Optimization

Katja Mombaur

Institut für Technische Informatik, Universität Heidelberg

Robotics 2 - 17 June 2019

Contents of lecture

- Introduction of basic terms in optimization
- Introduction of different problem classes in optimization
- Nonlinear optimization (NLP)
- Optimality conditions

last time

- Newton's methods
- Handling of constraints
- Newton for constrained problems: SQP methods

today

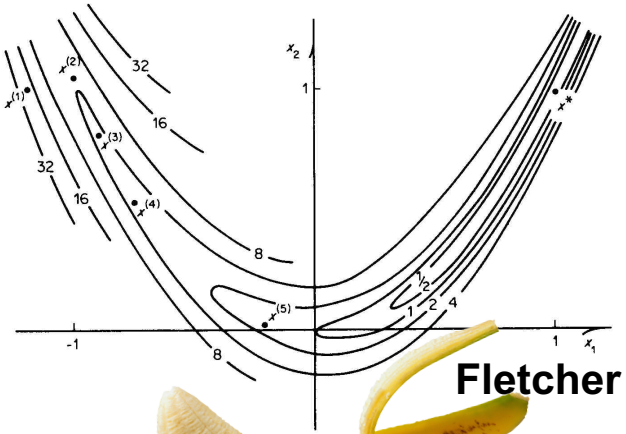
Classes of Optimization Problems: NLP

- Nonlinear Optimization Problem (NLP)

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s. t.} \quad & h(x) = 0 \\ & g(x) \geq 0 \end{aligned}$$

- Famous nonlinear example function
  - Rosenbrock function

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$



Fletcher

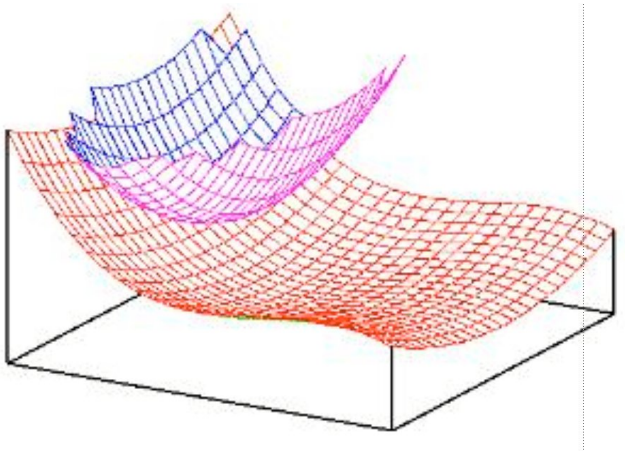
„Banana valley Function“

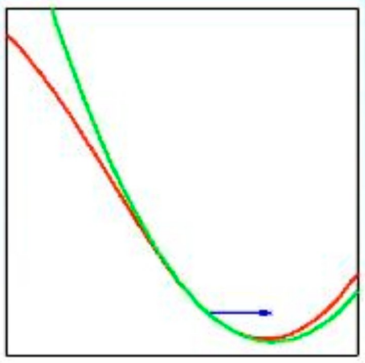
Newton‘s method

Δx_k minimizes a quadratic approximation of the nonlinear model

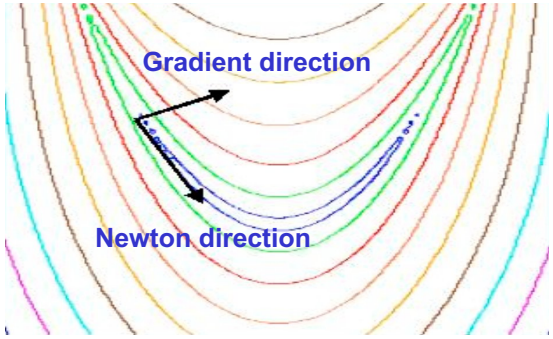
$$Q(p^k) = f(x^k) + \nabla f(x^k) p^k + \frac{1}{2} p^{kT} H^k p^k$$

with $H^k = \nabla^2 f(x^k)$





If the quadratic model is a good approximation of the nonlinear model, then a full step can be performed ($\alpha_k = 1$), otherwise it has to be adapted



Quasi-Newton methods

In practice, the evaluation of the second derivatives for the computation of the Hessian matrix H is very expensive!

➔ Use approximation B of Hessian H instead

➔ Make sure that approximation B is positive definite

$$x^{k+1} = x^k - M(x^k)\nabla f(x^k)$$
$$M(x) = B^{-1}(x) \quad B(x) \approx H(x)$$

- Methods are generally called Quasi-Newton methods or inexact Newton methods

- Steepest descent method: $p^k = -B^{-1}\nabla f(x^k)$ with $B = I$

Quasi Newton methods

Variants of Quasi Newton methods:

- Simplified Newton method (keep Hessian matrix of first point for all iterations)

$$B(x) \equiv H(x^0)$$

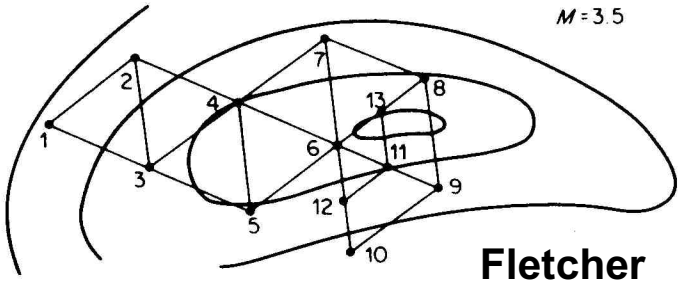
- Use same Hessian matrix for several iterations

if $\dfrac{\left\|\Delta x^{k+1}\right\|}{\left\|\Delta x^{k}\right\|} \leq \delta_{\max}$ then update $\quad B(x) \equiv H(x^{k+1})$

- Use update formulas for Hessian matrix (= Computation of new Hessian matrix from old Hessian matrix and gradient information).

Example for a different approach: Direct search methods

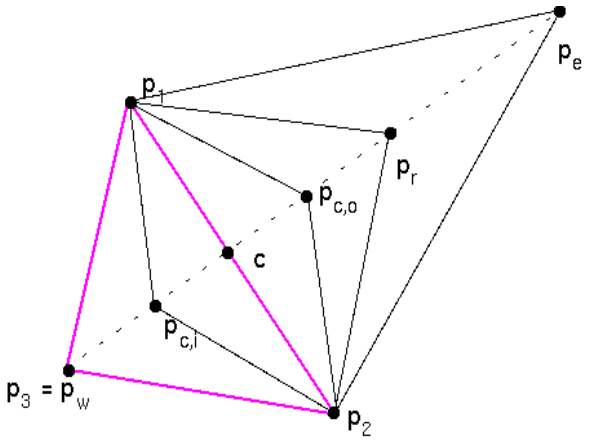
- Only use function evaluations
- Advantageous if first and second derivatives
  - don‘t exist (i.e. for non-smooth problems) or
  - are difficult to compute
- Do not explicitely compute or approximate derivatives
- Use a polytope with n+1 vertices in n-dimensional searchspace
- Polytope wanders through space adapting its shape



Fletcher

Direct search methods

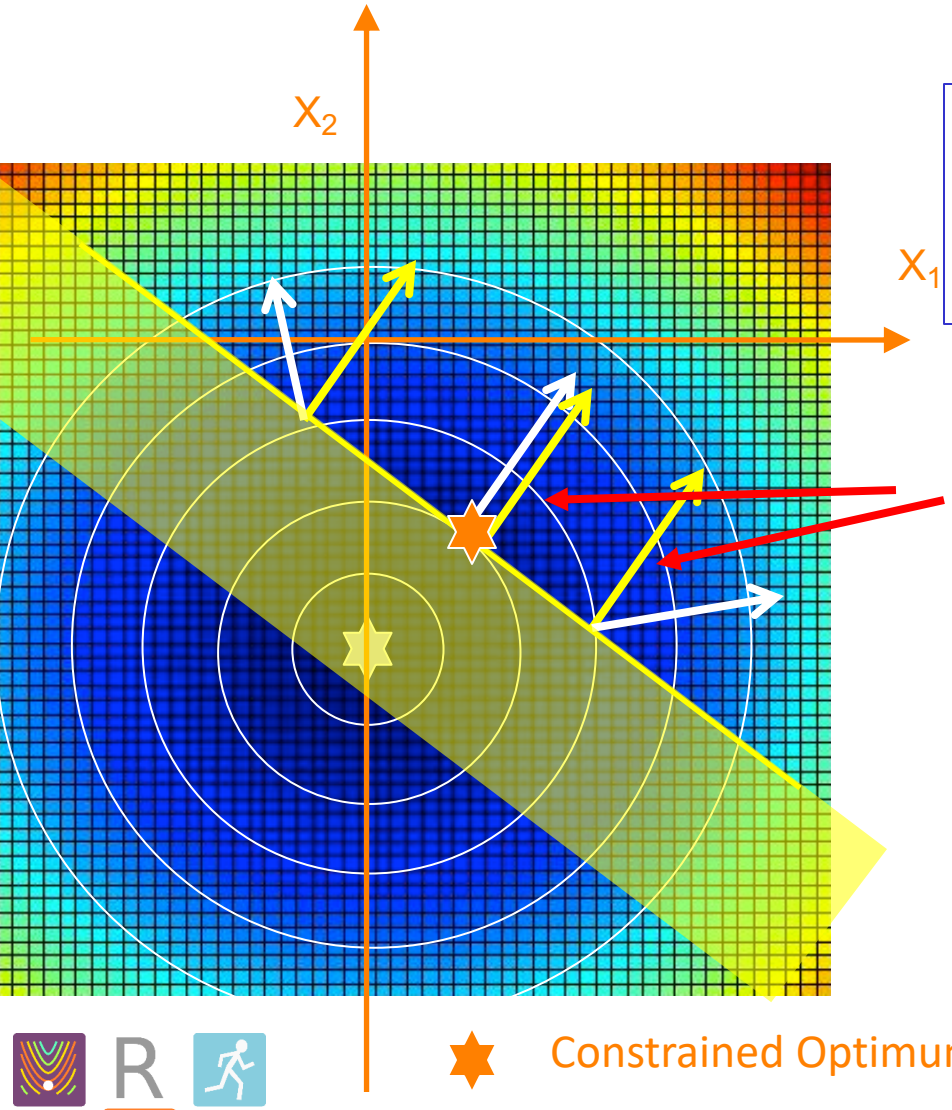
- Nelder-Mead:
  - worst point is replaced by a combination of reflection, expansion, contraction, scaling... Operations



- Other variants:
  - all except best point are replaced
  - ...

- Belong to the most frequently used optimization algorithms

Constrained optimization: Example problem – One active constraint



$$\min f(x)$$
$$h_1(x) := 1 + x_1 + x_2 \geq 0$$

Gradient ∇h_1 of the active inequality constraint

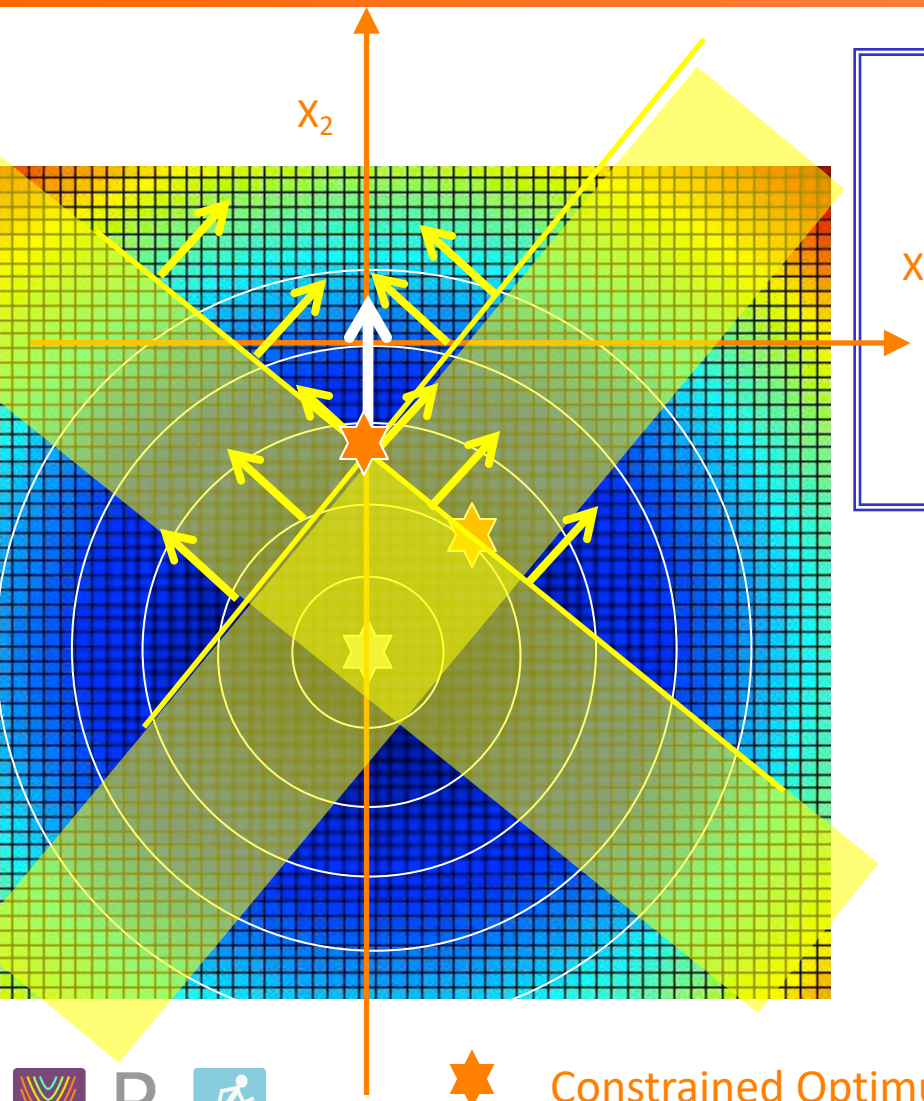
Constrained minimum

$$\nabla f(x^*) = \mu_1 \nabla h_1(x^*)$$

Lagrange multiplier

Constrained Optimum

Constrained optimization: Example problem – Two active constraints



$$\min f(x)$$

$$h_1(x) := 1 + x_1 + x_2 \geq 0$$

$$h_2(x) := 3 - x_1 + x_2 \geq 0$$

Equilibrium of forces

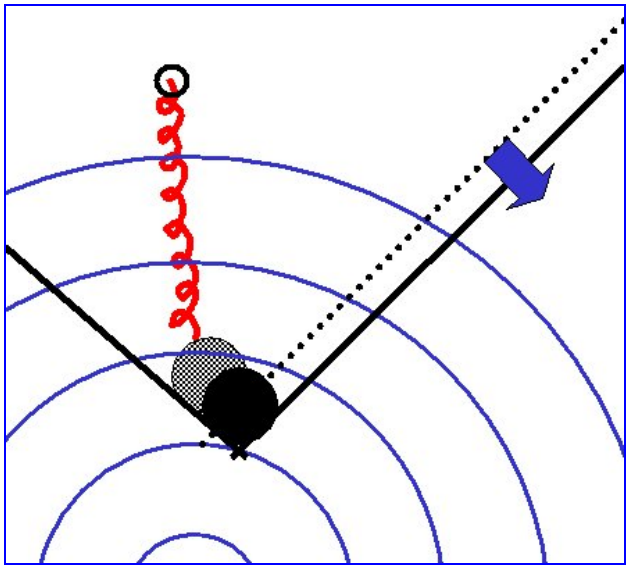
$$\nabla f(x^*) = \mu_1 \nabla h_1(x^*) + \mu_2 \nabla h_2(x^*)$$

„constraint forces“ (force generated by constraints)

$$\mu_1, \mu_2 \geq 0$$

Constrained Optimum

Multipliers as “shadow prices”



What happens if we relax a constraint? The feasible set gets bigger, such that the new minimum $f(x_\varepsilon^*)$ gets smaller.

How much can we gain?

$$f(x_\varepsilon^*) \approx f(x^*) - \mu\varepsilon$$

Old constraint: $h(x) \geq 0$

New constraint: $h(x) + \varepsilon \geq 0$

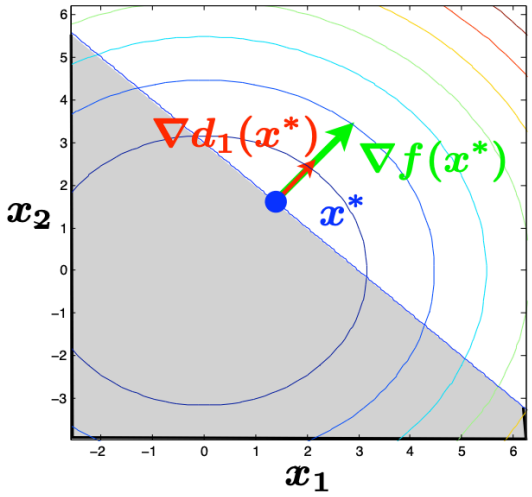
Multipliers show the hidden costs of constraints.

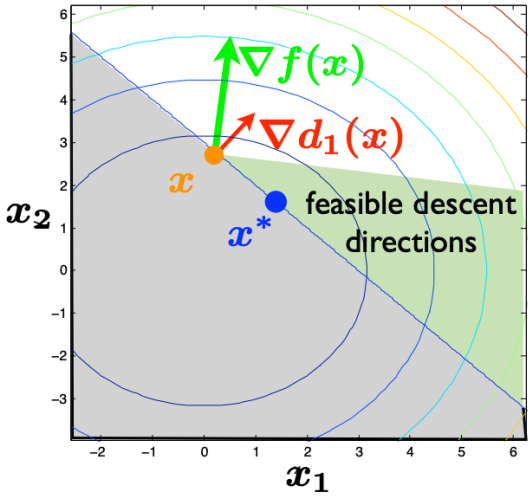
(also called „shadow prices“)

Constrained optimum

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) = x_1^2 + x_2^2$$

$$\text{subject to} \quad d_1(x) = x_1 + x_2 - 3 \geq 0$$





Kevin Carlberg

Treatment of constraints

Penalty and barrier methods

- Approximation of constrained optimization pb. by unconstrained pb.
- Penalty methods: penalty term for violation of constraints
- Barrier methods: penalty term for getting close to boundary
- Adjust penalty / barrier parameter (series of subproblems):

  $c \longrightarrow \infty$ approximation becomes increasingly accurate

- Resulting objective function is typically ill-conditioned
- Use of some unconstrained methods possible with special care

Better alternative: Lagrange methods

The Lagrange Function

For an NLP

$$\begin{aligned} \min\ & f(x) \\ g(x) &= 0 \\ h(x) &\geq 0 \end{aligned}$$

we can define the

Lagrange function - Modification of the objective function

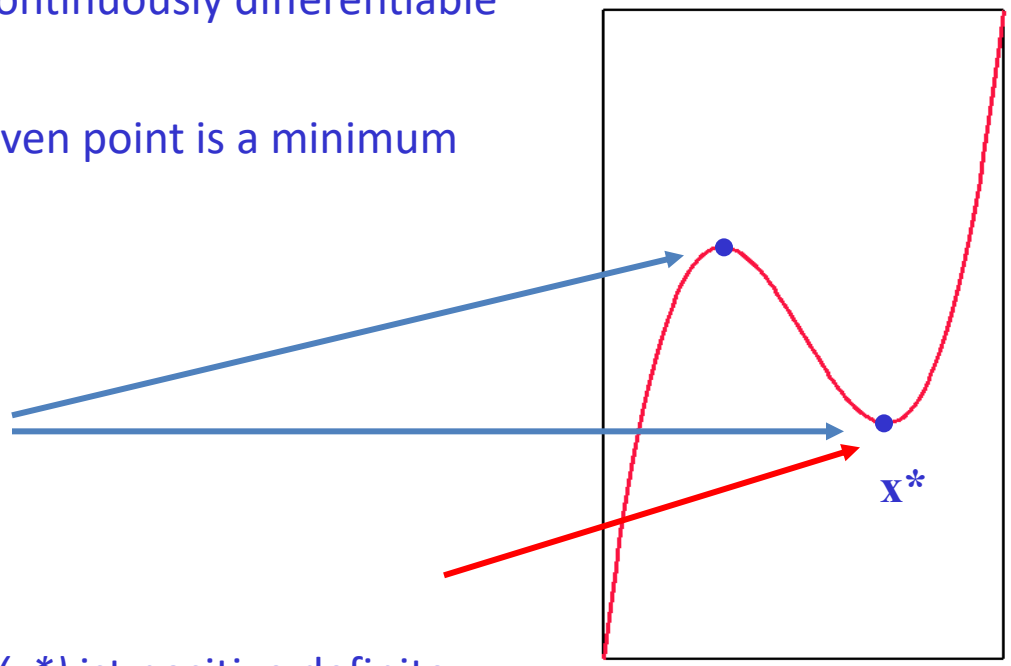
$$L(x,\lambda,\mu) := f(x^*) - \sum \lambda_i g_i(x) - \sum \mu_i h_i(x)$$

- The multipliers of the equality constraints λ_i can have an arbitrary sign in the solution
- The multipliers of the inequality constraints μ_i can not become negative (s. shadow prices)
- for inactive inequality constraints multipliers μ_i are equal to zero.

Reminder: Optimality conditions for the unconstrained case

Assumption: f is twice continuously differentiable

We want to test if a given point is a minimum



- Necessary condition:
  $\nabla f(x^*)=0$ (stationary Punkt)

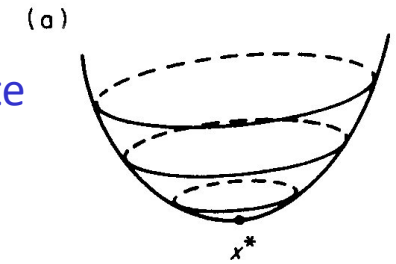
- Sufficient condition:
  x^* is a statioary point and $\nabla^2 f(x^*)$ ist positive definite

Reminder:
Different types of stationary points in the unconstrained case

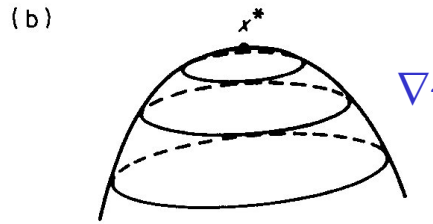
(a)-(c) x^* is stationary point $\nabla f(x^*)=0$

$\nabla^2 f(x^*)$ positive definite

Minimum



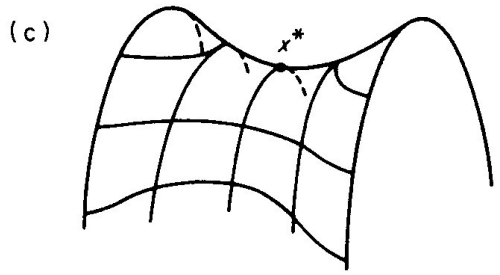
Minimum



Maximum

$\nabla^2 f(x^*)$ negative definite

Maximum



Saddle

$\nabla^2 f(x^*)$ indefinite

Saddle point

First order optimality conditions for constrained optimization problems

Necessary conditions of first order = Karush-Kuhn-Tucker (KKT) conditions:

- x^* is a local minimum and regular (i.e. the Jacobian (gradient matrix) of the active constraints is regular).
- Then there exist multipliers λ^*, μ^*, such that

$$\nabla_x L(x^*, \lambda^*, \mu^*) = 0$$

(„Equilibrium“ $\nabla_x L = \nabla f - \sum \lambda_i \nabla g_i - \sum \mu_i \nabla h_i$)

- and the so called complementarity condition holds:

$$\mu^{*T} h(x^*) = 0$$

i.e. $\mu_i^* = 0$ or $h_i(x^*) = 0$ for all i

- The equality constraints hold

$$g(x^*) = 0$$

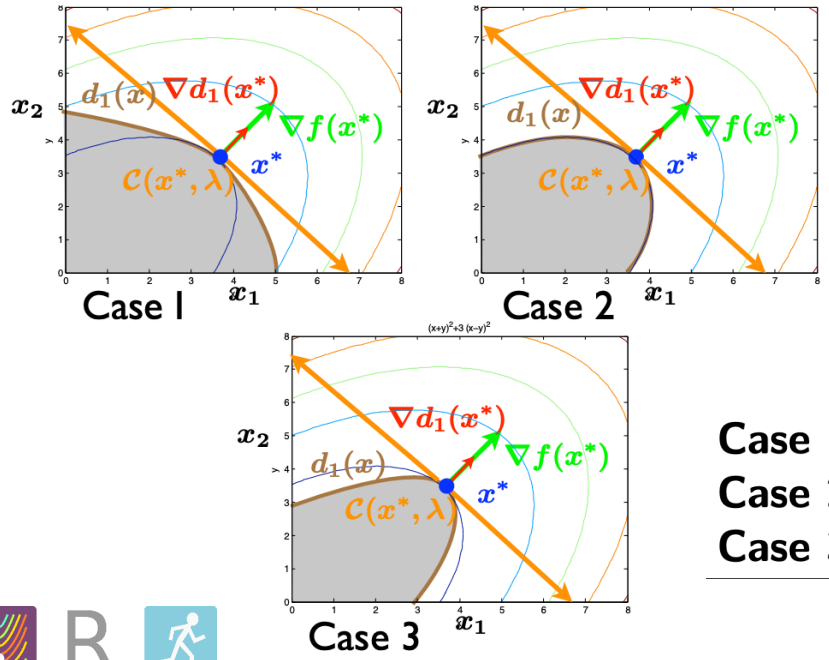
Second order optimality conditions for the constrained case

- Second order necessary conditions

D1. KKT conditions hold
D2. $p^T \nabla^2 \mathcal{L}(x^*, \gamma) p \geq 0$ *for all* $p \in \mathcal{C}(x^*, \gamma)$

- Second order sufficient conditions

E1. KKT conditions hold
E2. $p^T \nabla^2 \mathcal{L}(x^*, \gamma) p > 0$ *for all* $p \in \mathcal{C}(x^*, \gamma)$.



Case 1: E1 and E2 are satisfied (sufficient conditions hold)
Case 2: D1 and D2 are satisfied (necessary conditions hold)
Case 3: D1 holds, D2 does not (necessary conditions **failed**)

Kevin Carlberg

SQP – Sequential quadratic programming

- Is generally the most efficient method for nonlinear optimization

$$\begin{aligned} &\min f(x) \\ g(x) &= 0 \\ h(x) &\geq 0 \end{aligned}$$

- General idea: apply Newton's method to iterativey solving the KKT conditions

$$\nabla_x L(x^*, \lambda^*, \mu^*) = 0$$
$$g(x^*) = 0$$
$$h(x^*) \geq 0$$

Newton's method for KKT conditions

Calculate zeros of the equation

$$F(x^*) = \nabla_x L\,(x^*, \lambda^*, \mu^*) = 0$$

to satisfy first order optimality constraints

Taylor series:

$$F(x^{k+1}) = F(x^k) + \frac{d}{dx} F(x^k)(x^{k+1} - x^k) + \ldots = 0$$

$$\Rightarrow x^{k+1} = x^k \underbrace{- \left(\frac{d}{dx} F(x^k) \right)^{-1} F(x^k)}_{p^k}$$

Also take constraints into account, i.e. respect zeroes of equality constraints.

For simplicity, inequality will be omitted in derivation.

Newton‘s method for KKT conditions

- Find zeros of equation:

$$F(\underbrace{x, \lambda}_{y}) = \begin{pmatrix} \nabla f(x) - \nabla g(x) \cdot \lambda \\ g(x) \end{pmatrix} = 0$$

- Jacobian of F

$$\nabla_y F = \nabla_{x,\lambda} F(x, \lambda) = \begin{pmatrix} \nabla_x^2 \mathcal{L}(x, \lambda) & -\nabla_x g(x) \\ \nabla_x^T g(x) & 0 \end{pmatrix}$$

- Compute Newton iteration $y_{k+1} = y_k + \Delta y_k$
- from $\nabla_y F(y_k) \cdot \Delta y_k = -F(y_k)$

$$\begin{pmatrix} \nabla_x^2 \mathcal{L}(x_k, \lambda_k) & -\nabla_x g(x_k) \\ \nabla_x^T g(x_k) & 0 \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) - \nabla g(x_k) \cdot \lambda_k \\ g(x_k) \end{pmatrix}$$

$$\begin{pmatrix} \nabla_x^2 \mathcal{L}(x_k, \lambda_k) & -\nabla_x g(x_k) \\ \nabla_x^T g(x_k) & 0 \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \lambda_{k+1} \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) \\ g(x_k) \end{pmatrix}$$

Solution of the quadratic problem

- This is equivalent to the solution of a quadratic problem in every iteration of the SQP method (hence the name)

Hessian of the Lagrange function

$$\min_{\Delta x} \; (\nabla f^k)^T \Delta x + \frac{1}{2} \Delta x^T H^k \Delta x$$

$$g(x^k) + \nabla g(x^k)^T \Delta x = 0$$

$$h(x^k) + \nabla h(x^k)^T \Delta x \geq 0$$

- Use Active set strategy, to identify active inequality constraints
- **Practical SQP methods**
  - Use update formulas for the Hessian matrix (s. Quasi-Newton methods)
  - Use step size adjustment

What you should keep in mind about optimization

- Basic terminology: optimum, objective function, constraints (different types)
- Different types of optimization problems (in particular NLPs /QPs)
- Optimality conditions for unconstrained and constrained methods
- Handling of constraints / Lagrange function
- Optimization algorithms:
  - Steepest descent
  - Newton‘s method
  - SQP method
  - Direct search


Thank you very much for your attention!