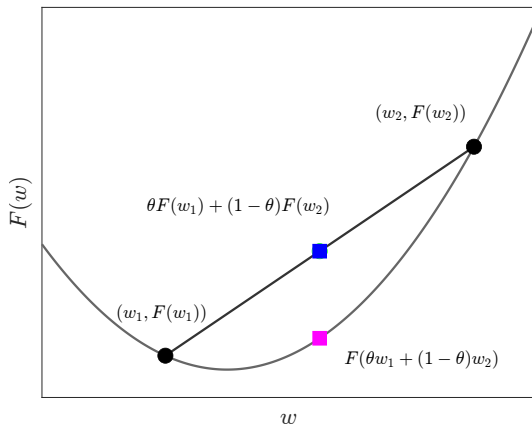
- ▶ A function F is convex if for every $w_1, w_2 \in \mathbb{R}^n$ and $\theta \in [0, 1]$ it holds that

$$F(\theta w_1 + (1 - \theta)w_2) \leq \theta F(w_1) + (1 - \theta)F(w_2)$$

- ▶ F is concave if and only if $-F$ is convex
- ▶ F is convex if and only if the epigraph

$$\text{epi}F = \{(w, t) \in \mathbb{R}^{n_w+1} \mid F(w) \leq t\}$$

is a convex set



A convex optimization problem

$$\begin{array}{ll}\min_w & F(w) \\ \text{s.t.} & G(w) = 0 \\ & H(w) \geq 0\end{array}$$

An optimization problem is convex if the objective function F is convex and the feasible set Ω is convex.

- ▶ Every locally optimal solution is global
- ▶ First-order conditions are necessary and sufficient (we come back to this)

A convex optimization problem

$$\begin{array}{ll}\min_w & F(w) \\ \text{s.t.} & G(w) = 0 \\ & H(w) \geq 0\end{array}$$

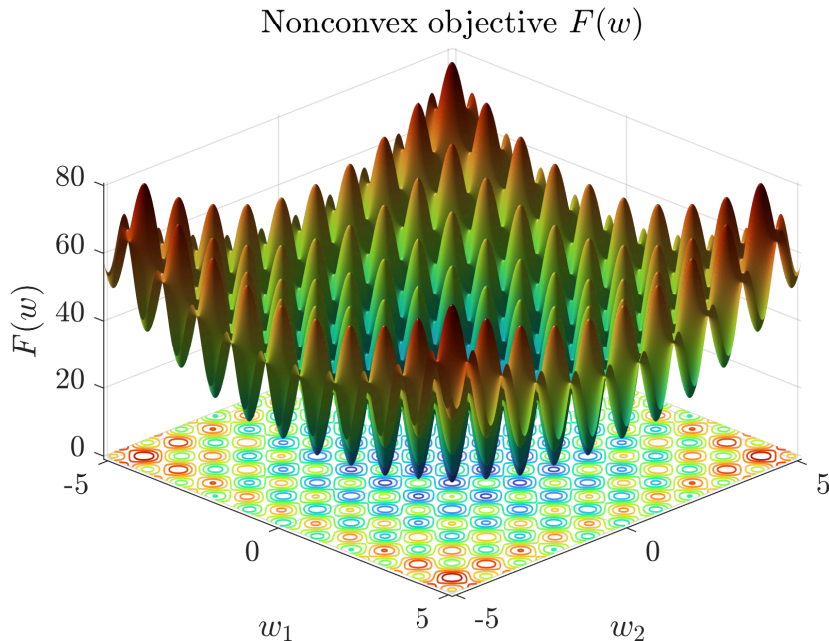
An optimization problem is convex if the objective function F is convex and the feasible set Ω is convex.

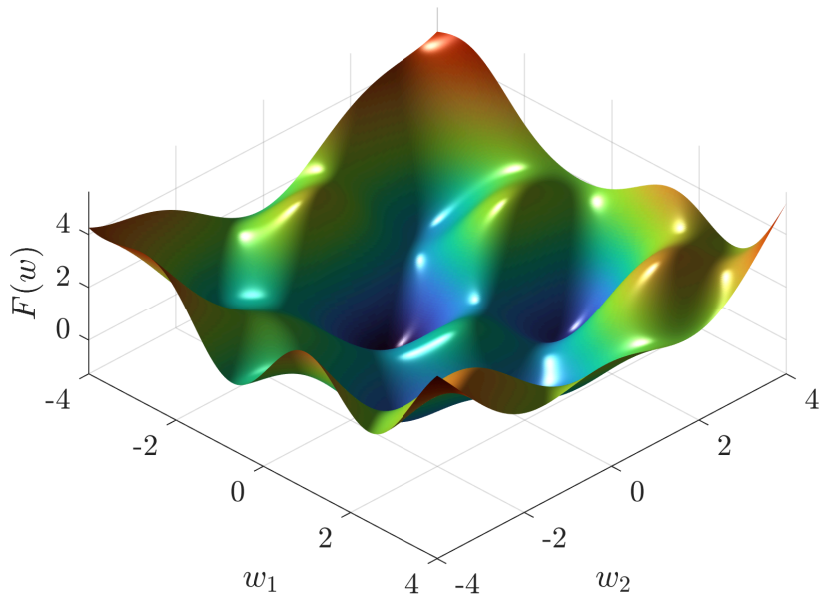
- ▶ Every locally optimal solution is global
- ▶ First-order conditions are necessary and sufficient (we come back to this)

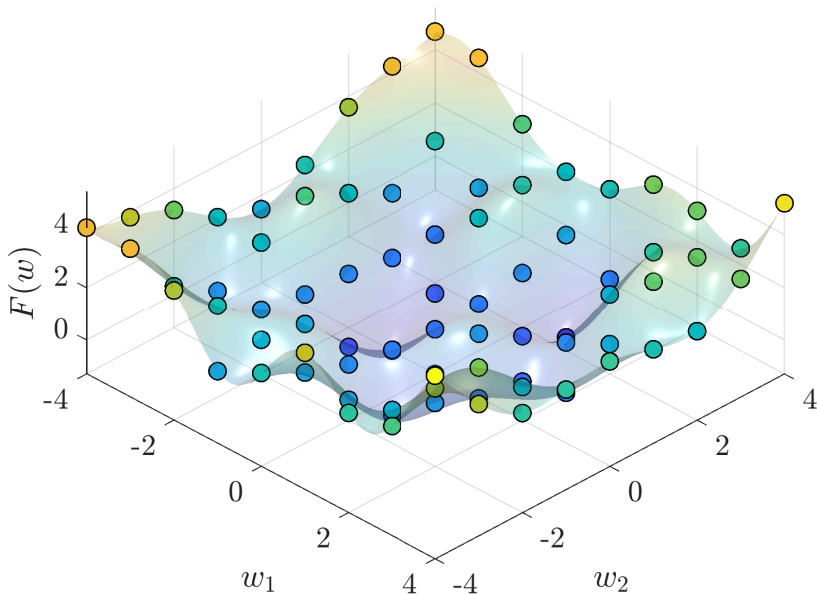
R. T. Rockafellar, SIAM Review, 1993

"...in fact, the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity."

Nonconvex problem with lots of minima









Basic definitions

Some classifications of optimization problems

Optimality conditions for unconstrained optimization

Newton's method for unconstrained optimization

Optimality conditions for constrained optimization

Optimization algorithms for constrained optimization

Optimization problems can be:

- ▶ unconstrained ($\Omega = \mathbb{R}^n$) or constrained ($\Omega \subset \mathbb{R}^n$)
- ▶ convex or nonconvex
- ▶ linear or nonlinear
- ▶ differentiable or nonsmooth
- ▶ continuous or (mixed-)integer
- ▶ finite or infinite dimensional

Optimization problems can be:

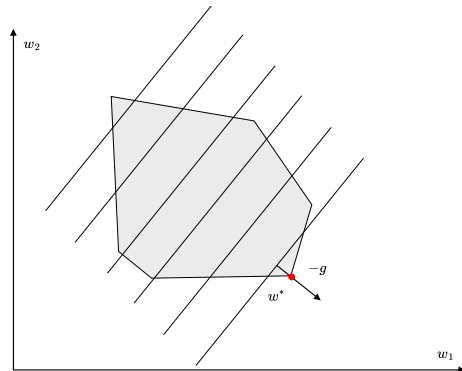
- ▶ unconstrained ($\Omega = \mathbb{R}^n$) or constrained ($\Omega \subset \mathbb{R}^n$)
- ▶ convex or nonconvex
- ▶ linear or nonlinear
- ▶ differentiable or nonsmooth
- ▶ continuous or (mixed-)integer
- ▶ finite or infinite dimensional

Yurii Nesterov, Lectures on Convex Optimization, 2018.

"... the main fact, which should be known to any person dealing with optimization models, is that in general, optimization problems are unsolvable."

Linear program

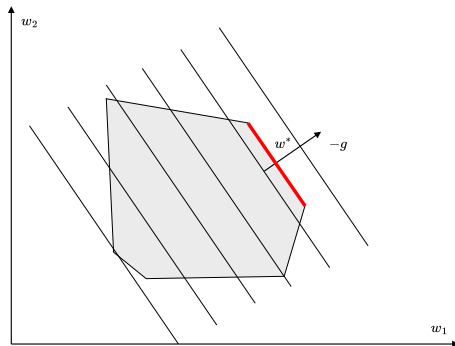
$$\begin{aligned} \min_w \quad & g^\top w \\ \text{s.t.} \quad & Aw - b = 0 \\ & Cw - d \geq 0 \end{aligned}$$



- ▶ Convex optimization problem
- ▶ 1947: simplex method by Dantzig, 1984: polynomial time interior-point method by Karmarkar
- ▶ If not unbounded, the solution is always at edge or vertex of the feasible set
- ▶ Today very mature and reliable

Linear program

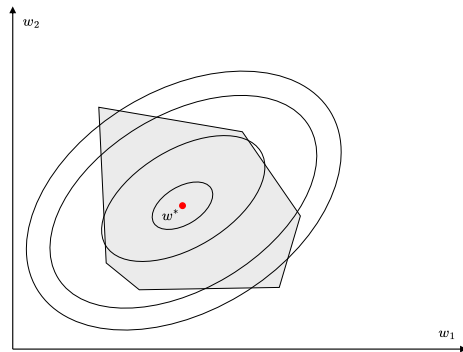
$$\begin{aligned} \min_w \quad & g^\top w \\ \text{s.t.} \quad & Aw - b = 0 \\ & Cw - d \geq 0 \end{aligned}$$



- ▶ Convex optimization problem
- ▶ 1947: simplex method by Dantzig, 1984: polynomial time interior-point method by Karmarkar
- ▶ If not unbounded, the solution is always at edge or vertex of the feasible set
- ▶ Today very mature and reliable

Quadratic program

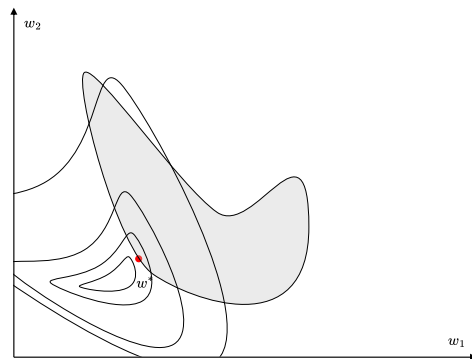
$$\begin{aligned} \min_w \quad & \frac{1}{2} w^\top Q w + g^\top w \\ \text{s.t.} \quad & A w - b = 0 \\ & C w - d \geq 0 \end{aligned}$$



- ▶ Depending on Q , can be convex and nonconvex
- ▶ Solved online in linear model predictive control
- ▶ Many good solvers: HPIPM, qpOASES, DAQP, Gurobi, OSQP, OOQP, quadprog, ...
- ▶ Subproblems in nonlinear optimization

Nonlinear programming problem

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$

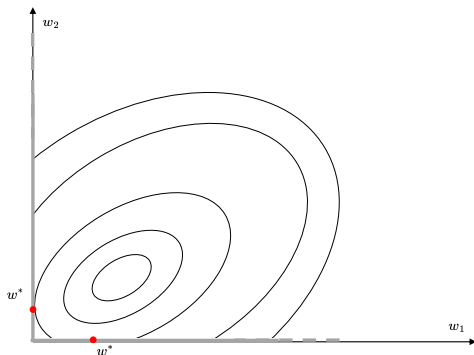


- ▶ Can be convex and nonconvex
- ▶ Solved with iterative Newton-type algorithms
- ▶ Solved in nonlinear model predictive control

MPCC

$$\begin{aligned} \min_{w_0, w_1, w_2} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \\ & 0 \leq w_1 \perp w_2 \geq 0 \end{aligned}$$

$$w = [w_0^\top, w_1^\top, w_2^\top]^\top, \quad w_1 \perp w_2 \Leftrightarrow w_1^\top w_2 = 0$$



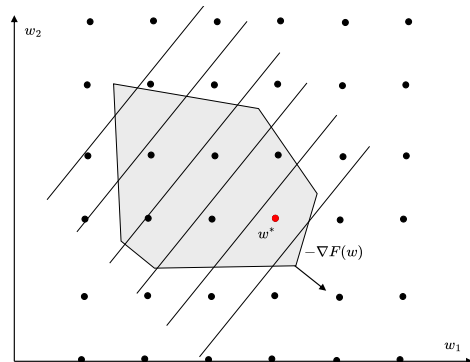
- ▶ More difficult than standard nonlinear programming since standard constraint qualifications fail to hold
- ▶ Feasible set is inherently nonsmooth and nonconvex
- ▶ Powerful modeling concept
- ▶ Requires specialized theory and algorithms, cf. [NOMICC26 summer school!](#)

MINLP

$$\begin{aligned} \min_{w_0 \in \mathbb{R}^p, w_1 \in \mathbb{Z}^q} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$

$$w = [w_0^\top, w_1^\top]^\top, n = p + q$$

- Combinatorial problem, feasible set is finite
- Branch and bound, brunch and cut methods
- Requires solution of many relaxed continuous convex or nonconvex problems, cf. [NOMICC26 summer school!](#)



Continuous-time Optimal Control Problem

$$\min_{x(\cdot), u(\cdot)} \int_0^T L_c(x(t), u(t)) dt + E(x(T))$$

$$\text{s.t. } x(0) = \bar{x}_0$$

$$\dot{x}(t) = f_c(x(t), u(t))$$

$$0 \geq h(x(t), u(t)), \quad t \in [0, T]$$

$$0 \geq r(x(T))$$

- ▶ Decision variables $x(\cdot)$ and $u(\cdot)$ in infinite dimensional function space
- ▶ Infinitely many constraints ($t \in [0, T]$)
- ▶ Smooth ordinary differential equation (ODE) $\dot{x}(t) = f_c(x(t), u(t))$
- ▶ More generally, dynamic models can be based on
 - ▶ Differential Algebraic Equations (DAE)
 - ▶ Partial Differential Equations (PDE)
 - ▶ Nonsmooth ODE
 - ▶ Stochastic ODE/PDE
- ▶ OCP can be convex or nonconvex
- ▶ All or some components of $u(t)$ may take integer values (mixed-integer OCP)



Continuous-time OCP

$$\min_{x(\cdot), u(\cdot)} \int_0^T L_c(x(t), u(t)) dt + E(x(T))$$

$$\text{s.t. } x(0) = \bar{x}_0$$

$$\dot{x}(t) = f_c(x(t), u(t))$$

$$0 \geq h(x(t), u(t)), \quad t \in [0, T]$$

$$0 \geq r(x(T))$$

Direct methods like direct collocation, multiple shooting. *First discretize, then optimize.*

Continuous-time OCP

$$\begin{aligned} \min_{x(\cdot), u(\cdot)} \quad & \int_0^T L_c(x(t), u(t)) dt + E(x(T)) \\ \text{s.t.} \quad & x(0) = \bar{x}_0 \\ & \dot{x}(t) = f_c(x(t), u(t)) \\ & 0 \geq h(x(t), u(t)), \quad t \in [0, T] \\ & 0 \geq r(x(T)) \end{aligned}$$

Discrete-time OCP (an NLP)

$$\begin{aligned} \min_{x, u} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k) + E(x_N) \\ \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{k+1} = f(x_k, u_k) \\ & 0 \geq h(x_k, u_k), \quad k = 0, \dots, N-1 \\ & 0 \geq r(x_N) \end{aligned}$$

Direct methods like direct collocation, multiple shooting. *First discretize, then optimize.*



Discrete time NMPC Problem (an NLP)

$$\begin{aligned} \min_{x,u} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k) + E(x_N) \\ \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{k+1} = f(x_k, u_k) \\ & 0 \geq h(x_k, u_k), \quad k = 0, \dots, N-1 \\ & 0 \geq r(x_N) \end{aligned}$$

Variables $x = (x_0, \dots, x_N)$ and $u = (u_0, \dots, u_{N-1})$ can be summarized in vector $w = (x, u) \in \mathbb{R}^n$.

Discrete time NMPC Problem (an NLP)

$$\begin{aligned} \min_{x,u} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k) + E(x_N) \\ \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{k+1} = f(x_k, u_k) \\ & 0 \geq h(x_k, u_k), \quad k = 0, \dots, N-1 \\ & 0 \geq r(x_N) \end{aligned}$$

Variables $x = (x_0, \dots, x_N)$ and $u = (u_0, \dots, u_{N-1})$ can be summarized in vector $w = (x, u) \in \mathbb{R}^n$.

Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$



Basic definitions

Some classifications of optimization problems

Optimality conditions for unconstrained optimization

Newton's method for unconstrained optimization

Optimality conditions for constrained optimization

Optimization algorithms for constrained optimization

Algebraic characterization of **unconstrained** local optima

Consider the unconstrained problem: $\min_{w \in \mathbb{R}^n} F(w)$

First-Order **Necessary** Condition of Optimality (FONC)

$$w^* \text{ local optimum} \Rightarrow \nabla F(w^*) = 0, w^* \text{ stationary point}$$

Second-Order **Necessary** Condition of Optimality (SONC)

$$w^* \text{ local optimum} \Rightarrow \nabla^2 F(w^*) \succeq 0$$

Algebraic characterization of **unconstrained** local optima

Consider the unconstrained problem: $\min_{w \in \mathbb{R}^n} F(w)$

First-Order **Necessary** Condition of Optimality (FONC)

$$w^* \text{ local optimum} \Rightarrow \nabla F(w^*) = 0, w^* \text{ stationary point}$$

Second-Order **Necessary** Condition of Optimality (SONC)

$$w^* \text{ local optimum} \Rightarrow \nabla^2 F(w^*) \succeq 0$$

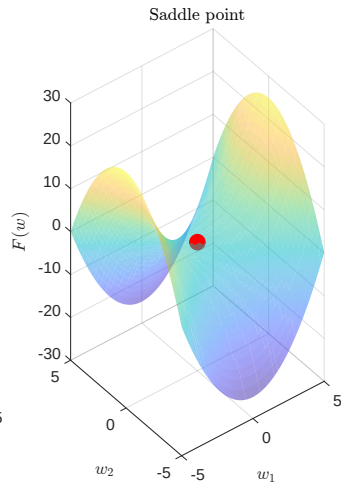
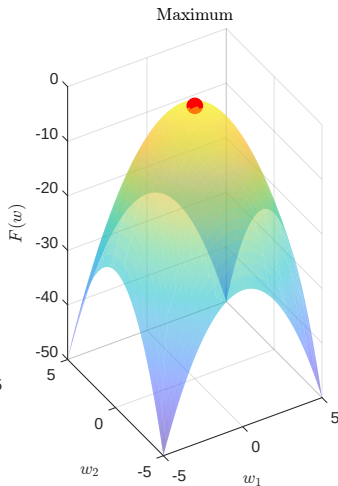
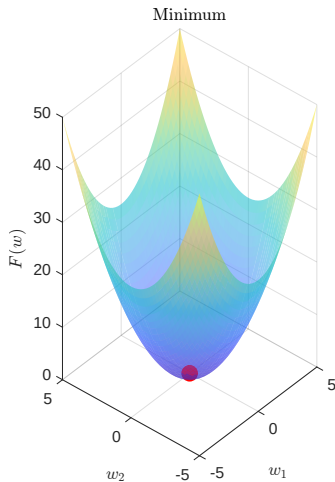
Second-Order **Sufficient** Conditions of Optimality (SOSC)

$$\nabla F(w^*) = 0 \text{ and } \nabla^2 F(w^*) \succ 0 \Rightarrow w^* \text{ strict local minimum}$$

$$\nabla F(w^*) = 0 \text{ and } \nabla^2 F(w^*) \prec 0 \Rightarrow w^* \text{ strict local maximum}$$

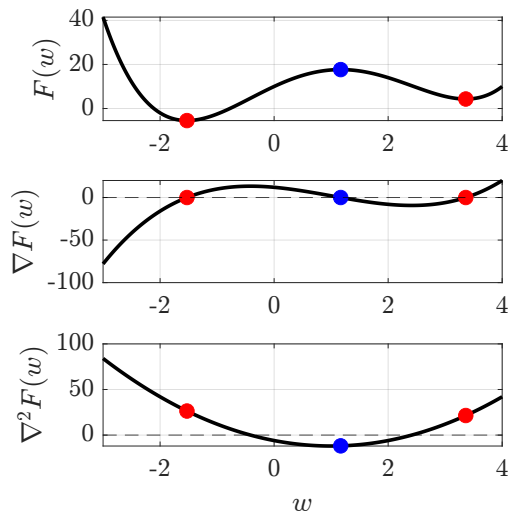
No conclusion can be drawn in the case $\nabla^2 F(w^*)$ is indefinite!

Type of stationary points

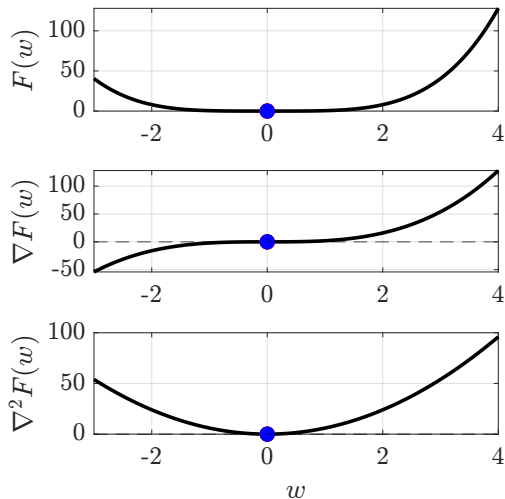


A stationary point can be a minimum, maximum or a saddle point

- Necessary conditions: find a candidate point (or to exclude points)
- Sufficient conditions: verify optimality of a candidate point



- ▶ Necessary conditions: find a candidate point (or to exclude points)
- ▶ Sufficient conditions: verify optimality of a candidate point
- ▶ A minimizer must satisfy SONC, but does not have to satisfy SOSC



Linearization of F at linearization point $\bar{w}$

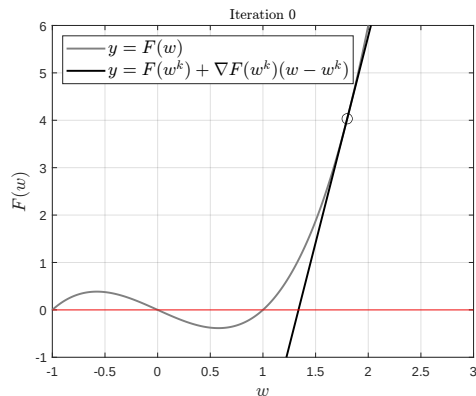
equals

First order Taylor series at $\bar{w}$

equals

$$F_L(w; \bar{w}) := F(\bar{w}) + \frac{\partial F}{\partial w}(\bar{w})(w - \bar{w})$$

(for continuously differentiable $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$)



Newton's method illustration for solving $F(w) = 0$

Linearization of F at linearization point $\bar{w}$

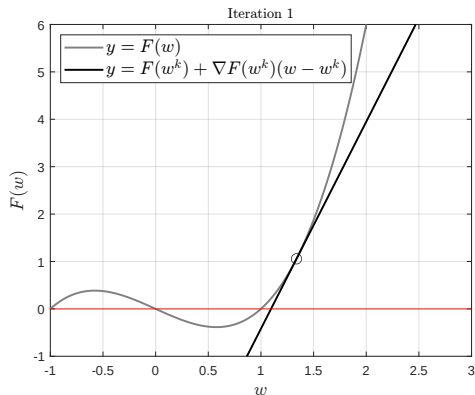
equals

First order Taylor series at $\bar{w}$

equals

$$F_L(w; \bar{w}) := F(\bar{w}) + \nabla_w F(\bar{w})^\top (w - \bar{w})$$

(for continuously differentiable $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$)



Newton's method illustration for solving $F(w) = 0$

Linearization of F at linearization point $\bar{w}$

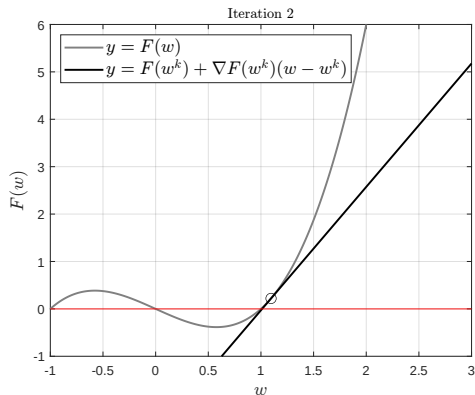
equals

First order Taylor series at $\bar{w}$

equals

$$F_L(w; \bar{w}) := F(\bar{w}) + \nabla_w F(\bar{w})^\top (w - \bar{w})$$

(for continuously differentiable $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$)



Newton's method illustration for solving $F(w) = 0$

Linearization of F at linearization point $\bar{w}$

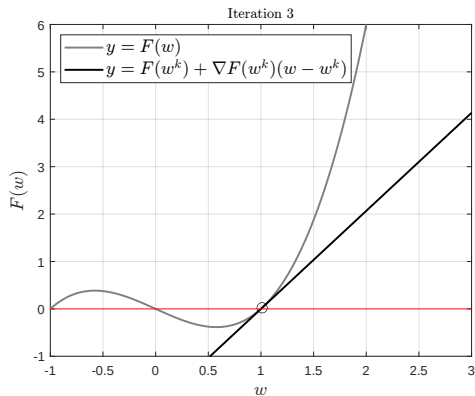
equals

First order Taylor series at $\bar{w}$

equals

$$F_L(w; \bar{w}) := F(\bar{w}) + \nabla_w F(\bar{w})^\top (w - \bar{w})$$

(for continuously differentiable $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$)



We want to solve $\min_{w \in \mathbb{R}^{n_w}} F(w)$ with $F : \mathbb{R}^{n_w} \rightarrow \mathbb{R}$ twice continuously differentiable.

Iterative Algorithm

An “iterative algorithm” generates a sequence $w_0, w_1, w_2, \dots$ of so called “iterates” with $w_k \rightarrow w^*$.

We look for stationary points therefore we regard the equation

We want to solve $\min_{w \in \mathbb{R}^{n_w}} F(w)$ with $F : \mathbb{R}^{n_w} \rightarrow \mathbb{R}$ twice continuously differentiable.

Iterative Algorithm

An “iterative algorithm” generates a sequence $w_0, w_1, w_2, \dots$ of so called “iterates” with $w_k \rightarrow w^*$.

We look for stationary points therefore we regard the equation

$$\nabla F(w^*) = 0.$$

Idea: linearize the nonlinear equation at w_k to compute $w_{k+1} = w_k + \Delta w$

$$\begin{aligned} \nabla F(w_k) + \underbrace{\frac{\partial}{\partial w}(\nabla F(w_k))}_{\nabla^2 F(w_k)} \Delta w &= 0 \\ -(\nabla^2 F(w_k))^{-1} \nabla F(w_k) &= \Delta w \end{aligned}$$

Δw is called the “Newton-step”, $\nabla^2 F(w_k)$ the Hessian.



The Newton's method can be obtained by a quadratic objective function, i.e. a second order Taylor approximation

Let m_k the quadratic model with objective F

$$\begin{aligned}m_k(w_k + \Delta w) &= F(w_k) + \nabla F(w_k)^\top \Delta w + \frac{1}{2} \Delta w^\top \nabla^2 F(w_k) \Delta w \\ &\approx F(w_k + \Delta w)\end{aligned}$$

We obtain Δw_k by

$$\Delta w_k = \arg \min_{\Delta w} m_k(w_k + \Delta w)$$

which translates to solving the equation

$$\nabla m(w_k + \Delta w) = \nabla F(w_k) + \nabla^2 F(w_k) \Delta w_k = 0$$

$$\Delta w_k = -(\nabla^2 F(w_k))^{-1} \nabla F(w_k)$$

Same formula but different interpretation!



Basic definitions

Some classifications of optimization problems

Optimality conditions for unconstrained optimization

Newton's method for unconstrained optimization

Optimality conditions for constrained optimization

Optimization algorithms for constrained optimization



Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \end{aligned}$$

Lagrangian function: $\mathcal{L}(w, \lambda) = F(w) - \lambda^\top G(w)$



Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \end{aligned}$$

Lagrangian function: $\mathcal{L}(w, \lambda) = F(w) - \lambda^\top G(w)$

Definition (LICQ)

A point w satisfies Linear Independence Constraint Qualification (**LICQ**) if

$\nabla G(w) := \frac{\partial G(w)}{\partial w}^\top$ is full column rank.

First-order necessary conditions for equality constrained optimization

Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \end{aligned}$$

Lagrangian function: $\mathcal{L}(w, \lambda) = F(w) - \lambda^\top G(w)$

Definition (LICQ)

A point w satisfies Linear Independence Constraint Qualification (**LICQ**) if $\nabla G(w) := \frac{\partial G(w)}{\partial w}^\top$ is full column rank.

First-order necessary conditions (in convex case also sufficient)

Let F, G in $\mathcal{C}^1$. If w^* is a (local) **minimizer**, and w^* satisfies **LICQ**, then there is a **unique vector** λ such that:

$$\nabla_w \mathcal{L}(w^*, \lambda^*) = \nabla F(w^*) - \nabla G(w^*) \lambda = 0$$

dual feasibility

$$\nabla_\lambda \mathcal{L}(w^*, \lambda^*) = G(w^*) = 0$$

primal feasibility

The Karush-Kuhn-Tucker (KKT) conditions

The “gradient equals zero” for constrained optimization.

Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$

$$\mathcal{L}(w, \lambda, \mu) = F(w) - \lambda^\top G(w) - \mu^\top H(w)$$

The Karush-Kuhn-Tucker (KKT) conditions

The “gradient equals zero” for constrained optimization.

Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$

$$\mathcal{L}(w, \lambda, \mu) = F(w) - \lambda^\top G(w) - \mu^\top H(w)$$

Definition (LICQ)

A point w satisfies LICQ if

$$[\nabla G(w), \quad \nabla H_{\mathcal{A}}(w)]$$

is full column rank.

Active set $\mathcal{A}(w) = \{i \mid H_i(w) = 0\}$

The Karush-Kuhn-Tucker (KKT) conditions

The “gradient equals zero” for constrained optimization.

Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$

$$\mathcal{L}(w, \lambda, \mu) = F(w) - \lambda^\top G(w) - \mu^\top H(w)$$

Definition (LICQ)

A point w satisfies LICQ if

$$[\nabla G(w), \quad \nabla H_{\mathcal{A}}(w)]$$

is full column rank.

Active set $\mathcal{A}(w) = \{i \mid H_i(w) = 0\}$

Theorem (KKT conditions - FONC for constrained optimization)

Let F, G, H be $\mathcal{C}^1$. If w^* is a (local) *minimizer and satisfies LICQ*, then there are *unique vectors λ^* and μ^* such that (w^*, λ^*, μ^*) satisfies:*

$$\nabla_w \mathcal{L}(w^*, \lambda^*, \mu^*) = 0, \quad \mu^* \geq 0,$$

$$G(w^*) = 0, \quad H(w^*) \geq 0$$

$$\mu_i^* H_i(w^*) = 0, \quad \forall i$$

dual feasibility

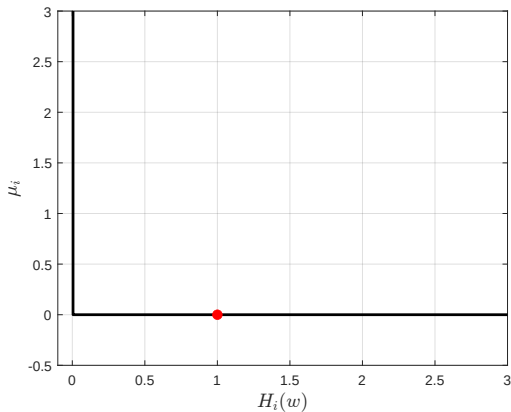
primal feasibility

complementary slackness

The complementary slackness condition



- ▶ complementarity conditions
 $0 \leq \mu^* \perp H(w^*) \geq 0$
- ▶ same as $\min(\mu^*, H(w^*)) = 0$
- ▶ zero level set of min is an L-shaped set

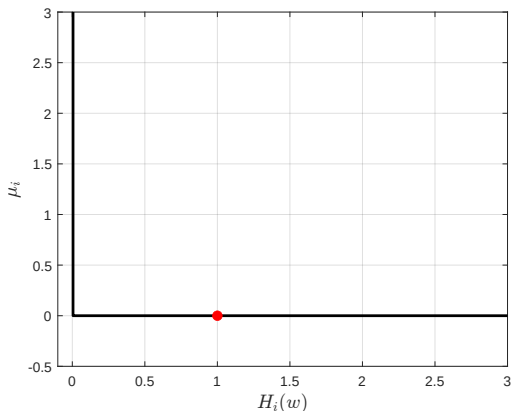


The complementary slackness condition

- ▶ complementarity conditions
 $0 \leq \mu^* \perp H(w^*) \geq 0$
- ▶ same as $\min(\mu^*, H(w^*)) = 0$
- ▶ zero level set of $\min$ is an L-shaped set

Cases:

- ▶ $H_i(w^*) > 0$ then $\mu_i^* = 0$, and $H_i(w)$ is inactive

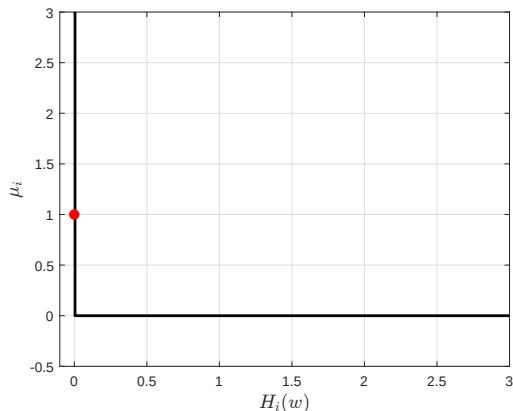


The complementary slackness condition

- ▶ complementarity conditions
 $0 \leq \mu^* \perp H(w^*) \geq 0$
- ▶ same as $\min(\mu^*, H(w^*)) = 0$
- ▶ zero level set of $\min$ is an L-shaped set

Cases:

- ▶ $H_i(w^*) > 0$ then $\mu_i^* = 0$, and $H_i(w)$ is **inactive**
- ▶ $\mu_i^* > 0$ and $H_i(w) = 0$ then $H_i(w)$ is **strictly active**

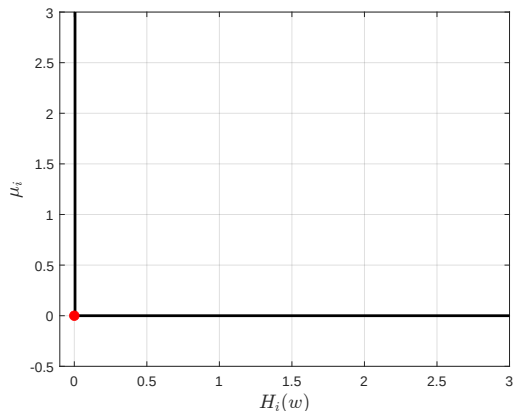


The complementary slackness condition

- ▶ complementarity conditions
 $0 \leq \mu^* \perp H(w^*) \geq 0$
- ▶ same as $\min(\mu^*, H(w^*)) = 0$
- ▶ zero level set of $\min$ is an L-shaped set

Cases:

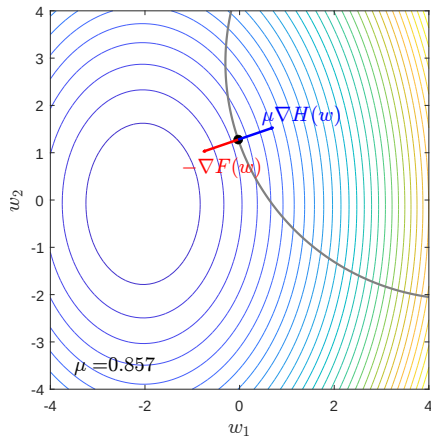
- ▶ $H_i(w^*) > 0$ then $\mu_i^* = 0$, and $H_i(w)$ is **inactive**
- ▶ $\mu_i^* > 0$ and $H_i(w) = 0$ then $H_i(w)$ is **strictly active**
- ▶ $\mu_i^* = 0$ and $H_i(w) = 0$ then $H_i(w)$ is **weakly active**



Some intuition for the KKT conditions



$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$



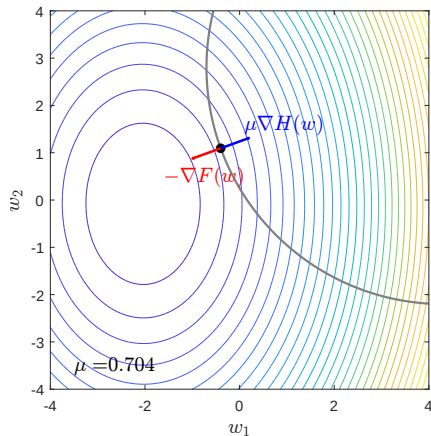
Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

Some intuition for the KKT conditions



$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

► $-\nabla F$ is the gravity



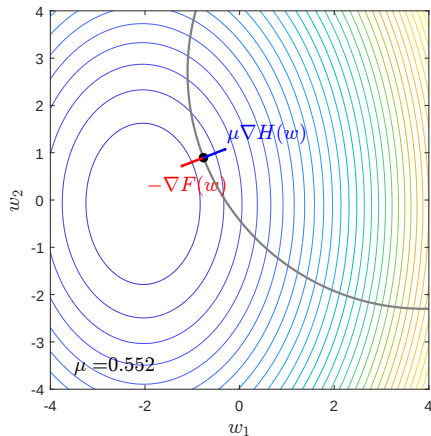
Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

Some intuition for the KKT conditions



$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

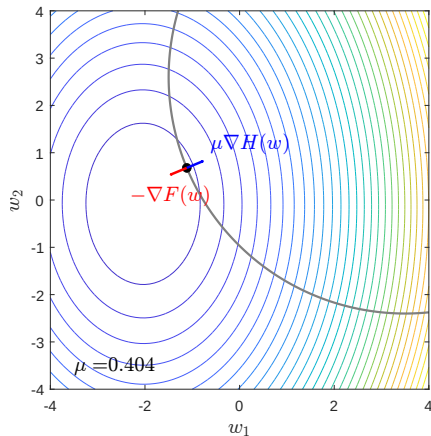
► $-\nabla F$ is the gravity



Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

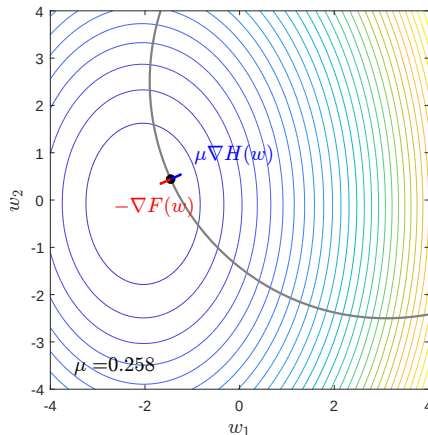
- ▶ $-\nabla F$ is the gravity
- ▶ $\mu \nabla H$ is the force of the fence. Sign $\mu \geq 0$ means the fence can only "push" the ball



Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

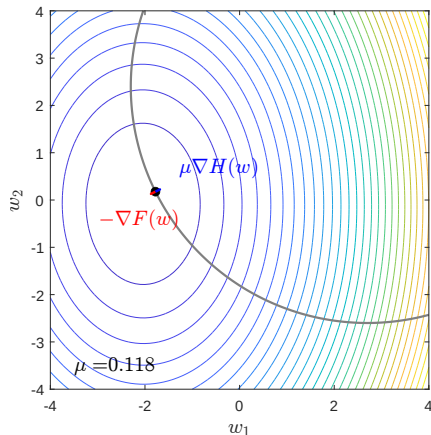
- ▶ $-\nabla F$ is the gravity
- ▶ $\mu \nabla H$ is the force of the fence. Sign $\mu \geq 0$ means the fence can only "push" the ball
- ▶ ∇H gives the direction of the force and μ adjusts the magnitude.



Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

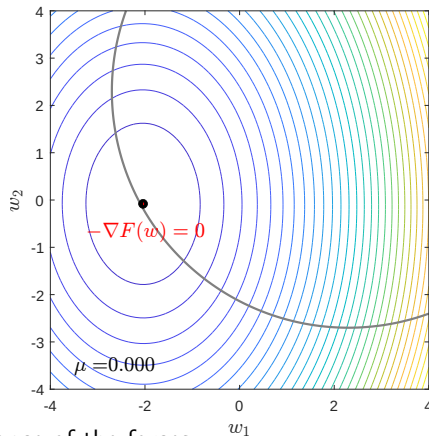
- ▶ $-\nabla F$ is the gravity
- ▶ $\mu \nabla H$ is the force of the fence. Sign $\mu \geq 0$ means the fence can only "push" the ball
- ▶ ∇H gives the direction of the force and μ adjusts the magnitude.
- ▶ active constraint: $H(w) = 0, \mu > 0$



Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

- ▶ $-\nabla F$ is the gravity
- ▶ $\mu \nabla H$ is the force of the fence. Sign $\mu \geq 0$ means the fence can only "push" the ball
- ▶ ∇H gives the direction of the force and μ adjusts the magnitude.
- ▶ active constraint: $H(w) = 0, \mu > 0$
- ▶ weakly active constraint:
 $H(w) = 0, \mu = 0$ the ball touches the fence but no force is needed



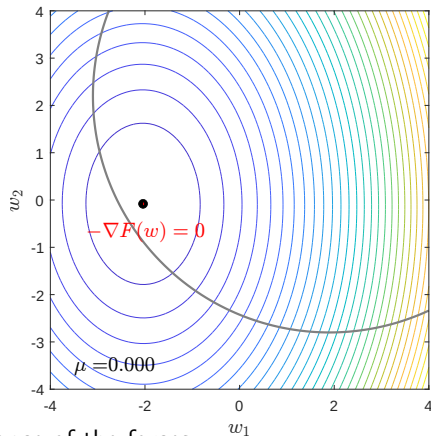
Balance of the forces:

$$\nabla \mathcal{L}(w, \mu) = \nabla F(w) - \nabla H(w) \mu = 0$$

Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

- ▶ $-\nabla F$ is the gravity
- ▶ $\mu \nabla H$ is the force of the fence. Sign $\mu \geq 0$ means the fence can only "push" the ball
- ▶ ∇H gives the direction of the force and μ adjusts the magnitude.
- ▶ active constraint: $H(w) = 0, \mu > 0$
- ▶ weakly active constraint:
 $H(w) = 0, \mu = 0$ the ball touches the fence but no force is needed
- ▶ inactive constraint: $H(w) > 0, \mu = 0$



Balance of the forces:

$$\nabla \mathcal{L}(w, \mu) = \nabla F(w) - \nabla H(w) \mu = 0$$

Animation inspired by Lecture 2 of the Winter School on Numerical Optimal Control with Differential Algebraic Equations by S. Gros and M. Diehl, Freiburg, 2016



Basic definitions

Some classifications of optimization problems

Optimality conditions for unconstrained optimization

Newton's method for unconstrained optimization

Optimality conditions for constrained optimization

Optimization algorithms for constrained optimization

In direct methods, we have to solve the discretized optimal control problem, which is a Nonlinear Program (NLP)

General Nonlinear Program (NLP)

$$\min_{w \in \mathbb{R}^n} F(w) \quad \text{s.t.} \quad \begin{cases} G(w) &= 0 \\ H(w) &\geq 0 \end{cases}$$

We first treat the case without inequality constraints

NLP only with equality constraints

$$\min_{w \in \mathbb{R}^n} F(w) \quad \text{s.t.} \quad G(w) = 0$$

Lagrange function

$$\mathcal{L}(w, \lambda) = F(w) - \lambda^\top G(w)$$

For an optimal solution w^* there exist multipliers λ^* such that

Nonlinear root-finding problem

$$\begin{aligned}\nabla_w \mathcal{L}(w^*, \lambda^*) &= 0 \\ G(w^*) &= 0\end{aligned}$$

Use Newton's method to solve:

$$\begin{aligned}\nabla_w \mathcal{L}(w^*, \lambda^*) &= 0 \\ G(w^*) &= 0 \quad ?\end{aligned}$$

Given an iterate (w^k, λ^k) , the linearization reads as:

$$\begin{aligned}\nabla_w \mathcal{L}(w^k, \lambda^k) + \nabla_w^2 \mathcal{L}(w^k, \lambda^k) \Delta w - \nabla_w G(w^k) \Delta \lambda &= 0 \\ G(w^k) + \nabla_w G(w^k)^\top \Delta w &= 0\end{aligned}$$

Due to $\nabla \mathcal{L}(w^k, \lambda^k) = \nabla F(w^k) - \nabla G(w^k) \lambda^k$, this is equivalent to:

$$\begin{aligned}\nabla_w F(w^k) + \nabla_w^2 \mathcal{L}(w^k, \lambda^k) \Delta w - \nabla_w G(w^k) \lambda^+ &= 0 \\ G(w^k) + \nabla_w G(w^k)^\top \Delta w &= 0\end{aligned}$$

with the shorthand $\lambda^+ = \lambda^k + \Delta \lambda$

Conditions

$$\begin{array}{rcl} \nabla_w F(w^k) & + \nabla_w^2 \mathcal{L}(w^k, \lambda^k) \Delta w & - \nabla_w G(w^k) \lambda^+ = 0 \\ G(w^k) & + \nabla_w G(w^k)^\top \Delta w & = 0 \end{array}$$

are the KKT optimality conditions of a quadratic program (QP), namely:

Quadratic program

$$\begin{array}{ll} \min_{\Delta w \in \mathbb{R}^n} & \nabla F(w^k)^\top \Delta w + \frac{1}{2} \Delta w^\top A^k \Delta w \\ \text{s.t.} & G(w^k) + \nabla G(w^k)^\top \Delta w = 0, \end{array}$$

with $A^k = \nabla_w^2 \mathcal{L}(w^k, \lambda^k)$

The full step Newton's Method iterates by solving in each iteration the QP

Quadratic program in Sequential Quadratic Programming (SQP)

$$\begin{aligned} \min_{\Delta w \in \mathbb{R}^n} \quad & \nabla F(w^k)^\top \Delta w + \frac{1}{2} \Delta w^\top A^k \Delta w \\ \text{s.t.} \quad & G(w^k) + \nabla G(w^k)^\top \Delta w = 0, \end{aligned}$$

with $A^k = \nabla_w^2 \mathcal{L}(w^k, \lambda^k)$

This obtains as solution the step Δw^k and the new multiplier $\lambda_{QP}^+ = \lambda^k + \Delta \lambda^k$

New iterate

$$\begin{aligned} w^{k+1} &= w^k + \Delta w^k \\ \lambda^{k+1} &= \lambda^k + \Delta \lambda^k = \lambda_{QP}^+ \end{aligned}$$

This is the “full step, exact Hessian SQP method for equality constrained optimization”.

Regard again NLP with both, equality and inequality constraints:

NLP with equality and inequality constraints

$$\min_{w \in \mathbb{R}^n} F(w) \quad \text{s.t.} \quad \begin{cases} G(w) = 0 \\ H(w) \geq 0 \end{cases}$$

Lagrangian function for NLP with equality and inequality constraints

$$\mathcal{L}(w, \lambda, \mu) = F(w) - \lambda^\top G(w) - \mu^\top H(w)$$

Theorem (Karush-Kuhn-Tucker (KKT) conditions)

Let F, G, H be $\mathcal{C}^1$. If w^* is a (local) minimizer and satisfies LICQ, then there are unique vectors λ^* and μ^* such that (w^*, λ^*, μ^*) satisfies:

$$\nabla_w \mathcal{L}(w^*, \lambda^*, \mu^*) = 0$$

$$G(w^*) = 0$$

$$H(w^*) \geq 0$$

$$\mu^* \geq 0$$

$$H(w^*)^\top \mu^* = 0$$

- ▶ Last three *complementarity conditions* make the KKT conditions nonsmooth
- ▶ This system cannot be solved by plain Newton's method. But we can use SQP...

By Linearizing all functions within the KKT Conditions, and setting $\lambda^+ = \lambda^k + \Delta\lambda$ and $\mu^+ = \mu^k + \Delta\mu$, we obtain the KKT conditions of a Quadratic Program (QP)

QP with inequality constraints

$$\begin{aligned} \min_{\Delta w \in \mathbb{R}^n} \quad & \nabla F(w^k)^\top \Delta w + \frac{1}{2} \Delta w^\top A^k \Delta w \\ \text{s.t.} \quad & G(w^k) + \nabla G(w^k)^\top \Delta w = 0 \\ & H(w^k) + \nabla H(w^k)^\top \Delta w \geq 0 \end{aligned}$$

with $A^k = \nabla_w^2 \mathcal{L}(w^k, \lambda^k, \mu^k)$

By Linearizing all functions within the KKT Conditions, and setting $\lambda^+ = \lambda^k + \Delta\lambda$ and $\mu^+ = \mu^k + \Delta\mu$, we obtain the KKT conditions of a Quadratic Program (QP)

QP with inequality constraints

$$\begin{aligned} \min_{\Delta w \in \mathbb{R}^n} \quad & \nabla F(w^k)^\top \Delta w + \frac{1}{2} \Delta w^\top A^k \Delta w \\ \text{s.t.} \quad & G(w^k) + \nabla G(w^k)^\top \Delta w = 0 \\ & H(w^k) + \nabla H(w^k)^\top \Delta w \geq 0 \end{aligned}$$

with $A^k = \nabla_w^2 \mathcal{L}(w^k, \lambda^k, \mu^k)$

- ▶ QP solution: $\Delta w^k, \lambda_{\text{QP}}^+, \mu_{\text{QP}}^+$
- ▶ full step: $w^{k+1} = w^k + \Delta w^k, \lambda^{k+1} = \lambda_{\text{QP}}^+, \mu^{k+1} = \mu_{\text{QP}}^+$
- ▶ nonsmooth complementarity conditions resolved at QP level

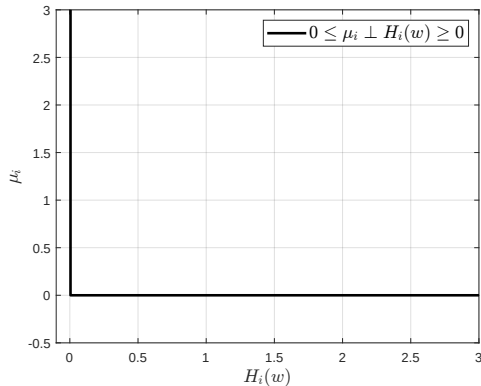
NLP with inequalities

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

KKT conditions

$$\begin{aligned} \nabla F(w) - \nabla H(w)\mu &= 0 \\ 0 \leq \mu \perp H(w) &\geq 0 \end{aligned}$$

- Main difficulty: nonsmoothness of complementarity conditions



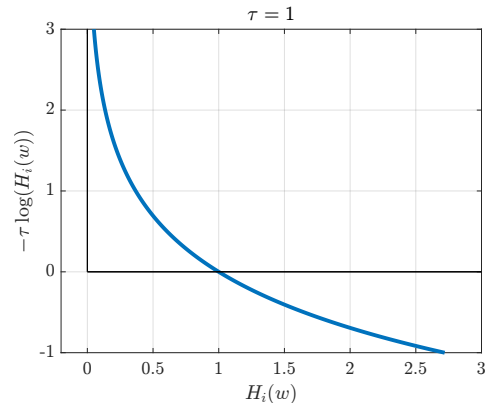
NLP with inequalities

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

Idea: put inequality constraint into objective

Barrier problem

$$\min_{w \in \mathbb{R}^n} F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$



$\tau \log(H_i(w))$ approximates:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

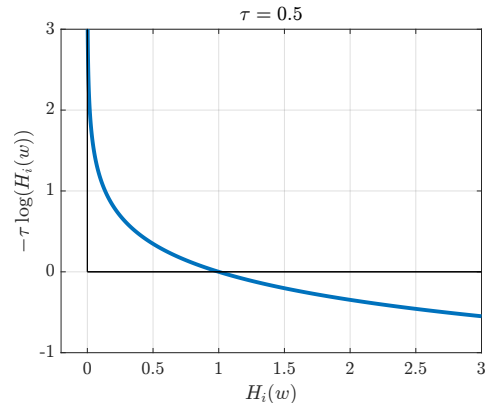
NLP with inequalities

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

Idea: put inequality constraint into objective

Barrier problem

$$\min_{w \in \mathbb{R}^n} F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$



$\tau \log(H_i(w))$ approximates:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

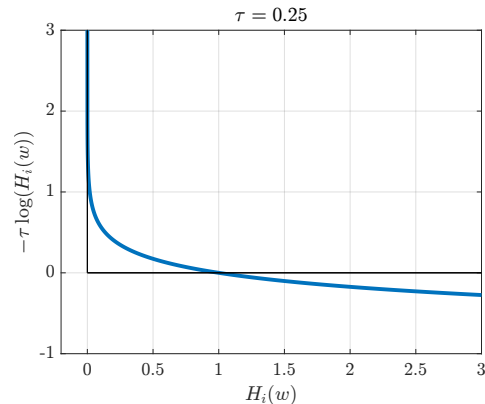
NLP with inequalities

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

Idea: put inequality constraint into objective

Barrier problem

$$\min_{w \in \mathbb{R}^n} F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$



$\tau \log(H_i(w))$ approximates:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

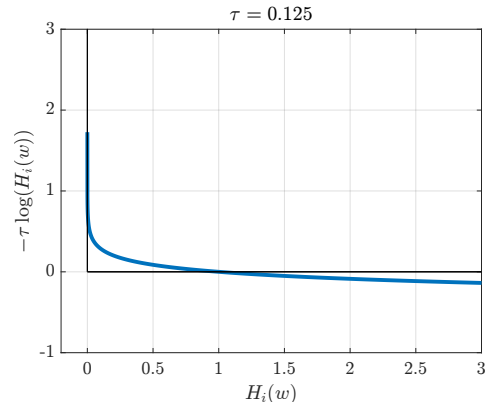
NLP with inequalities

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

Idea: put inequality constraint into objective

Barrier problem

$$\min_{w \in \mathbb{R}^n} F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$



$\tau \log(H_i(w))$ approximates:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

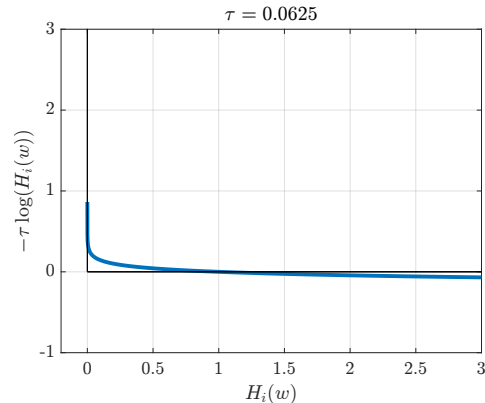
NLP with inequalities

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

Idea: put inequality constraint into objective

Barrier problem

$$\min_{w \in \mathbb{R}^n} F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$



$\tau \log(H_i(w))$ approximates:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

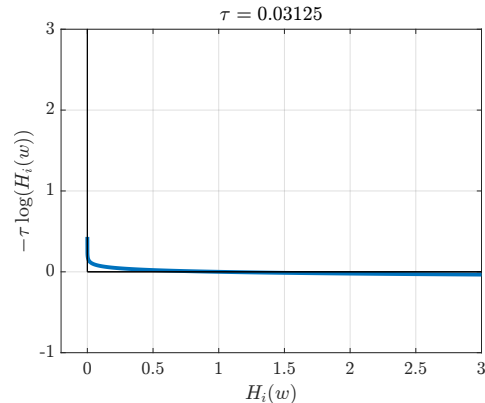
NLP with inequalities

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \end{aligned}$$

Idea: put inequality constraint into objective

Barrier problem

$$\min_{w \in \mathbb{R}^n} F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$



$\tau \log(H_i(w))$ approximates:

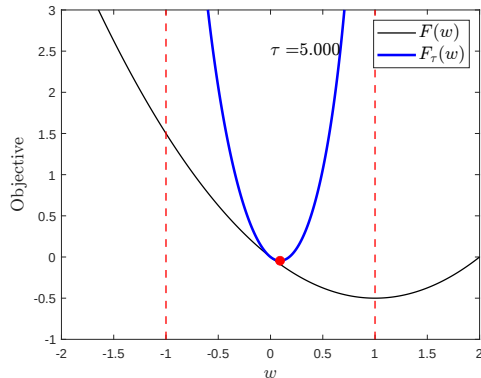
$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

Example NLP

$$\begin{aligned} \min_{w \in \mathbb{R}} \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_{w \in \mathbb{R}^2} 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$

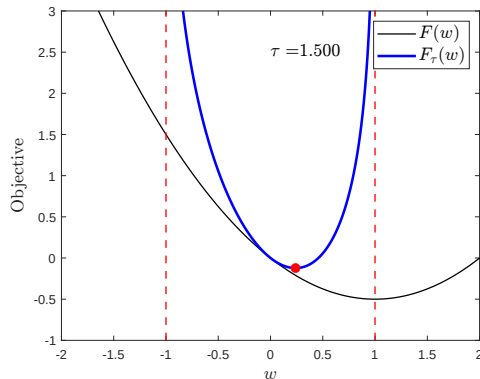


Example NLP

$$\begin{aligned} \min_{w \in \mathbb{R}} \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_{w \in \mathbb{R}^2} 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$

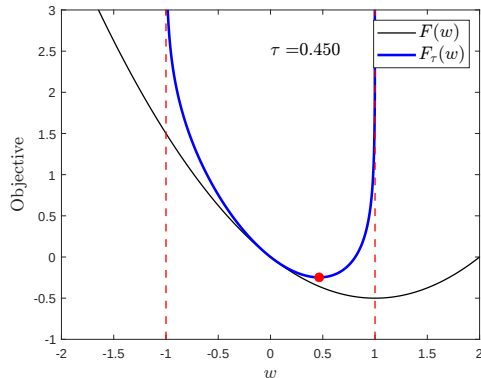


Example NLP

$$\begin{aligned} \min_{w \in \mathbb{R}} \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_{w \in \mathbb{R}^2} 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$

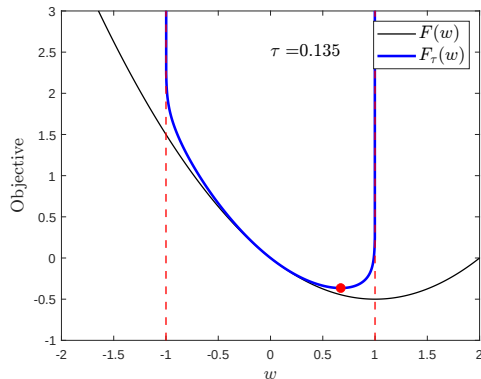


Example NLP

$$\begin{aligned} \min_{w \in \mathbb{R}} \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_{w \in \mathbb{R}^2} 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$

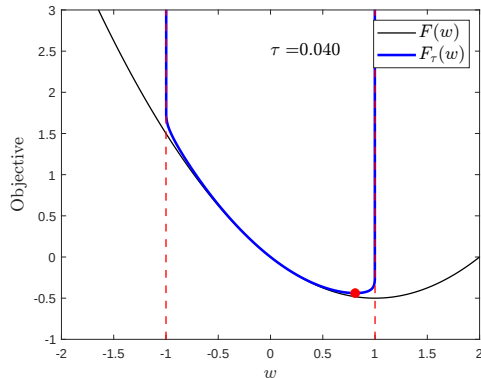


Example NLP

$$\begin{aligned} \min_{w \in \mathbb{R}} \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_{w \in \mathbb{R}^2} 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$

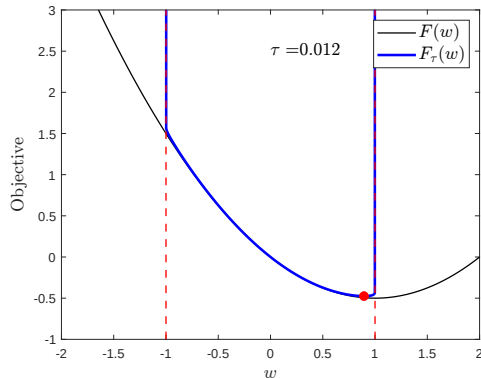


Example NLP

$$\begin{aligned} \min_{w \in \mathbb{R}} \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_{w \in \mathbb{R}^2} 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$

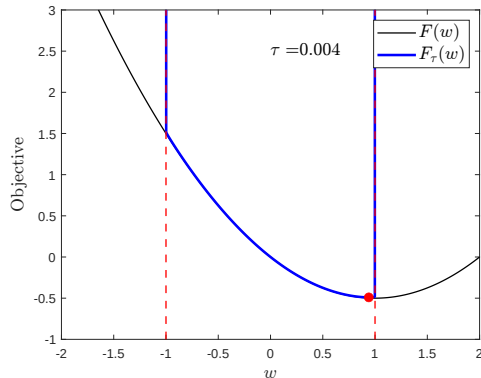


Example NLP

$$\begin{aligned} \min_{w \in \mathbb{R}} \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_{w \in \mathbb{R}^2} 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$



Barrier problem

$$\min_{w \in \mathbb{R}^n} F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$

KKT conditions ($\nabla F_\tau(w) = 0$)

$$\nabla F(w) - \tau \sum_{i=1}^m \frac{1}{H_i(w)} \nabla H_i(w) = 0$$

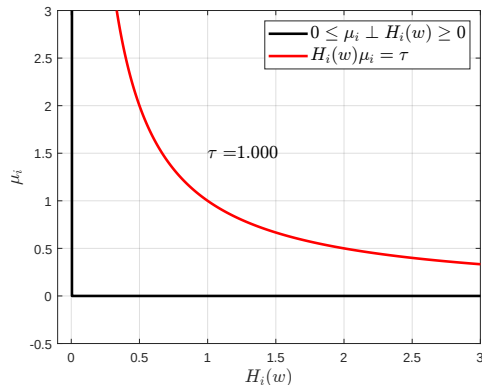
Introduce variable $\mu_i = \frac{\tau}{H_i(w)}$

Smoothed KKT conditions

$$\nabla F(w) - \nabla H(w)\mu = 0$$

$$H_i(w)\mu_i = \tau$$

$$(H_i(w) > 0, \mu_i > 0)$$



Barrier problem

$$\min_{w \in \mathbb{R}^n} F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$

KKT conditions ($\nabla F_\tau(w) = 0$)

$$\nabla F(w) - \tau \sum_{i=1}^m \frac{1}{H_i(w)} \nabla H_i(w) = 0$$

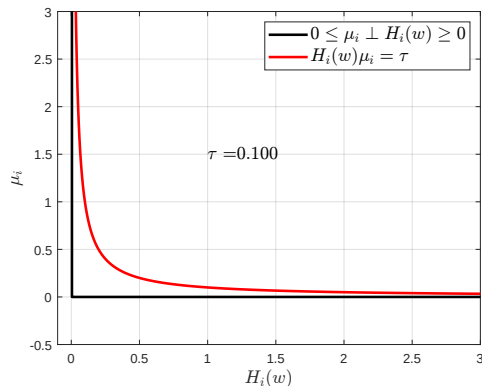
Introduce variable $\mu_i = \frac{\tau}{H_i(w)}$

Smoothed KKT conditions

$$\nabla F(w) - \nabla H(w)\mu = 0$$

$$H_i(w)\mu_i = \tau$$

$$(H_i(w) > 0, \mu_i > 0)$$



Barrier problem

$$\min_{w \in \mathbb{R}^n} F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$

KKT conditions ($\nabla F_\tau(w) = 0$)

$$\nabla F(w) - \tau \sum_{i=1}^m \frac{1}{H_i(w)} \nabla H_i(w) = 0$$

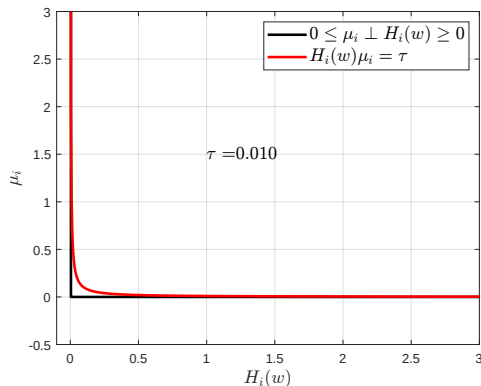
Introduce variable $\mu_i = \frac{\tau}{H_i(w)}$

Smoothed KKT conditions

$$\nabla F(w) - \nabla H(w) \mu = 0$$

$$H_i(w) \mu_i = \tau$$

$$(H_i(w) > 0, \mu_i > 0)$$



Barrier problem

$$\min_{w \in \mathbb{R}^n} F(w) - \tau \sum_{i=1}^m \log(H_i(w)) =: F_\tau(w)$$

KKT conditions ($\nabla F_\tau(w) = 0$)

$$\nabla F(w) - \tau \sum_{i=1}^m \frac{1}{H_i(w)} \nabla H_i(w) = 0$$

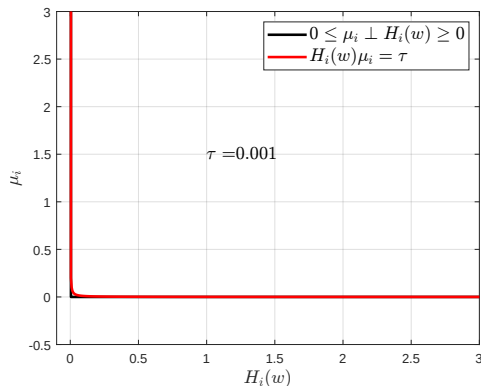
Introduce variable $\mu_i = \frac{\tau}{H_i(w)}$

Smoothed KKT conditions

$$\nabla F(w) - \nabla H(w) \mu = 0$$

$$H_i(w) \mu_i = \tau$$

$$(H_i(w) > 0, \mu_i > 0)$$



Nonlinear programming problem

$$\begin{aligned} \min_{\substack{w \in \mathbb{R}^n \\ s \in \mathbb{R}^{n_H}}} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0, \\ & H(w) - s = 0, \\ & s \geq 0 \end{aligned}$$

Smoothed KKT conditions

$$R_\tau(w, s, \lambda, \mu) = \begin{bmatrix} \nabla_w \mathcal{L}(w, \lambda, \mu) \\ G(w) \\ H(w) - s \\ \text{diag}(s)\mu - \tau \mathbf{1} \end{bmatrix} = 0$$

$(s, \mu > 0)$

Solve approximately with Newton's method for fixed τ

$$R_\tau(w, s, \lambda, \mu) + \nabla R_\tau(w, s, \lambda, \mu) \Delta z = 0$$

with $z = (w, s, \lambda, \mu)$

Line-search

Find $\alpha \in (0, 1)$

$$w^{k+1} = w^k + \alpha \Delta w$$

$$s^{k+1} = s^k + \alpha \Delta s$$

$$\lambda^{k+1} = \lambda^k + \alpha \Delta \lambda$$

$$\mu^{k+1} = \mu^k + \alpha \Delta \mu$$

such that $s^{k+1} > 0, \mu^{k+1} > 0$

and reduce $\tau \dots$

- ▶ Optimization problems come in many variants (LP, QP, NLP, MPCC, MINLP, OCP, ...)
- ▶ Each problem class be addressed with suitable software.
- ▶ Lagrangian function, convexity, and KKT conditions are important concepts.
- ▶ For convex problems KKT conditions sufficient for global optimality.
- ▶ Newton-type optimization for NLP solves the nonsmooth KKT conditions via Sequential Quadratic Programming (SQP) or via the Interior-Point Method.
- ▶ NLP solvers need to evaluate first and second order derivatives (e.g. via CasADi).

- ▶ Moritz Diehl, "Numerical optimization," Lecture notes, 2017.
- ▶ Jorge Nocedal, Stephen J. Wright, Numerical optimization. New York, NY: Springer New York, 2006.
- ▶ Stephen Boyd, Lieven Vandenberghe, Convex optimization. Cambridge University Press, 2004.