Introduction to Optimization

September 27, 2019

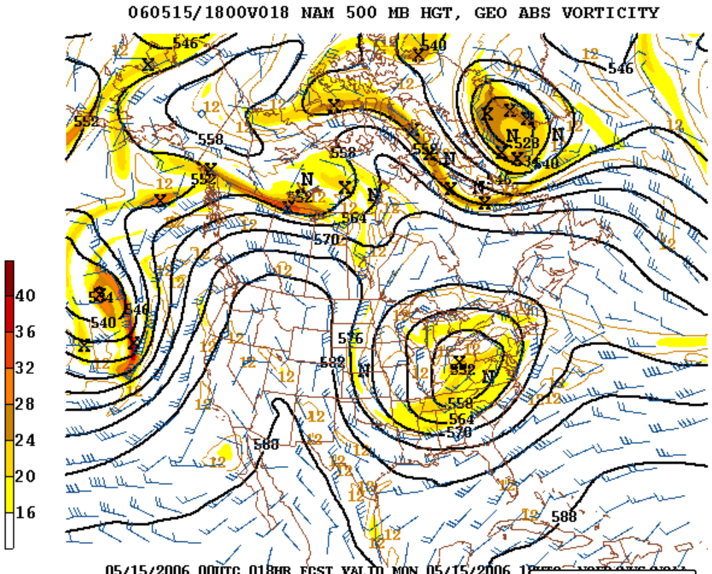
TC2 - Optimisation

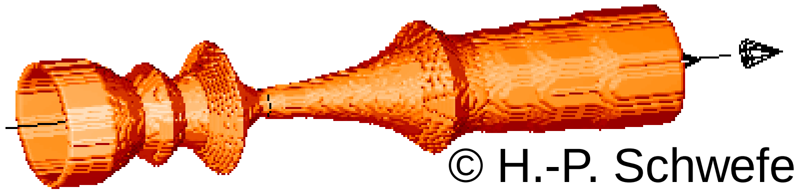
Université Paris-Saclay, Orsay, France

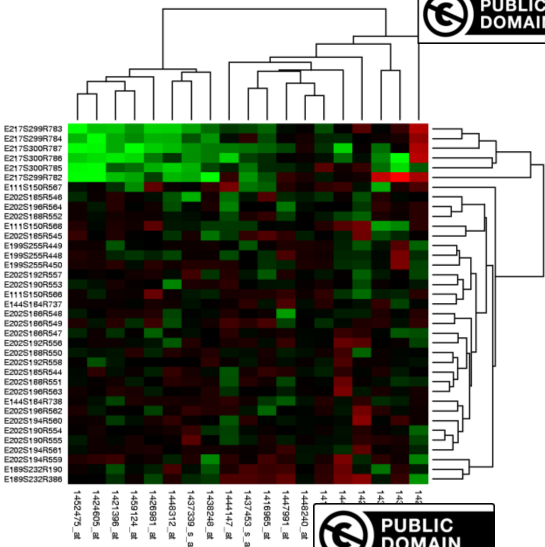
Anne Auger and Dimo Brockhoff

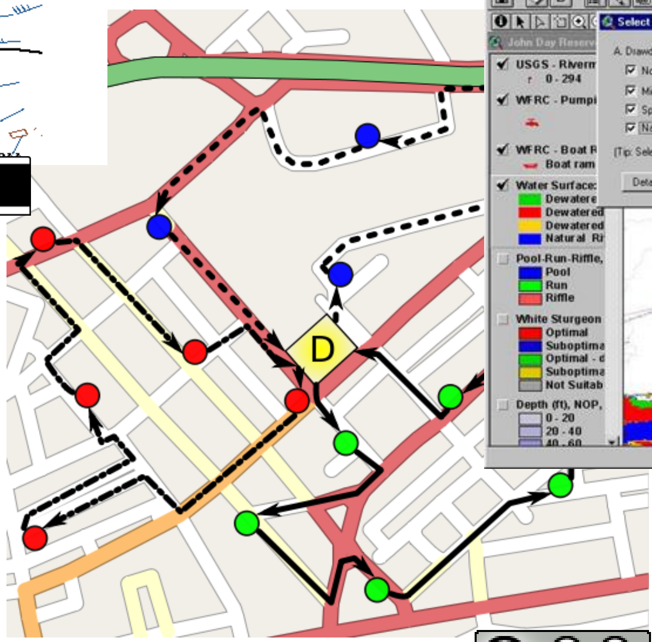
Inria Saclay – Ile-de-France

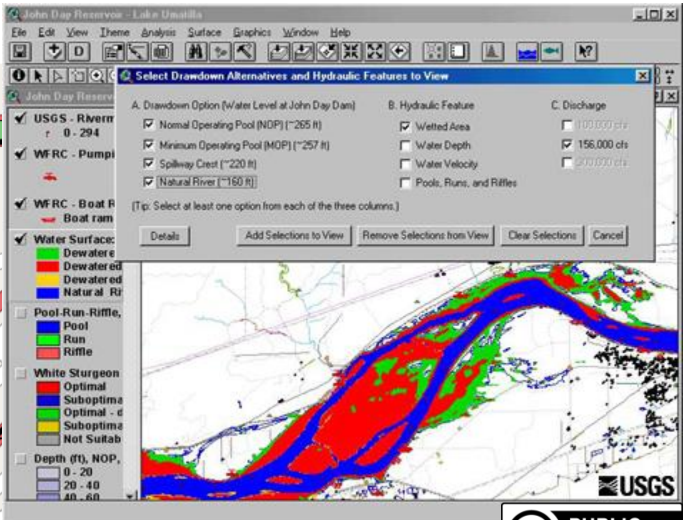
What is Optimization?











What is Optimization?

Typically, we aim at

- finding solutions x which minimize f(x) in the shortest time possible
  (maximization is reformulated as minimization)

- or finding solutions x with as small f(x) in the shortest time possible
  (if finding the exact optimum is not possible)

Course Overview

Date		Topic
Fri, 27.9.2019	DB	Introduction
Fri, 4.10.2019 (4hrs)	AA	Continuous Optimization I: differentiability, gradients, convexity, optimality conditions
Fri, 11.10.2019 (4hrs)	AA	Continuous Optimization II: constrained optimization, gradient-based algorithms, stochastic gradient
Fri, 18.10.2019 (4hrs)	DB	Continuous Optimization III: stochastic algorithms, derivative-free optimization, critical performance assessment
Wed, 30.10.2019	DB	Discrete Optimization I: graph theory, greedy algorithms
Fri, 15.11.2019	DB	Discrete Optimization II: dynamic programming, heuristics
Fri, 22.11.2018		final exam

Course Overview

Date		Topic
Fri, 27.9.2019	DB	Introduction
Fri, 4.10.2019 (4hrs)	AA	Continuous Optimization I: differentiability, gradients, convexity, optimality conditions
Fri, 11.10.2019 (4hrs)	AA	Continuous Optimization II: constrained optimization, gradient-based algorithms, stochastic gradient
Fri, 18.10.2019 (4hrs)	DB	Continuous Optimization III: stochastic algorithms, derivative-free optimization, critical performance assessment [1st written test]
Wed, 30.10.2019	DB	Discrete Optimization I: graph theory, greedy algorithms
Fri, 15.11.2019	DB	Discrete Optimization II: dynamic programming, heuristics [2nd written test]
Fri, 22.11.2018		final exam

Remarks

- possibly not clear yet what the lecture is about in detail

- but there will be always examples and small exercises to learn "on-the-fly" the concepts and fundamentals

Overall goals:

❶ give a broad overview of where and how optimization is used

❷ understand the fundamental concepts of optimization algorithms

❸ be able to apply common optimization algorithms on real-life (engineering) problems

The Exam

- open book: take as much material as you want
- multiple-choice
- Friday, 22nd of November 2019

- counts 60% of overall grade

Intermediate Written Exams ("contrôle continu")

- instead of a group project
- two smaller written exams/tests of about 20min each
  - October 18 & November 15
  - most likely one on continuous, one on discrete optimization
- goal: spread learning of lecture content over the course
- account 20% each to overall grade
- could also be multiple choice (not yet decided)

All information also available at

`http://www.cmap.polytechnique.fr/`
`~dimo.brockhoff/optimizationSaclay/2019/`

(in particular the lecture slides)

Advertisement I

Presentation

Blackbox Optimization

Lecture

Presentation Black Box Optimization Lecture

- Optional class “Black Box Optimization” ("Advanced Optimization")
- Taught by Anne Auger and me
- Advanced class, (even) closer to our actual research topic

Goals:

❶ present the latest knowledge on blackbox optimization algorithms and their foundations

❷ offer hands-on exercises on difficult common optimization problems

❸ give insights into what are current challenging research questions in the field of blackbox optimization (as preparation for a potential Master’s or PhD thesis in the field)

- ☺ relatively young research field with many interesting research questions (in both theory and algorithm design)
- ☺ related to real-world problems: also good for a job outside academia

Black Box Scenario

$$x \in \Omega \quad \longrightarrow \quad \textbf{black box} \quad \longrightarrow \quad \mathcal{F}(x) \in \mathbb{R}$$

Why are we interested in a black box scenario?

- objective function $\mathcal{F}$ often noisy, non-differentiable, or sometimes not even understood or available
- objective function $\mathcal{F}$ contains legacy or binary code, is based on numerical simulations or real-life experiments
- most likely, you will see such problems in practice...

Objective: find x with small $\mathcal{F}(x)$ with as few function evaluations as possible

assumption: internal calculations of algo irrelevant

What Makes an Optimization Problem Difficult?

- Search space too large

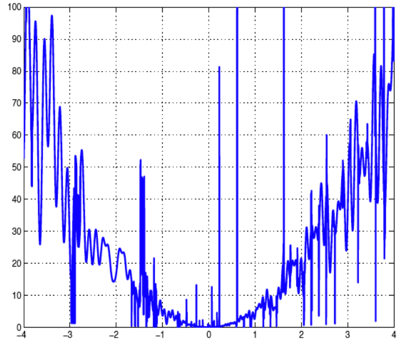
exhaustive search impossible

- Non conventional objective function or search space

mixed space, function that cannot be computed

- Complex objective function

non-smooth, non differentiable, noisy, ...



stochastic search algorithms
well suited because they:

- don't make many assumptions on $\mathcal{F}$
- are invariant wrt. translation/rotation of the search space, scaling of $\mathcal{F}$, ...
- are robust to noise

Planned Topics / Keywords

- Introduction to stochastic search algorithms, in particular
  - Evolutionary algorithms
  - Evolution Strategies and the CMA-ES algorithm in depth
  - Algorithms for large-scale problems (“big data”)
- Multiobjective optimization
- In more detail: Benchmarking black box algorithms

- Combination of lectures & exercises, theory & practice
- Connections with machine learning class of M. Sebag

Advertisement II: Master's Thesis Topics

RandOpt team
Inria and Ecole Polytechnique

Permanent members:
Anne Auger, Dimo Brockhoff, Nikolaus Hansen
https://team.inria.fr/randopt/team-members/

Master's theses available (PhD theses possible) :

- start anytime
- 6 months
- paid via Inria
- many topics around blackbox optimization
- theory ↔ algorithm design

constrained
large-scale
multiobjective
CMA-ES
blackbox optimization
expensive
theory
applications
algorithm design
benchmarking

http://randopt.gforge.inria.fr/thesisprojects/

Overview of Today's Lecture

- More examples of optimization problems
  - introduce some basic concepts of optimization problems such as domain, constraint, ...
- Beginning of continuous optimization part
  - typical difficulties in continuous optimization
  - differentiability
  - …   [we'll see how far we get]

General Context Optimization

Given:

set of possible solutions — *Search space*

quality criterion — *Objective function*

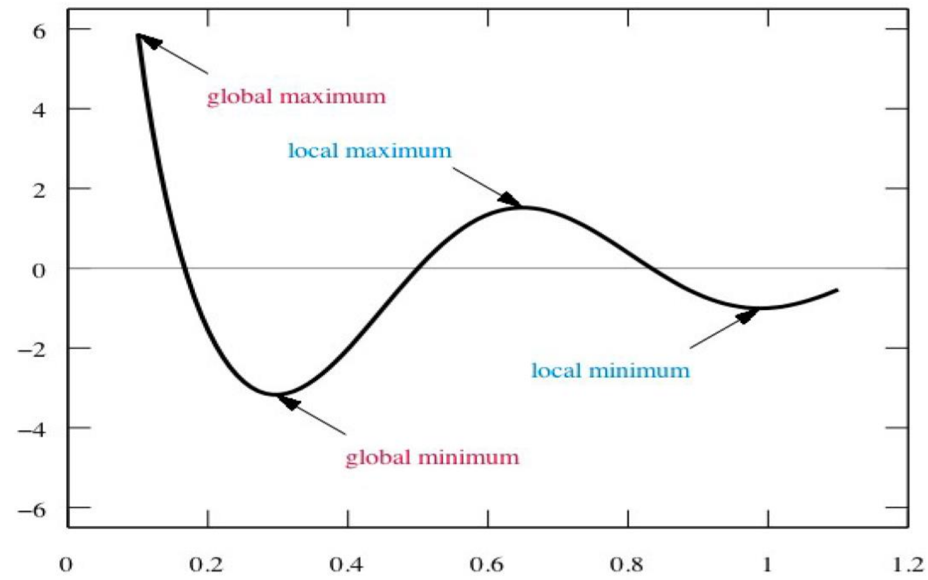
Objective:

Find the best possible solution for the given criterion

Formally:

Maximize or minimize

$$\mathcal{F} : \Omega \longmapsto \mathbb{R},$$
$$x \longmapsto \mathcal{F}(x)$$



Constraints

Maximize or minimize

$$\mathcal{F}: \Omega \longmapsto \mathbb{R},$$
$$x \longmapsto \mathcal{F}(x)$$

unconstrained Ω

Maximize or minimize

$$\mathcal{F}: \Omega \longmapsto \mathbb{R},$$
$$x \longmapsto \mathcal{F}(x)$$

where $g_i(x) \leq 0$

$$h_i(x) = 0$$

example of a constrained Ω

Constraints explicitly or implicitly define the feasible solution set

[e.g. ||x|| - 7 ≤ 0 vs. every solution should have at least 5 zero entries]

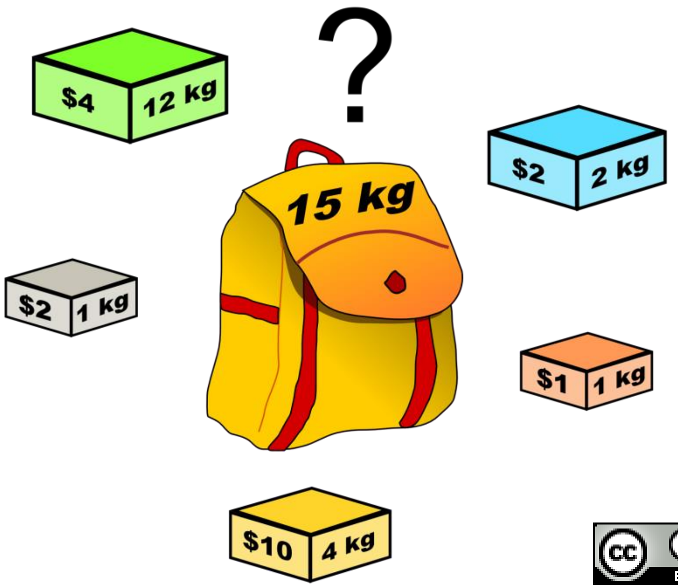
Hard constraints *must* be satisfied while soft constraints are preferred to hold but are not required to be satisfied

[e.g. constraints related to manufacturing precisions vs. cost constraints]

Knapsack Problem

- Given a set of objects with a given weight and value (profit)
- Find a subset of objects whose overall mass is below a certain limit and maximizing the total value of the objects

[Problem of ressource allocation with financial constraints]



Dake

$$\max \sum_{j=1}^{n} p_j x_j \quad \text{with } x_j \in \{0,1\}$$

$$\text{s.t.} \sum_{j=1}^{n} w_j x_j \leq W$$

$$\Omega = \{0,1\}^n$$

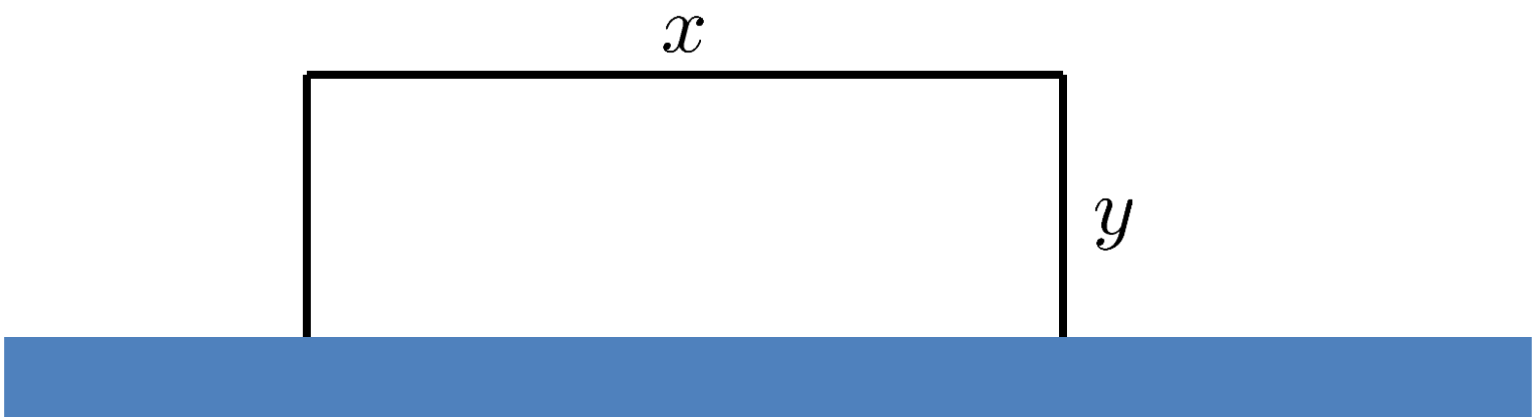
Traveling Salesperson Problem (TSP)

- Given a set of cities and their distances
- Find the shortest path going through all cities

$$\Omega = S_n \text{ (set of all permutations)}$$

Example 3: Continuous Optimization

A farmer has 500m of fence to fence off a rectangular field that is adjacent to a river. What is the maximal area he can fence off?



Exercise:

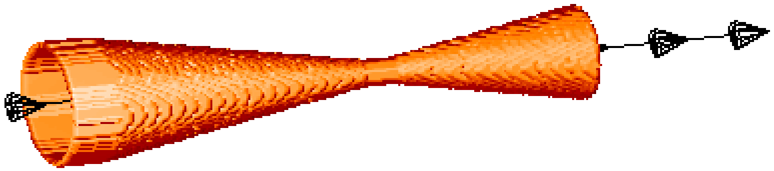
a) what is the search space?

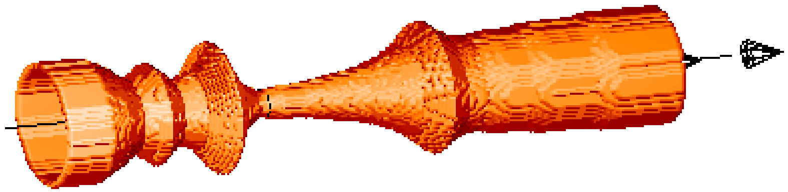
b) what is the objective function?

Example 4: A "Manual" Engineering Problem

Optimizing a Two-Phase Nozzle [Schwefel 1968+]

- maximize thrust under constant starting conditions
- one of the first examples of Evolution Strategies

initial design: 

final design: 

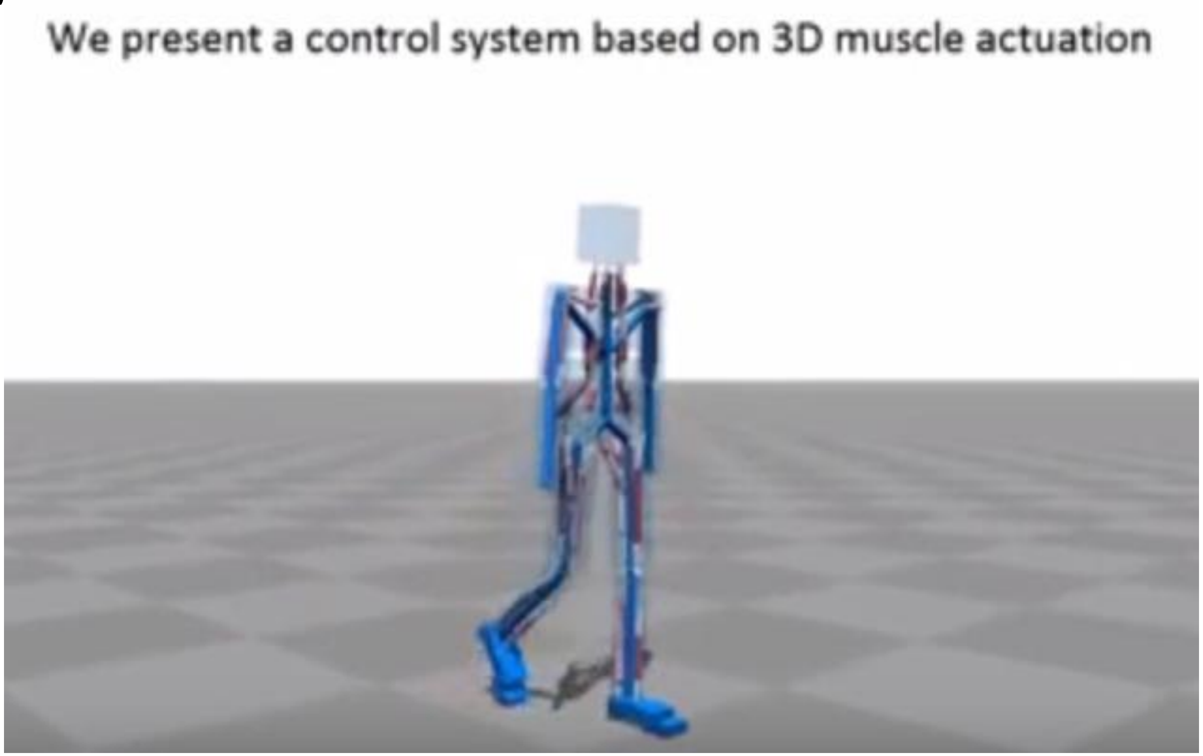
$\Omega =$ all possible nozzles of given number of slices

copyright Hans-Paul Schwefel

[http://ls11-www.cs.uni-dortmund.de/people/schwefel/EADemos/]

Example 5: Continuous Optimization Problem

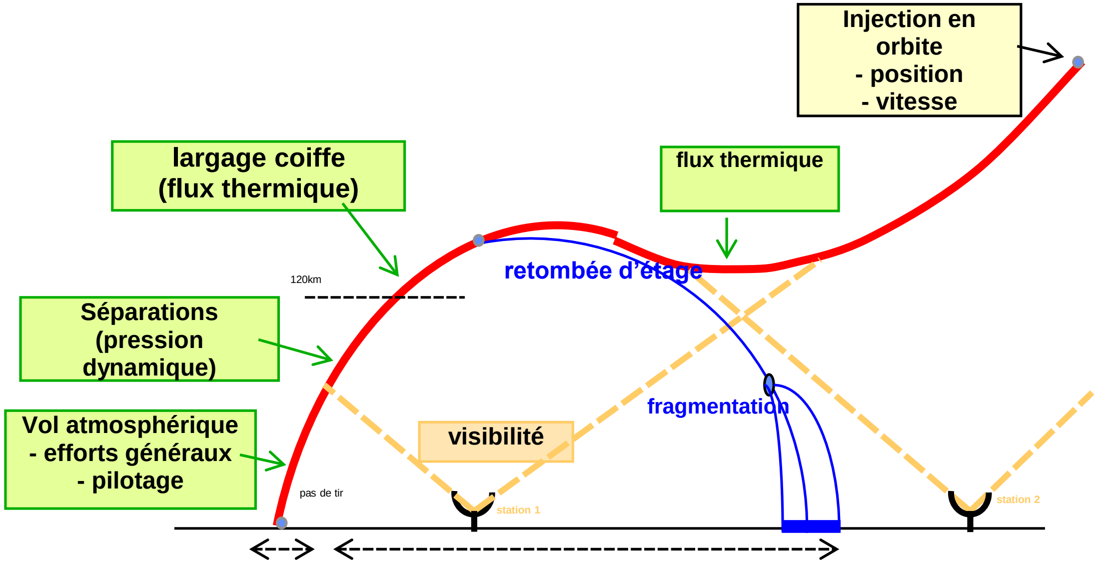
Computer simulation teaches itself to walk upright (virtual robots (of different shapes) learning to walk, through stochastic optimization (CMA-ES)), by Utrecht University:



https://www.youtube.com/watch?v=pgaEE27nsQw

T. Geitjtenbeek, M. Van de Panne, F. Van der Stappen: "Flexible Muscle-Based Locomotion for Bipedal Creatures", SIGGRAPH Asia, 2013.

Design of a Launcher



copyright by Astrium

Poppy

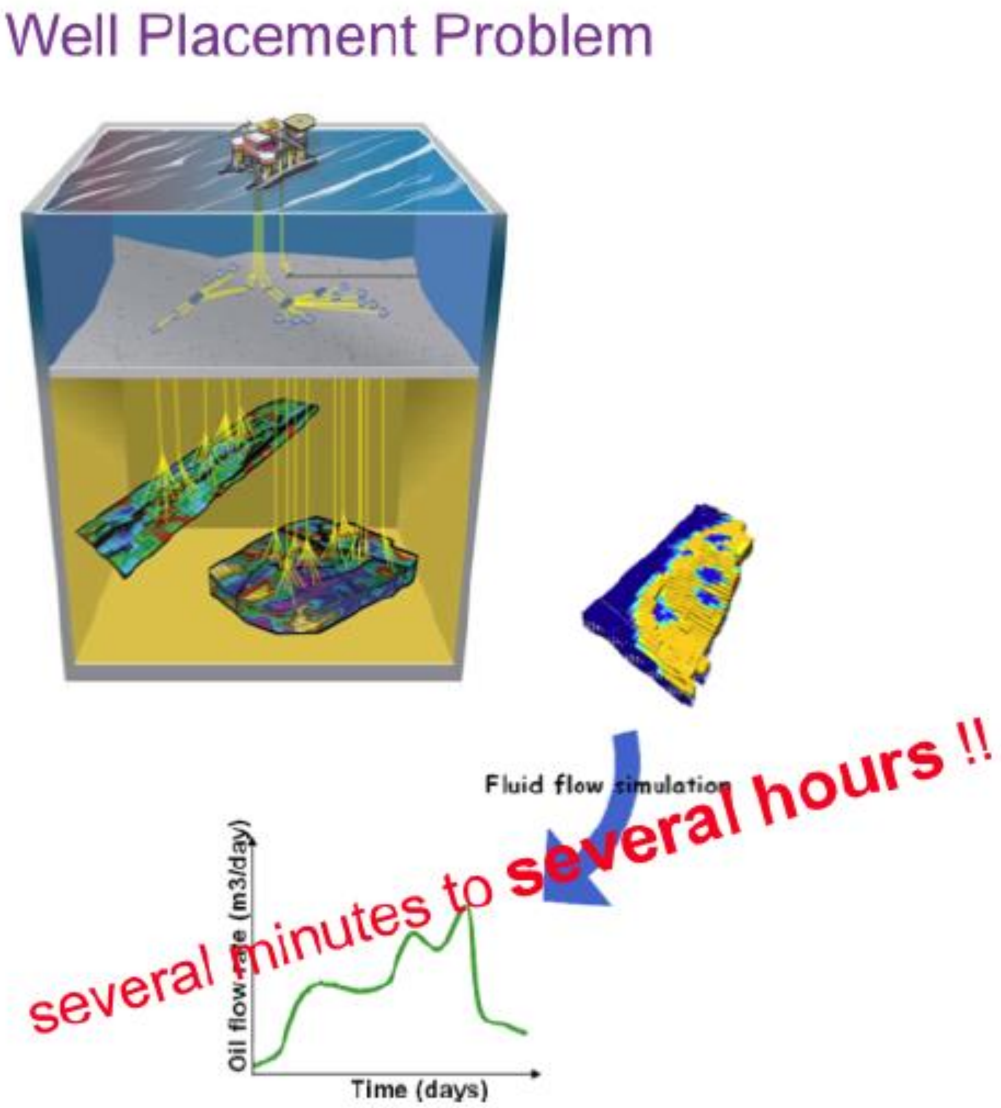
$$\Omega = \mathbb{R}^{23}$$

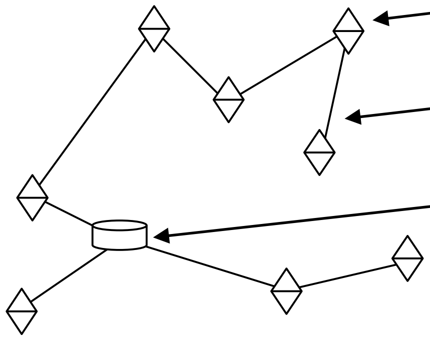
- Scenario: multi-stage launcher brings a satellite into orbit
- Minimize the overall cost of a launch
- Parameters: propellant mass of each stage / diameter of each stage / flux of each engine / parameters of the command law

23 continuous parameters to optimize

+ constraints

Example 7: An Expensive Real-World Problem





for a given structure, per well:

- angle & distance to previous well
- well depth

structure + $\mathbb{R}_+^3 \cdot \#\text{wells}$

$\sigma \in \Omega$: variable length!

Example 8: Data Fitting – Data Calibration

Objective

- Given a sequence of data points $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^p \times \mathbb{R}, i = 1, \ldots, N$, find a model "$y = f(\boldsymbol{x})$" that *"explains"* the data

  experimental measurements in biology, chemistry, ...

- In general, choice of a parametric model or family of functions $(f_\theta)_{\theta \in \mathbb{R}^n}$

  use of expertise for choosing model

  or only a simple model is affordable (e.g. linear, quadratic)

- Try to find the parameter $\theta \in \mathbb{R}^n$ fitting best to the data

Fitting best to the data

Minimize the quadratic error:

$$\min_{\theta \in \mathbb{R}^n} \sum_{i=1}^{N} |f_\theta(\boldsymbol{x}_i) - y_i|^2$$

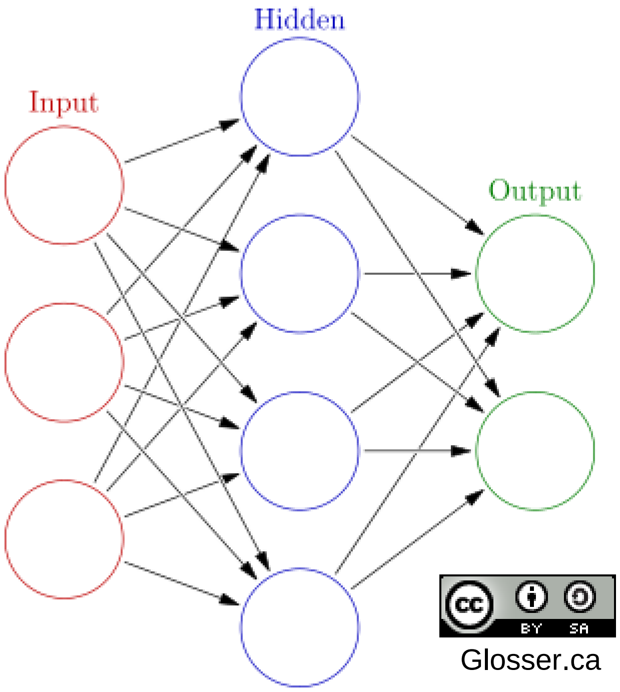
Example 9: Deep Learning

Actually the same idea:

match model best to given data

Model here:

artificial neural nets
with many hidden layers
(aka deep neural networks)



Parameters to tune:

- weights of the connections (continuous parameter)
- topology of the network (discrete)
- firing function (less common)

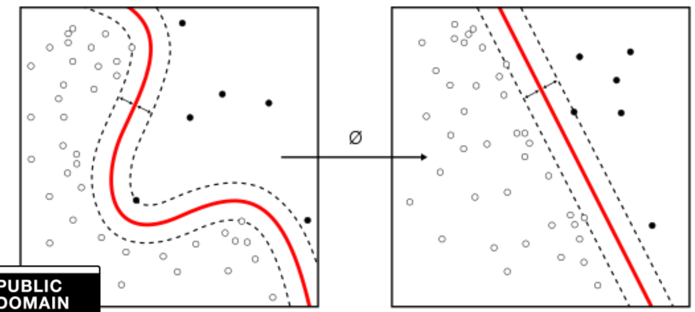
Specificity:

- large amount of training data, hence often batch learning

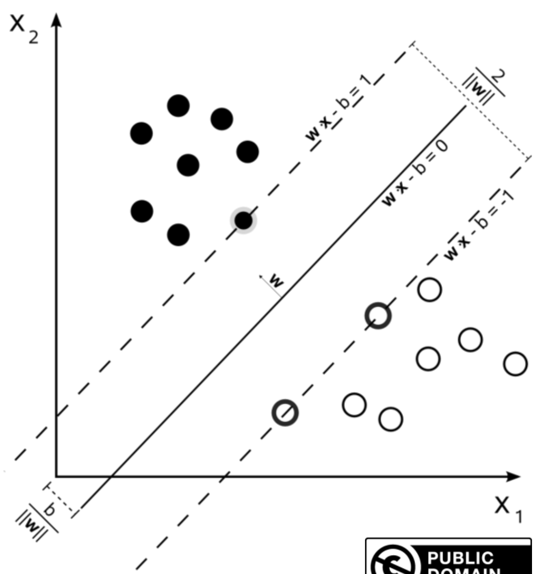
Example 10: Classification with SVMs

Scenario:

- supervised learning of 2-class samples
- Support Vector Machines (SVMs):
  - decide to which class a new sample belongs



  - learns from the training data the "best linear model" (= a hyperplane separating the two classes); non-linear transformations possible via the kernel trick



- hard margin (when data linearly separable):
  $\min \|\boldsymbol{w}\| \text{ s.t. } y_i\,(\boldsymbol{w}\cdot\boldsymbol{x}_i) - b \geq 1\ \forall 1 \leq i \leq n$
- soft margin (e.g. via hinge loss):
  $$\min \left[\frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i) - b)\right] + \lambda\left|\left|\boldsymbol{w}\right|\right|^2$$
  with λ being a tradeoff parameter (constrained optimization)

Example 11: Hyperparameter Tuning

Scenario:

- many existing algorithms (in ML and elsewhere) have internal parameters
    - "In machine learning, a hyperparameter is a parameter whose value is set before the learning process begins." --- Wikipedia
    - can be model parameters
        - #trees in random forest
        - #nodes in neural net
        - …
    - or other generic parameters such as learning rates, …
- choice has typically a big impact and is not always obvious
- search space often mixed discrete-continuous or even categorical

Coffee Tasting Problem

- Find a mixture of coffee in order to keep the coffee taste from one year to another
- Objective function = opinion of one expert

M. Herdy: "Evolution Strategies with subjective selection", 1996

Many Problems, Many Algorithms?

Observation:

- Many problems with different properties
- For each, it seems a different algorithm?

In Practice:

- often most important to categorize your problem first in order to find / develop the right method
- → problem types

Algorithm design is an art,
what is needed is skill, intuition, luck, experience,
special knowledge and craft

freely translated and adapted from Ingo Wegener (1950-2008)

Problem Types

- discrete vs. continuous
  - discrete: integer (linear) programming vs. combinatorial problems
  - continuous: linear, quadratic, smooth/nonsmooth, blackbox/DFO, ...
  - both discrete&continuous variables: mixed integer problem
  - categorical variables ("no order")
- unconstrained vs. constrained (and then which type of constraint)

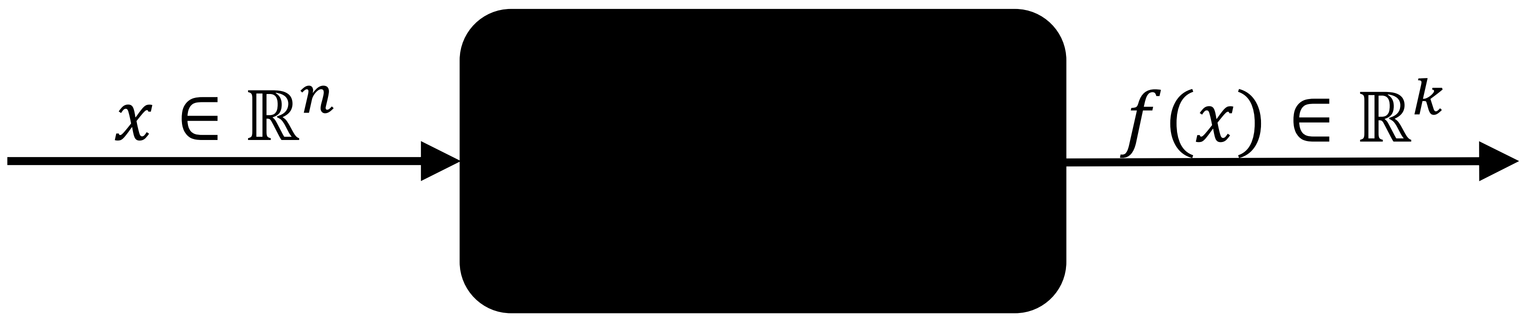
Not covered in this introductory lecture:

- deterministic vs. stochastic outcome of objective function(s)
- one or multiple objective functions

Example: Numerical Blackbox Optimization

Typical scenario in the continuous, unconstrained case:

Optimize $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}^k$



derivatives not available or not useful

General Concepts in Optimization

- search domain
  - discrete or continuous or mixed integer or even categorical
  - finite vs. infinite dimension
- constraints
  - bound constraints (on the variables only)
  - linear/quadratic/non-linear constraints
  - blackbox constraints
  - many more

(see e.g. Le Digabel and Wild (2015), https://arxiv.org/abs/1505.07881)

Further important aspects (in practice):

- deterministic vs. stochastic algorithms
- exact vs. approximation algorithms vs. heuristics
- anytime algorithms
- simulation-based optimization problem / expensive problem

continuous optimization

Continuous Optimization

- Optimize f: $\begin{cases} \Omega \subset \mathbb{R}^n \to \mathbb{R} \\ x = (x_1, \ldots, x_n) \to f(x_1, \ldots, x_n) \end{cases}$

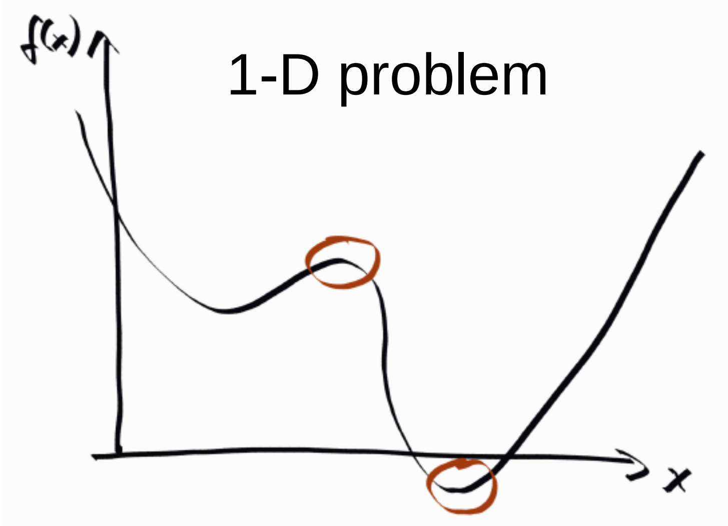
  (x_1) $\in \mathbb{R}$

  unconstrained optimization

- Search space is continuous, i.e. composed of real vectors $x \in \mathbb{R}^n$

- $n =$ dimension of the problem / dimension of the search space $\mathbb{R}^n$ (as vector space)

1-D problem



2-D level sets

Unconstrained vs. Constrained Optimization

Unconstrained optimization

$$\inf \{f(x) \mid x \in \mathbb{R}^n\}$$

Constrained optimization

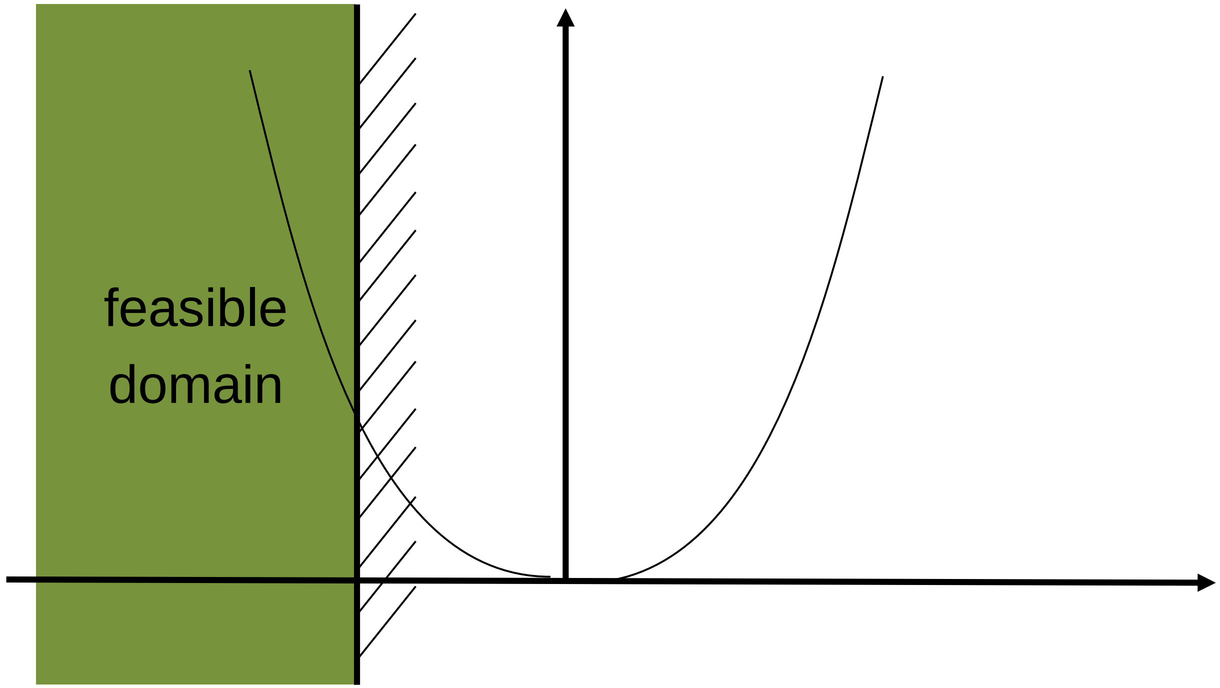
- Equality constraints: $\inf \{f(x) \mid x \in \mathbb{R}^n, g_k(x) = 0, 1 \leq k \leq p\}$

- Inequality constraints: $\inf \{f(x) \mid x \in \mathbb{R}^n, g_k(x) \leq 0, 1 \leq k \leq p\}$

where always $g_k\colon \mathbb{R}^n \to \mathbb{R}$

Example of a Constraint

$$\min_{x \in \mathbb{R}} f(x) = x^2 \text{ such that } x \leq -1$$



Analytical Functions

Example: 1-D

$$f_1(x) = a(x - x_0)^2 + b$$

where $x, x_0, b \in \mathbb{R}, a \in \mathbb{R}$

Generalization:

convex quadratic function

$$f_2(x) = (x - x_0)^T A\,(x - x_0) + b$$

where $x, x_0 \in \mathbb{R}^n, b \in \mathbb{R}\ \ , A \in \mathbb{R}^{\{\mathrm{n} \times n\}}$

and A symmetric positive definite (SPD)

Exercise:

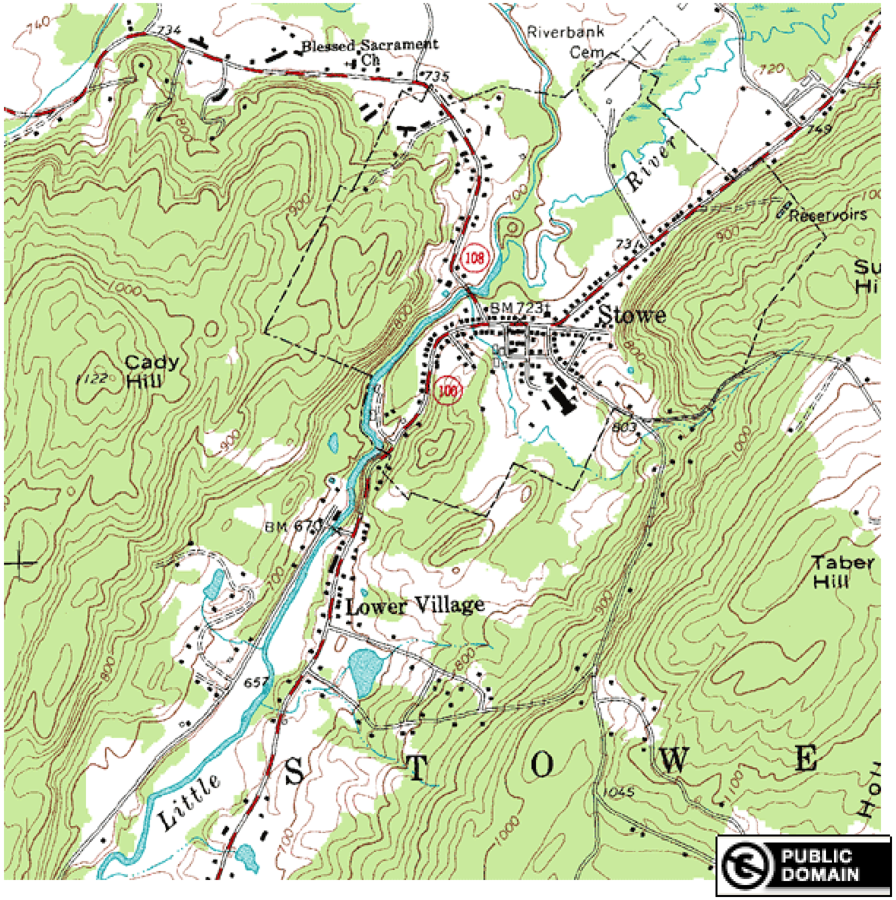
What is the minimum of $f_2(x)$?

Continuation of exercise:
What are the level sets of f_2?

Reminder: level sets of a function

$$L_c = \{x \in \mathbb{R}^n \mid f(x) = c\}$$

(similar to topography lines / level sets on a map)



Levels Sets of Convex Quadratic Functions

Continuation of exercise:
What are the level sets of f_2?

- Probably too complicated in general, thus an example here
- Consider $A = \begin{pmatrix} 9 & 0 \\ 0 & 1 \end{pmatrix}, b = 0, n = 2$
  a) Compute $f_2(x)$.
  b) Plot the level sets of $f_2(x)$.
  c) More generally, for $n = 2$, if A is SPD with eigenvalues $\lambda_1 = 9$ and $\lambda_2 = 1$, what are the level sets of $f_2(x)$?

What Makes a Function Difficult to Solve?

- dimensionality

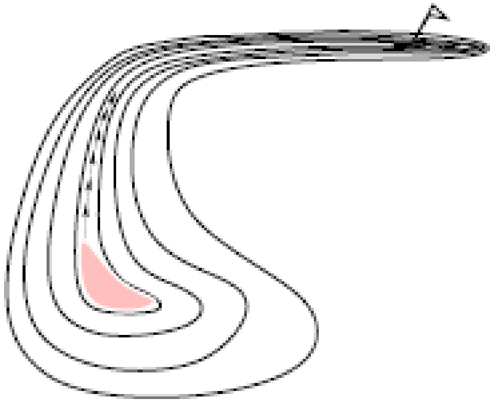
(considerably) larger than three

- non-separability

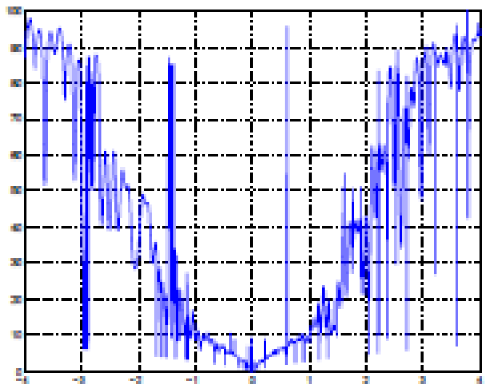
dependencies between the objective variables

- ill-conditioning
- ruggedness

non-smooth, discontinuous, multimodal, and/or noisy function



a narrow ridge



cut from 3D example, solvable with an evolution strategy

Curse of Dimensionality

- The term *Curse of dimensionality* (Richard Bellman) refers to problems caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space.

- Example: Consider placing 100 points onto a real interval, say $[0,1]$. To get similar coverage, in terms of distance between adjacent points, of the 10-dimensional space $[0,1]^{10}$ would require $100^{10} = 10^{20}$ points. The original 100 points appear now as isolated points in a vast empty space.

- Consequently, a search policy (e.g. exhaustive search) that is valuable in small dimensions might be useless in moderate or large dimensional search spaces.

Definition (Separable Problem)

A function f is separable if

$$\underset{(x_1,\ldots,x_n)}{\operatorname{argmin}} f(x_1,\ldots,x_n) = \left(\underset{x_1}{\operatorname{argmin}} f(x_1,\ldots),\ldots,\underset{x_n}{\operatorname{argmin}} f(\ldots,x_n)\right)$$

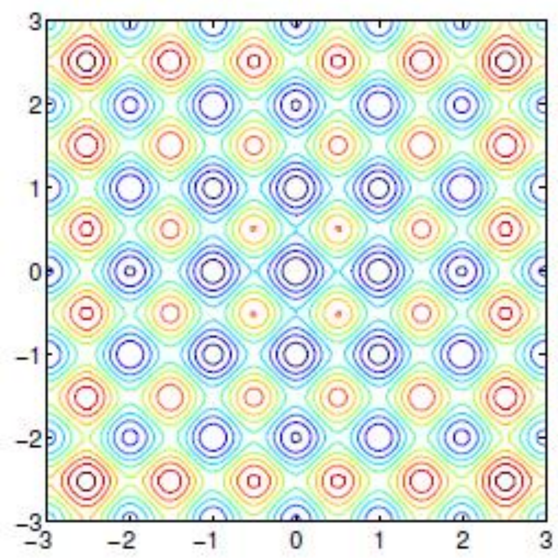
$\Longrightarrow$ it follows that f can be optimized in a sequence of n independent 1-D optimization processes

Example:

Additively decomposable functions

$$f(x_1,\ldots,x_n) = \sum_{i=1}^{n} f_i(x_i)$$

Rastrigin function



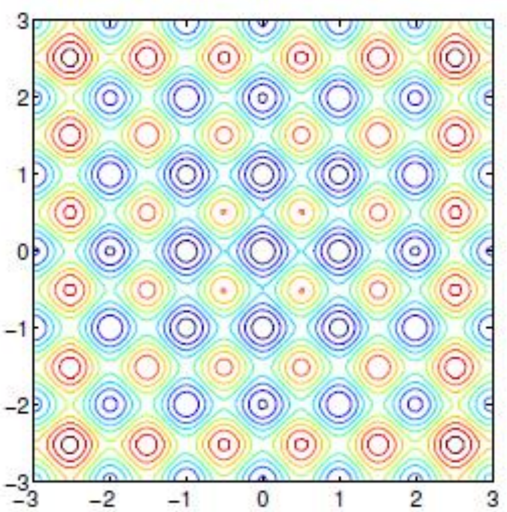
Non-Separable Problems

Building a non-separable problem from a separable one [1,2]

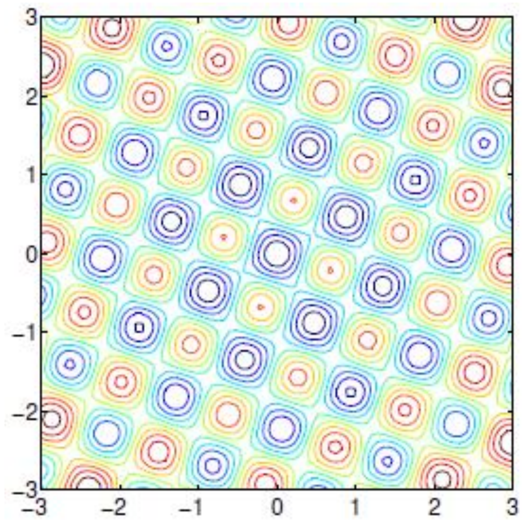
Rotating the coordinate system

- $f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$ separable
- $f : \boldsymbol{x} \mapsto f(R\boldsymbol{x})$ non-separable

R rotation matrix



$\overset{R}{\longrightarrow}$



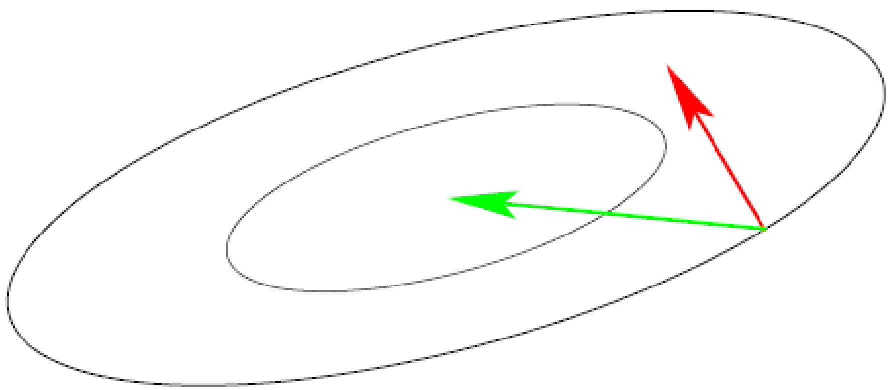
[1] N. Hansen, A. Ostermeier, A. Gawelczyk (1995). "On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation". Sixth ICGA, pp. 57-64, Morgan Kaufmann

[2] R. Salomon (1996). "Reevaluating Genetic Algorithm Performance under Coordinate Rotation of Benchmark Functions; A survey of some theoretical and practical aspects of genetic algorithms." BioSystems, 39(3):263-278

Consider the convex-quadratic function

$$f(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^*)^T H(\boldsymbol{x}-\boldsymbol{x}^*) = \frac{1}{2}\sum_i h_{i,i}x_i^2 + \frac{1}{2}\sum_{i,j} h_{i,j}x_i x_j$$

H is Hessian matrix of f and symmetric positive definite



gradient direction $-f'(x)^T$

Newton direction $-H^{-1}f'(x)^T$

Ill-conditioning means squeezed level sets (high curvature). Condition number equals nine here. Condition numbers up to 10^{10} are not unusual in real-world problems.

If $H \approx I$ (small condition number of H) first order information (e.g. the gradient) is sufficient. Otherwise second order information (estimation of H^{-1}) information necessary.

Different Notions of Optimum

Unconstrained case

- local vs. global
  - local minimum $\boldsymbol{x}^*$: $\exists$ a neighborhood V of $\boldsymbol{x}^*$ such that $\forall \boldsymbol{x} \in \mathrm{V}: f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*)$
  - global minimum: $\forall \boldsymbol{x} \in \Omega: f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*)$
- strict local minimum if the inequality is strict

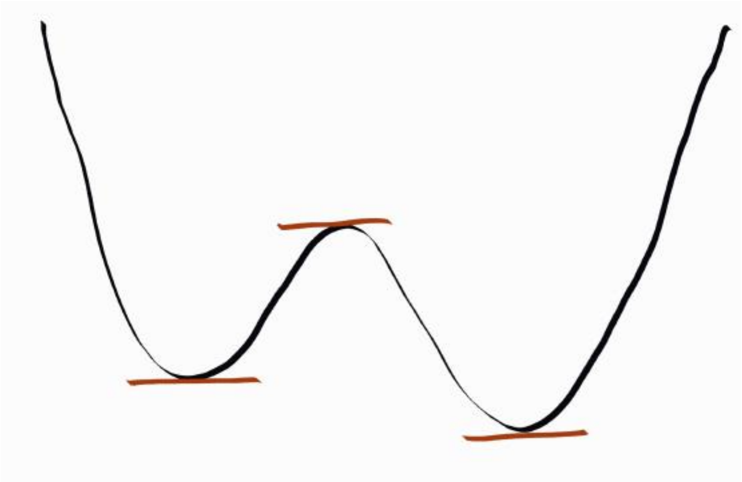
Constrained case

- a bit more involved
- hence, later in the lecture ☺

Mathematical Characterization of Optima

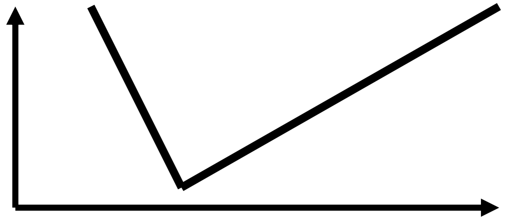
Objective: Derive general characterization of optima

Example: if $f: \mathbb{R} \to \mathbb{R}$ differentiable,
$f'(x) = 0$ at optimal points



- generalization to $f: \mathbb{R}^n \to \mathbb{R}$?
- generalization to constrained problems?

Remark: notion of optimum independent of notion of derivability

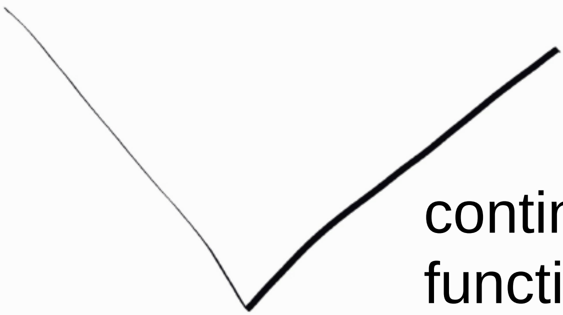


optima of such function can be easily approached by certain type of methods

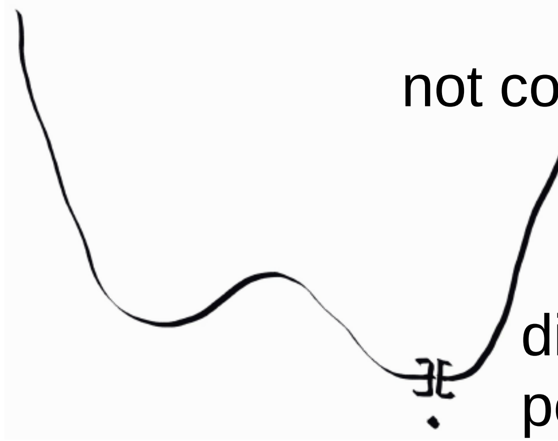
Reminder: Continuity of a Function

$f: (V, || \ ||_V) \longrightarrow (W, || \ ||_W)$ is continuous in $x \in V$ if

$\forall \epsilon > 0, \exists \eta > 0$ such that $\forall y \in V$: $||x - y||_V \leq \eta$; $||f(x) - f(y)||_W \leq \epsilon$



continuous function



not continuous

discontinuity point

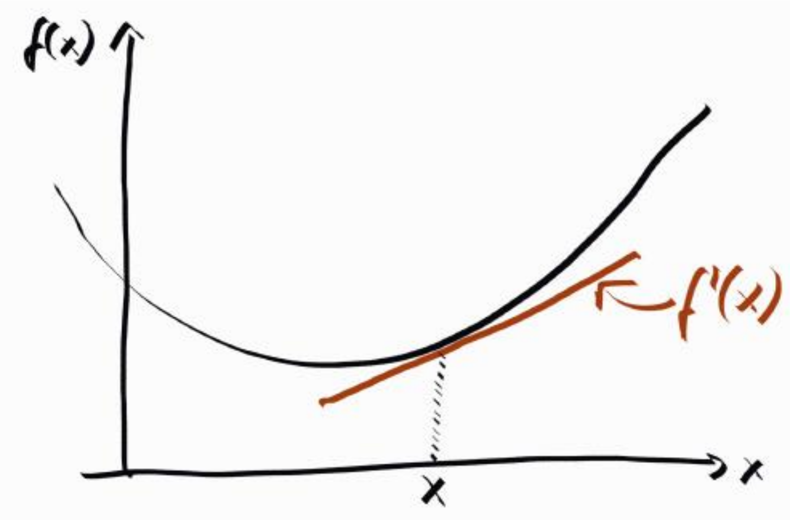
Reminder: Differentiability in 1D (n=1)

$f: \mathbb{R} \to \mathbb{R}$ is differentiable in $x \in \mathbb{R}$ if

$$\lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \text{ exists, } h \in \mathbb{R}$$

Notation:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$



The derivative corresponds to the slope of the tangent in x.

Reminder: Differentiability in 1D (n=1)

Taylor Formula (Order 1)

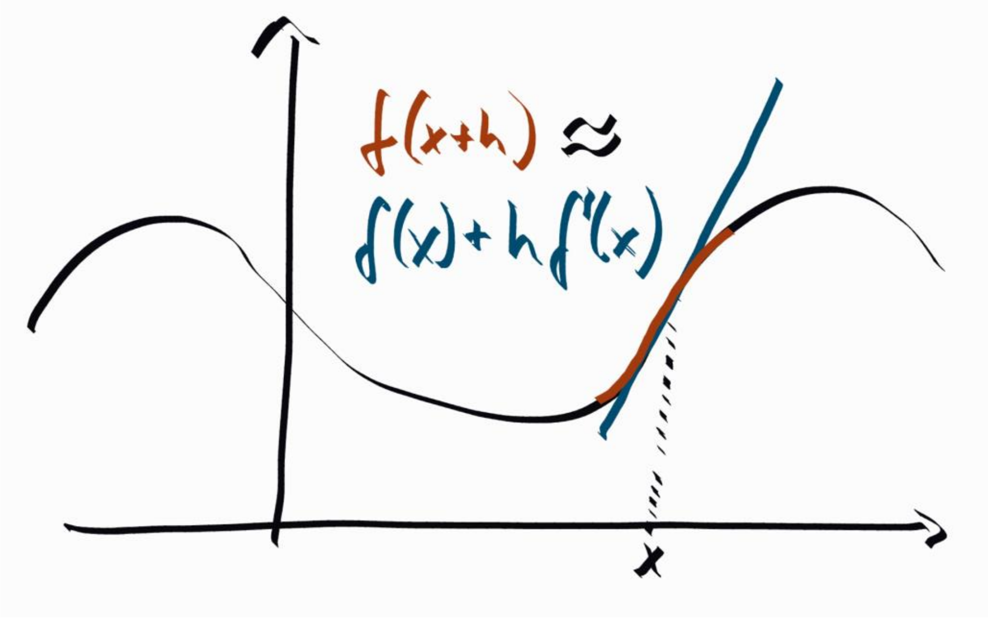
If f is differentiable in x then

$$f(x+h) = f(x) + f'(x)h + o(||h||)$$

i.e. for h small enough, $h \longmapsto f(x+h)$ is approximated by $h \longmapsto f(x) + f'(x)h$

$h \longmapsto f(x) + f'(x)h$ is called a first order approximation of $f(x+h)$

Geometrically:



The notion of derivative of a function defined on $\mathbb{R}^n$ is generalized via this idea of a linear approximation of $f(x+h)$ for h small enough.

How to generalize this to arbitrary dimension?

Gradient Definition Via Partial Derivatives

- In $(\mathbb{R}^n, ||\ \ ||_2)$ where $||\boldsymbol{x}||_2 = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$ is the Euclidean norm deriving from the scalar product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \ \boldsymbol{x}^T \boldsymbol{y}$

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

- Reminder: partial derivative in x_0

$$\mathrm{f_i}\colon\ y \to\ f\left(x_0^1, \dots, x_0^{i-1}, y, x_0^{i+1}, \dots, x_0^n\right)$$

$$\frac{\partial f}{\partial x_i}(x_0) = f_i{}'(x_0)$$

Exercise:

Compute the gradients of

a) $f(x) = x_1$ with $x \in \mathbb{R}^n$

b) $f(x) = a^T x$ with $\mathrm{a}, x \in \mathbb{R}^n$

c) $f(x) = x^T x$ $(= ||\mathrm{x}||^2)$ with $x \in \mathbb{R}^n$

Exercise: Gradients

Exercise:

Compute the gradients of

a) $f(x) = x_1$ with $x \in \mathbb{R}^n$

b) $f(x) = a^T x$ with $\mathrm{a}, x \in \mathbb{R}^n$

c) $f(x) = x^T x$ $(= ||\mathrm{x}||^2)$ with $x \in \mathbb{R}^n$

Some more examples:

- in $\mathbb{R}^n$, if $f(\boldsymbol{x}) = \boldsymbol{x}^T A \boldsymbol{x}$, then $\nabla f(\boldsymbol{x}) = (A + A^T)\boldsymbol{x}$
- in $\mathbb{R}$, $\nabla f(\boldsymbol{x}) = f'(\boldsymbol{x})$

Gradient: Geometrical Interpretation

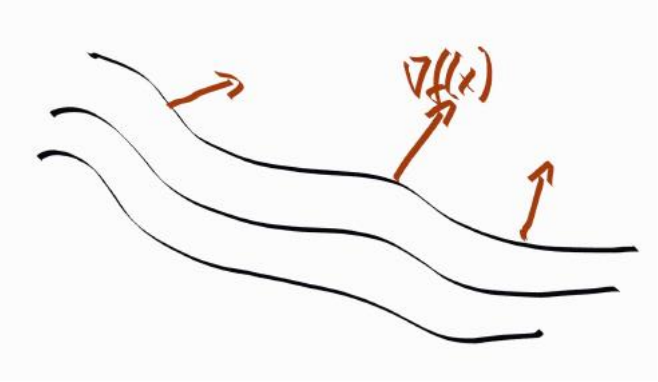
Exercise:

Let $L_c = \{\boldsymbol{x} \in \mathbb{R}^n \mid f(\boldsymbol{x}) = c\}$ be again a level set of a function $f(\boldsymbol{x})$. Let $\boldsymbol{x}_0 \in L_c \neq \emptyset$.

Compute the level sets for $f_1(\boldsymbol{x}) = \boldsymbol{a}^T\boldsymbol{x}$ and $f_2(\boldsymbol{x}) = ||\boldsymbol{x}||^2$ and the gradient in a chosen point x_0 and observe that $\nabla f(\boldsymbol{x_0})$ is ***orthogonal*** to the level set in x_0.

Again: if this seems too difficult, do it for two variables (and a concrete $\boldsymbol{a} \in \mathbb{R}^2$) and draw the level sets and the gradients.

More generally, the gradient of a differentiable function is orthogonal to its level sets.



Taylor Formula – Order One

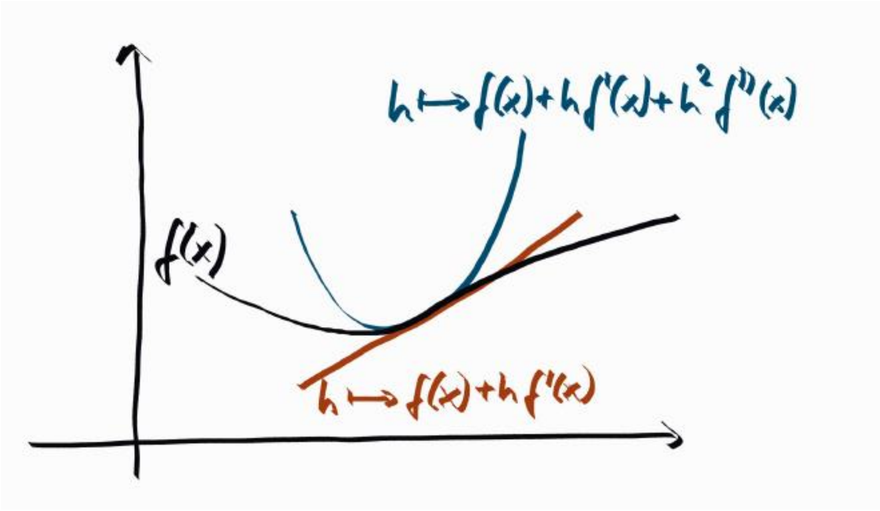
$$f(\boldsymbol{x}+\boldsymbol{h}) = f(\boldsymbol{x}) + \left(\nabla f(\boldsymbol{x})\right)^T \boldsymbol{h} + o(||\boldsymbol{h}||)$$

Reminder: Second Order Derivability in 1D

- Let $f: \mathbb{R} \to \mathbb{R}$ be a differentiable function and let $f': x \to f'(x)$ be its derivative.
- If f' is differentiable in x, then we denote its derivative as $f''(x)$
- $f''(x)$ is called the *second order derivative* of f.

Taylor Formula: Second Order Derivative

- If $f: \mathbb{R} \to \mathbb{R}$ is two times differentiable then
$$f(x+h) = f(x) + f'(x)h + f''(x)h^2 + o(||h||^2)$$
i.e. for h small enough, $h \to f(x) + hf'(x) + h^2 f''(x)$ approximates $h + f(x+h)$
- $h \to f(x) + hf'(x) + h^2 f''(x)$ is a quadratic approximation (or order 2) of f in a neighborhood of x



- The second derivative of $f: \mathbb{R} \to \mathbb{R}$ generalizes naturally to larger dimension.

Hessian Matrix

In $(\mathbb{R}^n, \langle x, y \rangle = x^T y)$, $\nabla^2 f(x)$ is represented by a symmetric matrix called the Hessian matrix. It can be computed as

$$\nabla^2(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Exercise on Hessian Matrix

Exercise:

Let $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\, \boldsymbol{x}$, $\boldsymbol{x} \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$.

Compute the Hessian matrix of f.

If it is too complex, consider $f: \begin{cases} \mathbb{R}^2 \to \mathbb{R} \\ \boldsymbol{x} \to \frac{1}{2}\boldsymbol{x}^T A\, \boldsymbol{x} \end{cases}$ with $A = \begin{pmatrix} 9 & 0 \\ 0 & 1 \end{pmatrix}$

Second Order Differentiability in $\mathbb{R}^n$

Taylor Formula – Order Two

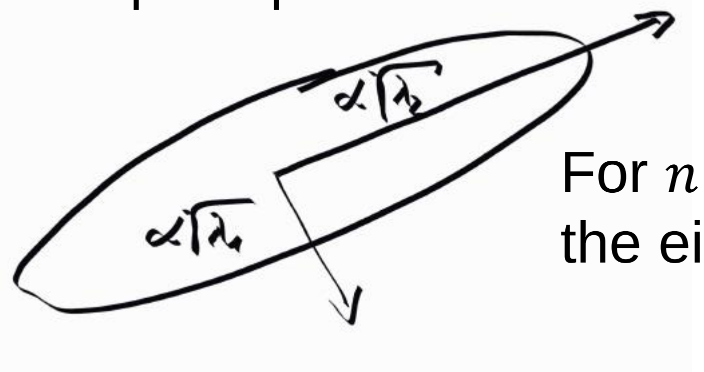
$$f(\boldsymbol{x}+\boldsymbol{h}) = f(\boldsymbol{x}) + \left(\nabla f(\boldsymbol{x})\right)^T \boldsymbol{h} + \frac{1}{2}\boldsymbol{h}^T\left(\nabla^2 f(\boldsymbol{x})\right) \boldsymbol{h} + o(||\boldsymbol{h}||^2)$$

Back to Ill-Conditioned Problems

We have seen that for a convex quadratic function

$f(x) = \frac{1}{2}(x - x_0)^T A(x - x_0) + b$ of $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$, A SPD, $b \in \mathbb{R}^n$:

1) The level sets are ellipsoids. The eigenvalues of A determine the lengths of the principle axes of the ellipsoid.



For $n = 2$, let λ_1, λ_2 be the eigenvalues of A.

2) The Hessian matrix of f equals to A.

Ill-conditioned convex quadratic problems are problems with large ratio between largest and smallest eigenvalue of A which means large ratio between longest and shortest axis of ellipsoid.

This corresponds to having an ill-conditioned Hessian matrix.

Gradient Direction Vs. Newton Direction

Gradient direction: $\nabla f(\boldsymbol{x})$

Newton direction: $\left(H(\boldsymbol{x})\right)^{-1} \cdot \nabla f(\boldsymbol{x})$

with $H(\boldsymbol{x}) = \nabla^2 f(\boldsymbol{x})$ being the Hessian at $\boldsymbol{x}$

Exercise:

Let again $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\, \boldsymbol{x}$, $\boldsymbol{x} \in \mathbb{R}^2$, $A = \begin{pmatrix} 9 & 0 \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{2\times 2}$.

Plot the gradient and Newton direction of f in a point $x \in \mathbb{R}^n$ of your choice (which should not be on a coordinate axis) into the same plot with the level sets, we created before.

Gradient Direction Vs. Newton Direction

Gradient direction: $\nabla f(\boldsymbol{x})$

Newton direction: $\left(H(\boldsymbol{x})\right)^{-1} \cdot \nabla f(\boldsymbol{x})$

with $H(\boldsymbol{x}) = \nabla^2 f(\boldsymbol{x})$ being the Hessian at $\boldsymbol{x}$

Exercise:

Let again $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\, \boldsymbol{x}$, $\boldsymbol{x} \in \mathbb{R}^2$, $A = \begin{pmatrix} 9 & 0 \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{2\times 2}$.

Plot the gradient and Newton direction of f in a point $x \in \mathbb{R}^n$ of your choice (which should not be on a coordinate axis) into the same plot with the level sets, we created before.

- remind level sets: axis-parallel ellipsoids, axis-ratio=3
- remind gradient: $A\boldsymbol{x}$
- remind Hessian: A

Conclusions

I hope it became clear...

…what kind of optimization problems we are interested in

…what are level sets and how to plot them

…what difficulties a problem can have

…what the gradient is
(and that it is generally orthogonal to the level sets)

…what the Hessian is

…which basic optimality conditions exist (1^{st} and 2^{nd} order)