Overview on Optimization Algorithms

Markus A. Köbis

NTNU, TMA4180, Optimering I, Spring 2021

2021-02-26

A Review Session

Idea

- OPTION A - a live session with a presentation highlighting the ´big picture' of what we have learned so far

What we don't do today

- Learn new material ;-)
- Go through proofs

Questions are welcome!

Talking about the 'Big Picture'(I)

Topics of Today:

NTNU Studier Forskning og innovasjon Studentliv Om NTNU

Vurderingsform	Vekting	Varighet	Hjelpemidler
Arbeider	30/100		
Hjemmeeksamen	70/100	4 timer	

Faglig innhold

Første og andre ordens nødvendige og tilstrekkelige (Karush-Kuhn-Tucker) optimalitetsbetingelser for ubegrensede og begrensede optimeringsproblemer i endelig-dimensjonale vektorrom. Grunnleggende konveksanalyse og Lagranges dualitetsteori og deres anvendelser for optimeringsproblemer og algoritmer. Oversikt over moderne optimeringsteknikker og algoritmer for glatte problemer (inklusive line-search/trust-region, kvasi-Newton, indre punkt og aktive sett metoder, SQP og augmented Lagrangian teknikker). Grunnleggende metoder for derivat-fri og ikke-glatte optimeringsproblem.

Læringsutbytte

Studenten som møter læringsmålene for kurset skal kunne:
(i) vurdere eksistens og entydighet av løsninger til et gitt optimeringsproblem;
(ii) validere konveksitet av funksjoner, sett, og optimeringsproblemer;
(iii) utlede nødvendige og tilstrekkelige optimalitetsbetingelser for et gitt optimeringsproblem;
(iv) løse små optimeringsproblemer analytisk;
(v) forklare de underliggende prinsipper og begrensninger av moderne teknikker og algoritmer for optimering;
(vi) anslå konvergenshastigheten og kompleksitetskrav i ulike optimeringsalgoritmer;
(vii) implementere optimeringsalgoritmer på en datamaskin;
(viii) bruke optimeringsalgoritmer for å løse modellproblemer i ingeniør- og realfag.

Læringsformer og aktiviteter

Talking about the 'Big Picture'(II)

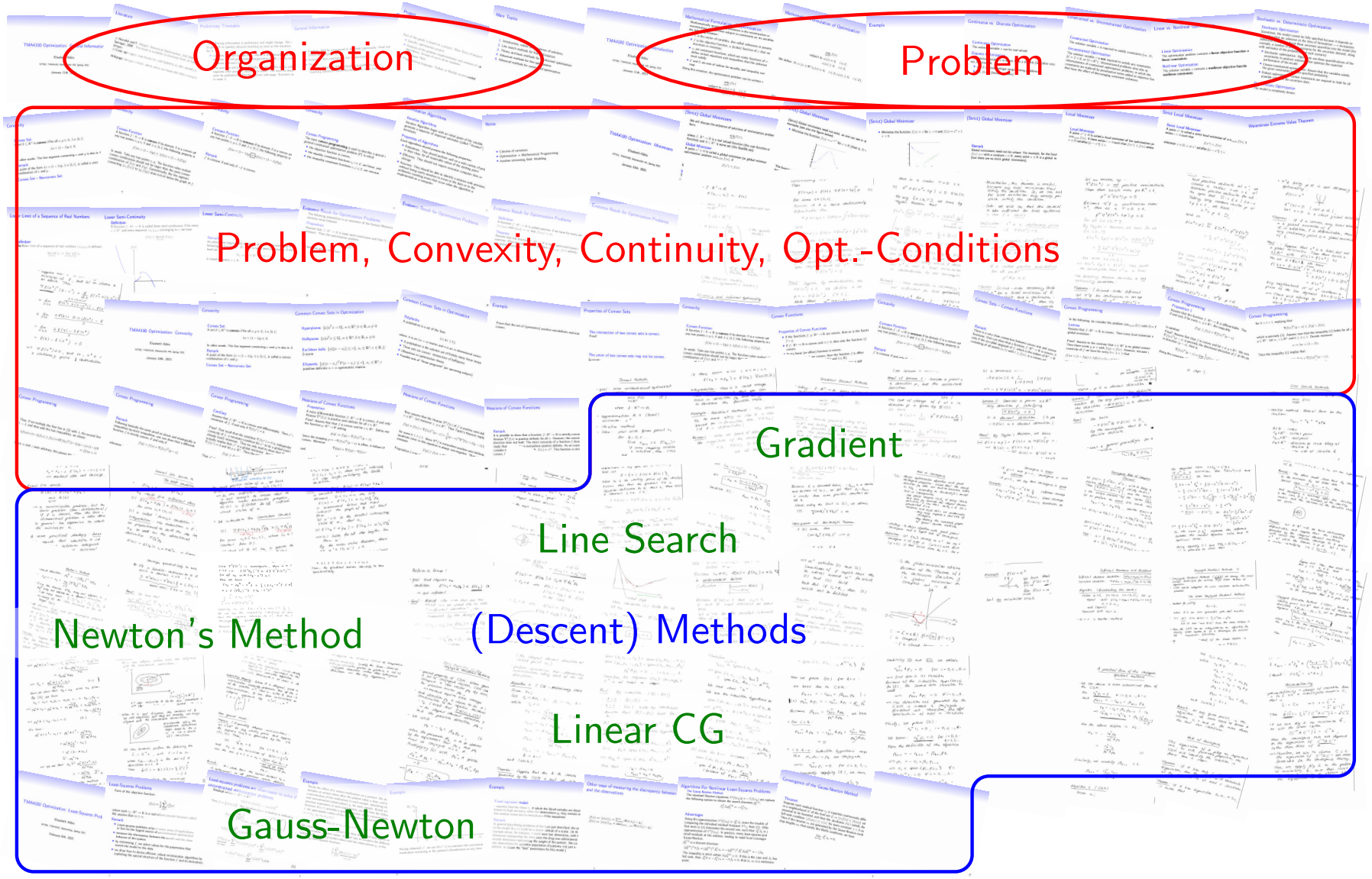
Topics of Today:

Recap of Theory and Methods for **Unconstrained** Optimization Problems

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

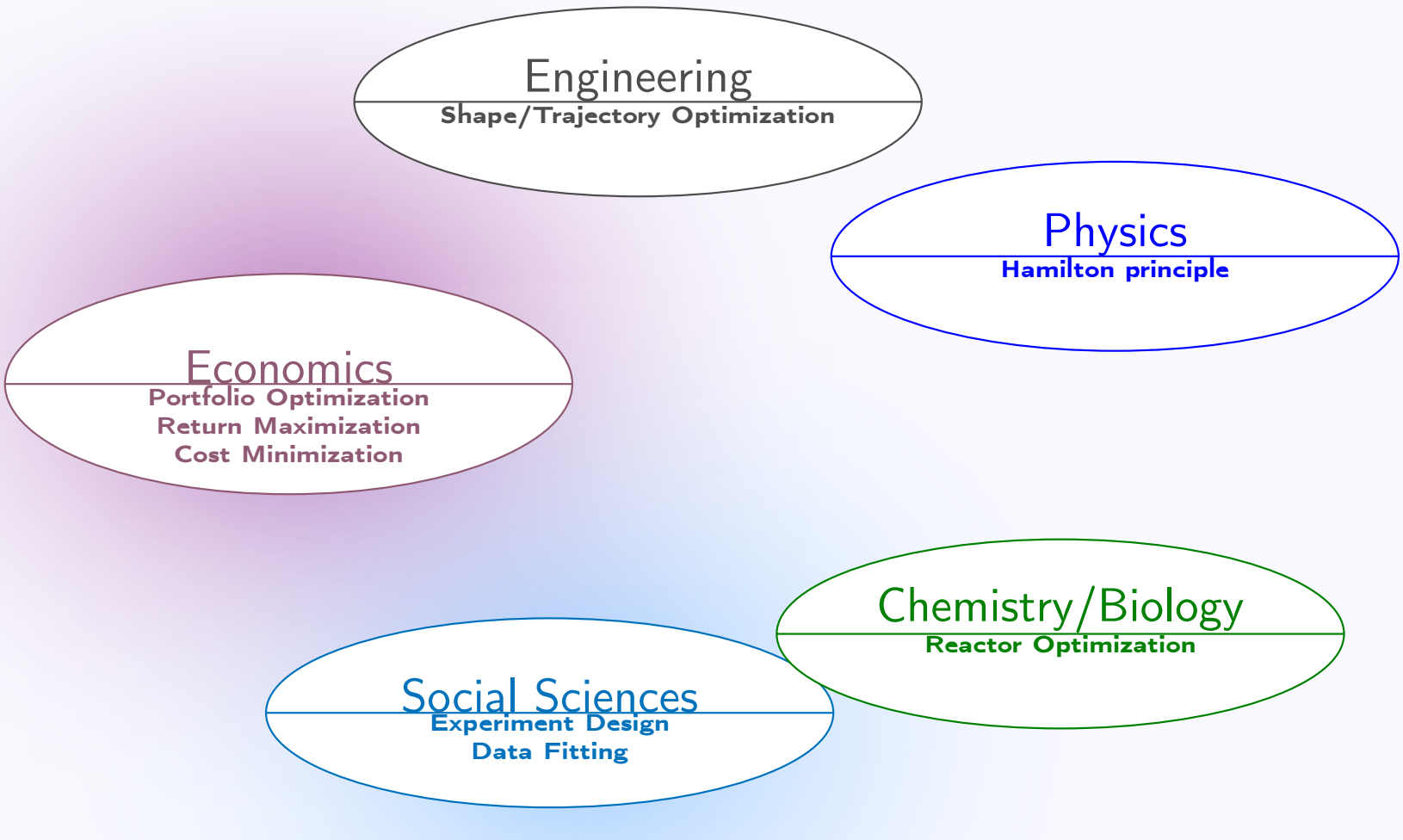
- (Practical) Problems and Challenges
- Theoretical Considerations
- **Algorithms**

Talking about the ‘Big Picture’(III)



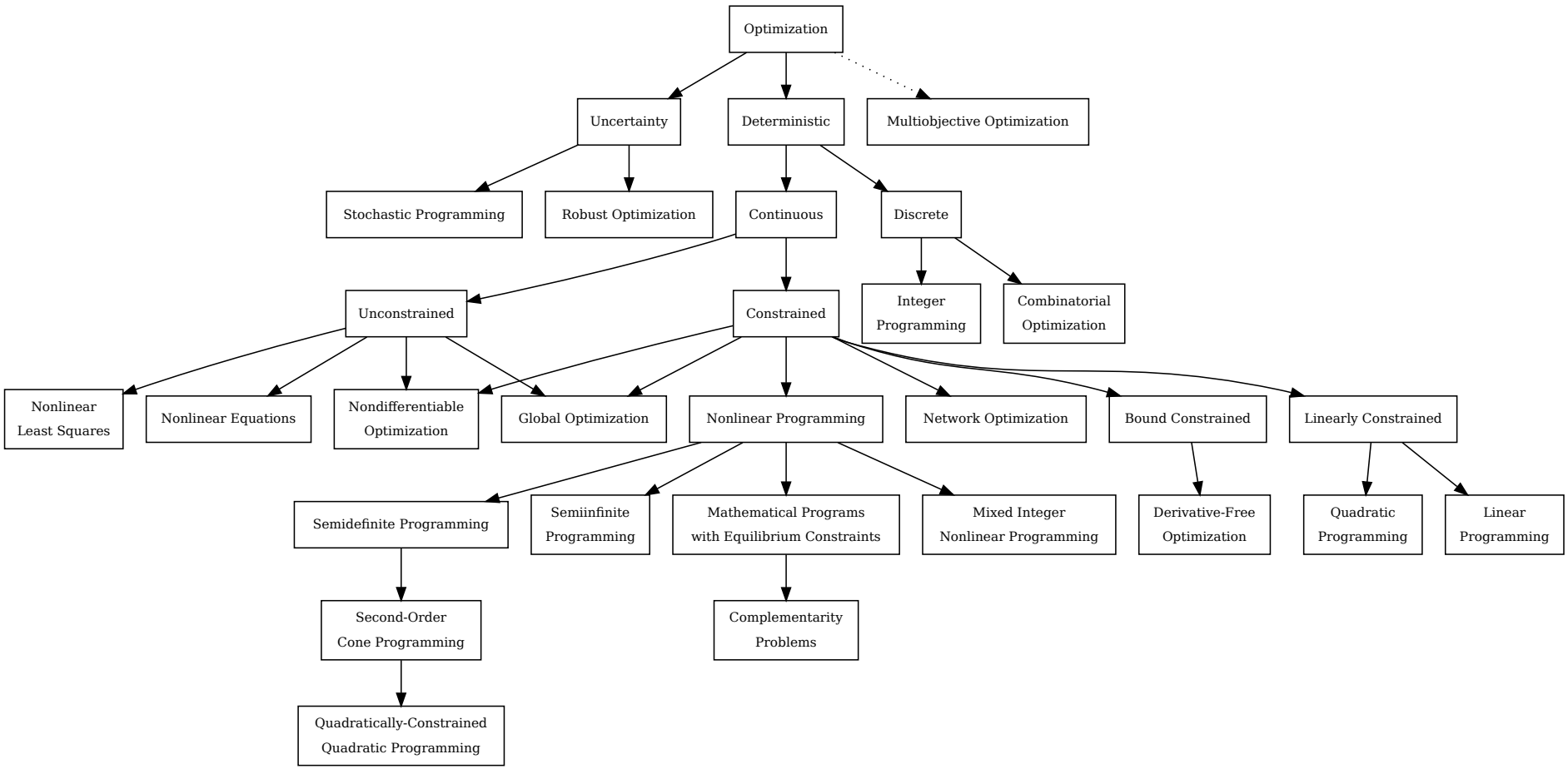
Problems and Challenges (I)

Optimization Problems in Science and Engineering



Taxonomy of Optimization Problems

(Not exhaustive)



Source: https://neos-guide.org/content/optimization-taxonomy

Quality of Algorithms

Properties of ´good' algorithms (Why isn't there a best one?)

- Accuracy
- Efficiency (also: Scalability)
  - Speed
    - Total execution time (avoid NP if possible, parallel vs. sequential)
    - Alternative: approximation quality after fixed time
    - Number of computer operations
      - Core function/ Linear algebra/GPU-routine-calls
      - Floating point operations (plus, times, roots, exp./trig, ...)
      - Bit operations
      - Number of $f/\nabla f/\nabla^2 f$ calls
  - Memory
- Robustness
  - Reliably find all solutions for a variety of problems
  - Feedback if algorithm unsuccessful, reproducibility
  - Provide additional information on problem structure
- User-friendliness, ability to include expert knowledge, interactivity
- Easily extendable and maintainable
- Cheap, trusted, supported, ...

Building the ´Big Picture'

Mathematical Foundations $\leftrightarrow$ Algorithms

When Do Minimizers Exist?

Weierstrass' theorem

Weierstrass Theorem

A **continuous** function on a **compact** domain attains its minimum.

- In practice: Very often enough
- Can one weaken the ´continuity' requirement?
  e.g. complicated functions in technical applications
- Can one weaken the ´compactness' requirement?
  e.g. optimal control/problems in mathematical physics

Karl Weierstrass, 1815–1897

Source: Public Domain, https://commons.wikimedia.org/w/index.php?curid=324146

Convexity

Convex Set

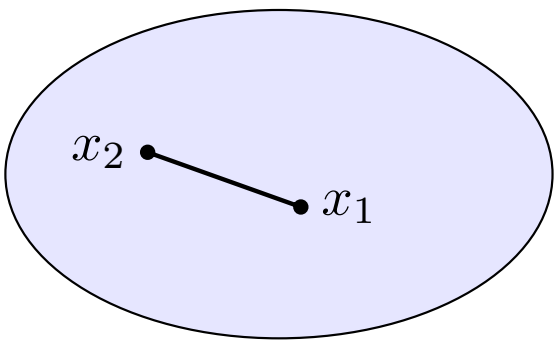
A set $S \subseteq \mathbb{R}^n$ is **convex** if for all $x_1, x_2 \in S$, $\lambda \in [0, 1]$

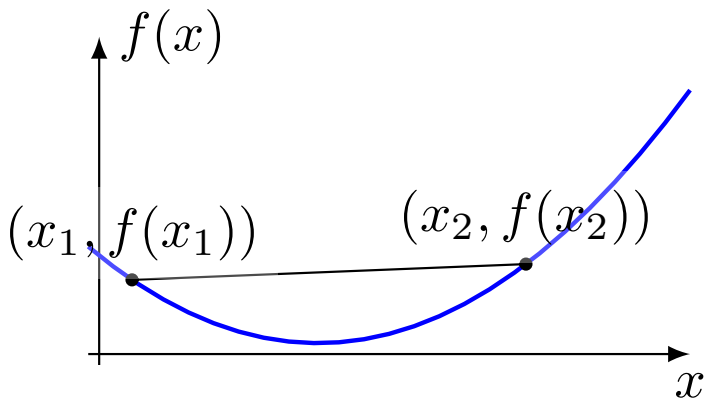
$$\underbrace{\lambda x_1 + (1 - \lambda) x_2}_{\text{convex combination}} \in S.$$

Convex Function

A function $f : S \to \mathbb{R}$ is **convex** if its domain S is a convex set and if for any two points $x_1, x_2 \in S$ and $\lambda \in [0, 1]$ the following property is satisfied:

$$f(\lambda x_1 + (1 - \lambda) x_2) \leq \lambda f(x_1) + (1 - \lambda) f(x_2).$$



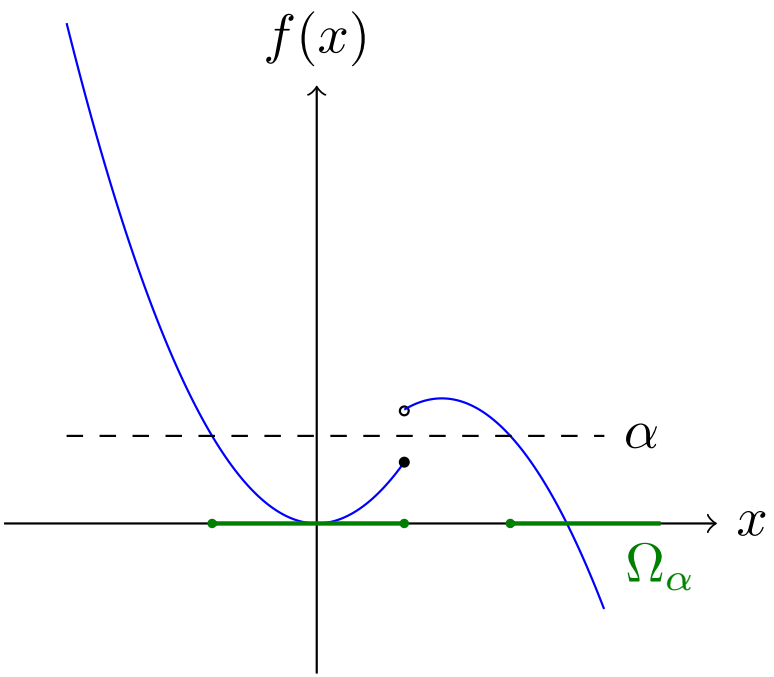


(Lower Semi-) Continuity

Definition

A function $f\colon \mathbb{R}^n \to \mathbb{R}$ is called *lower semi-continuous*, if for every $x \in \mathbb{R}^n$ and every sequence $(x_k)_{k\in\mathbb{N}}$ converging to x we have

$$f(x) \leq \liminf_{k\to\infty} f(x_k).$$



Remark

An alternative (equivalent) definition of lower semi-continuity is the following: A function $f\colon \mathbb{R}^n \to \mathbb{R}$ is lower semi-continuous, if the *lower level set*

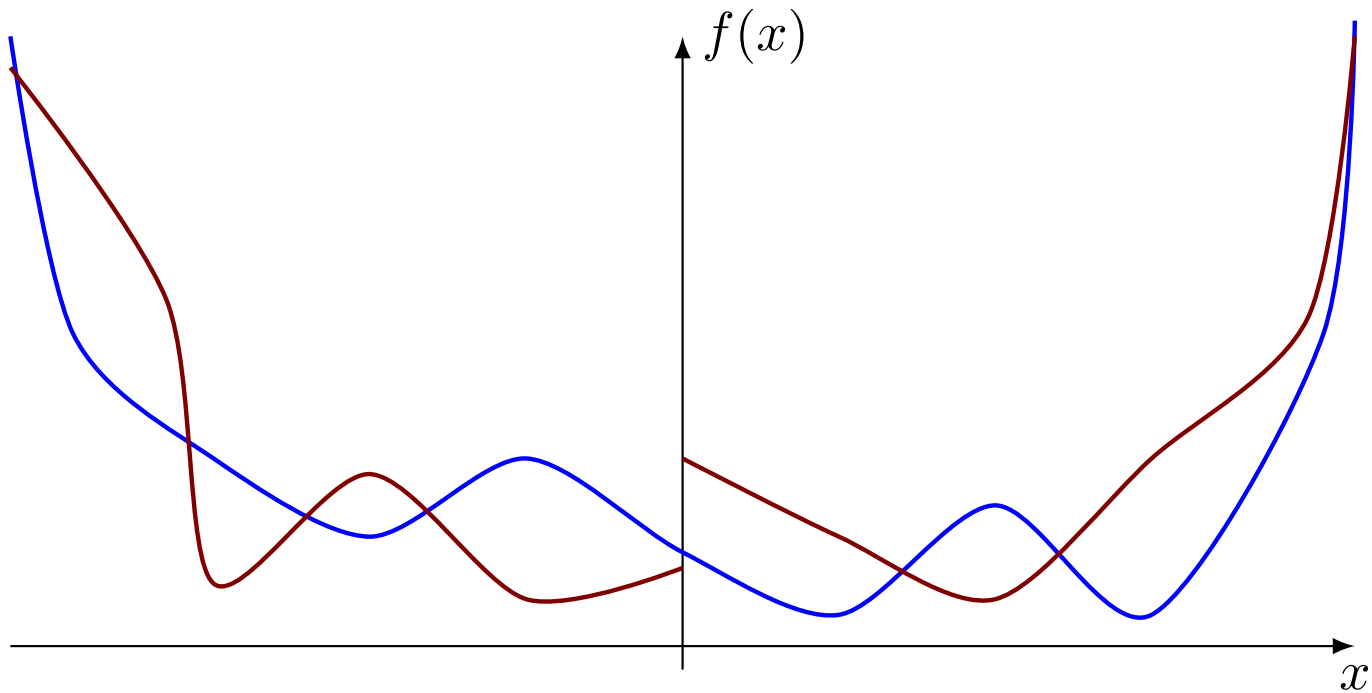
$$\Omega_\alpha(f) := \{x \in \mathbb{R}^n : f(x) \leq \alpha\}$$

is closed for every $\alpha \in \mathbb{R}$.

Coercivity

Definition

A function $f\colon \mathbb{R}^n \to \mathbb{R}$ is called coercive, if we have for every sequence $(x_k)_{k\in\mathbb{N}}$ with $\|x_k\| \to \infty$ that $f(x_k) \to \infty$.



Weierstrass revisited

Proposition

Assume that $f\colon \mathbb{R}^n \to \mathbb{R}$ is lower semi-continuous and that $\Omega \subset \mathbb{R}^n$ is compact. Then the optimization problem

$$\min_{x \in \Omega} f(x)$$

has at least one global minimizer.

Theorem

Assume that the function $f\colon \mathbb{R}^n \to \mathbb{R}$ is lower semi-continuous and coercive. Then the optimization problem $\min_{x \in \mathbb{R}^n} f(x)$ admits at least one global minimizer.

The Role of Convexity

Theorem

If $f\colon \mathbb{R}^n \to \mathbb{R}$ is convex, any local minimizer is a global minimizer. (This implies that optimality conditions (see below) can be used for global optimization.)

Designing Algorithms

Optimization Problem:

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

How to proceed?

Design computer method to find good points in $\mathbb{R}^n$ until you are finished

(a) Can the search be based on existing mathematical tools? (e.g. Solve (non-)linear systems, feasibility problem solvers, simulation methods etc.)

(b) How can you know that the algorithm is 'finished'?

$\rightarrow$ Optimality Conditions

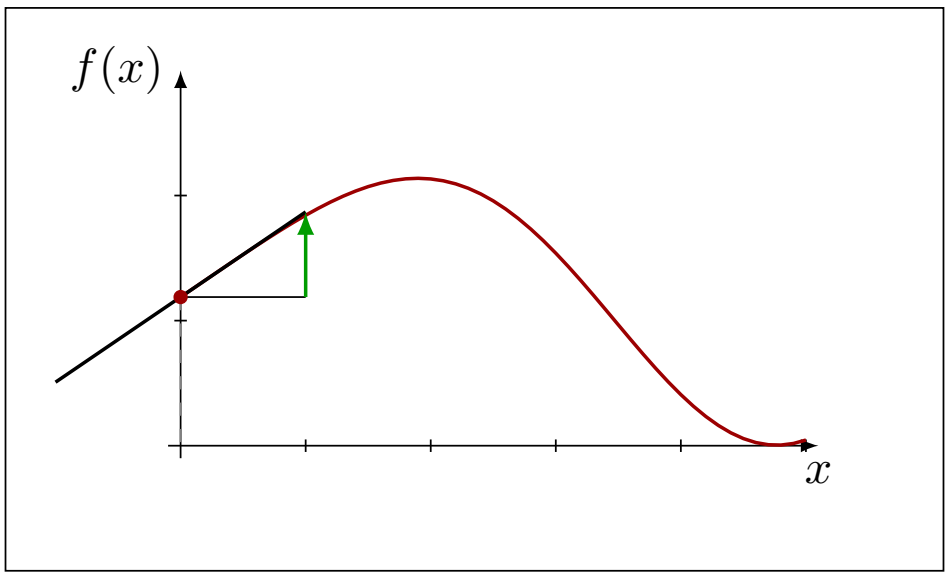
Optimality Conditions (I)

First Order Necessary Condition

Let $x^* \in \mathbb{R}^n$ be a *local* minimizer, and let $f\colon \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable in an open neighborhood of x^*. Then

$$\nabla f(x^*) = \mathbf{0}\,.$$

This condition classifies x^* as a stationary point



Optimality Conditions (II)

Second Order Necessary Condition

Let $x^* \in \mathbb{R}^n$ be a local minimizer of $f\colon \mathbb{R}^n \to \mathbb{R}$. If $\nabla^2 f$ exists and is continuous in an open neighborhood of x^*, then $\nabla f(x^*) = \mathbf{0}$ and $\nabla^2 f(x^*)$ is positive semi-definite.

Second Order Sufficient Condition

Let $\nabla^2 f$ be continuous in an open neighborhood of $x^* \in \mathbb{R}^n$, $\nabla f(x^*) = \mathbf{0}$ and let $\nabla^2 f(x^*)$ be positive definite.Then x^* is a strict local minimizer.

Geometry/Practice

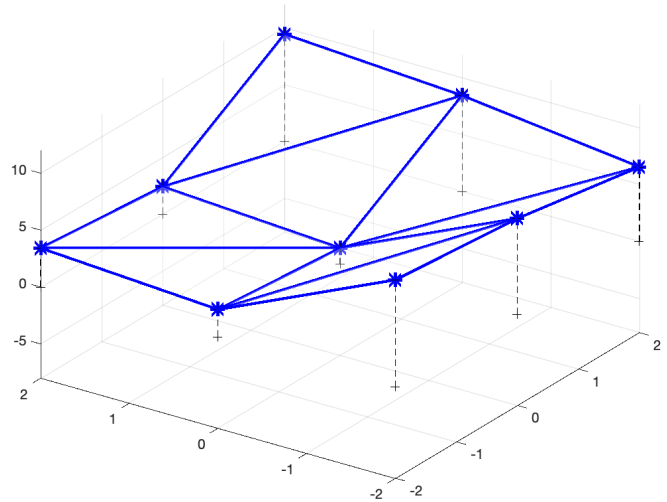
- Properties of a **Quadratic Approximation on** f
- In (large-scale) algorithms often not/only approximately used.

Designing Algorithms (II)

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

Algorithm: Computerized function (Input $\rightarrow$ output scheme) with limited internal memory, calculation power and precision.

Picture: Point cloud



Designing Algorithms (III)

Versatile methods for 'any' optimization problem

1. First Attempt: **Grid Search**: Evaluate f along a predefined grid an choose some good grids to refine this further.
   'Curse of Dimension': Number of grid points grows exponentially with dimension n of x.
2. Improvement (a): Solve a series of 1d-problems using 'classical methods'
   How to define these?
3. Improvement (b): Algorithms based on Heuristics/Physics/Biology
   e.g. Nelder-Mead Simplex Method, Simulated Annealing, Evolutionary Algorithms
4. (Often) Better: **Iterative Procedure**: Choose a starting point $x_0 \in \mathbb{R}^n$ and search along a sequence

   $$x_0, x_1, x_2, \ldots, x_k, x_{k+1}, \ldots \to x^*$$

   with improved properties of x_{k+1} over x_k.

   Given the scarce information (f is just input→output), the only way to measure 'improved properties'is through function values → descent methods

scaling invariance, also important to account for easy extension and maintenance of code

Descent Methods (I)

Basic Idea of Descent Methods

1. Start with initial $x_0 \in \mathbb{R}^n$
2. **For** $k = 0, 1, 2, \ldots$
   Find x_{k+1} such that $f(x_{k+1}) < f(x_k)$
   If some stopping criterion is fulfilled: STOP

Output: $x^* = x_{k+1}$

How to find a better x? Define **the search direction**

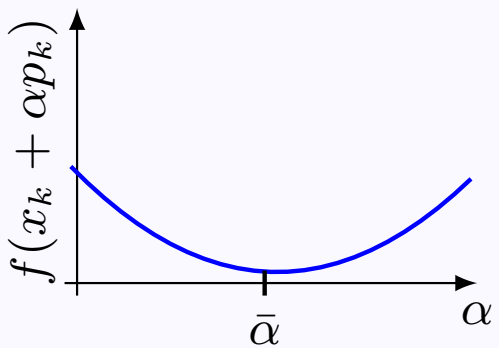
$$\lambda \cdot p_k := x_{k+1} - x_k \qquad \Leftrightarrow \qquad x_{k+1} = x_k + p_k$$

When is p_k good?

Definition

For a given $x_k \in \mathbb{R}^n$, a direction (nonzero vector (of undefined length)) $p_k \in \mathbb{R}^n$ is called a **descent direction** if there exists $\bar{\alpha} > 0$ such that

$$f(x_k + \alpha p_k) < f(x_k) \quad \forall \alpha \in (0, \bar{\alpha}).$$



Descent Methods (II)

Lemma: Characerizing descent directions for differentiable f

Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable.

1. Any direction p satisfying

$$\nabla f(x)^\top \cdot p < 0$$

   is a descent direction.
2. The negative gradient $-\nabla f(x)$ is a descent direction (if not zero)
3. For any positive definite matrix $B \in \mathbb{R}^{n\times n}$, the direction

$$-B \cdot \nabla f(x)$$

   is a descent direction (if x is not stationary).

Line Search (I)

In general form of the descent algorithm:

$$x_{k+1} := x_k + \alpha_k \cdot p_k$$

What is a good α_k?

1. Constant α:
   - Good for some special algorithms and when problem structure is well-understood
   - α too large: 'overshooting' and method may diverge
   - α too small: Potentially many steps and slow convergence
2. Determining a perfect value, exact line search: $\alpha_k := \operatorname{argmin}_x \underbrace{\Phi_k(\alpha)}_{=f(x_k+\alpha\cdot p_k)}$
   - Good if Φ_k-minimization is easy.
3. **Inexact Line Searches**: Use heuristics to obtain adequate step lengths at justifiable cost

Line Search (I)

Backtracking

(Sub-) Algorithm: Backtracking

Problem: Find a good/quick approximation to the solution of $\min_\alpha \Phi_k(\alpha)$.

Parameter: $\hat{\alpha} \in \mathbb{R}$ as initial guess, reduction constant $\rho \in (0, 1)$

Set: $\alpha = \hat{\alpha}$

While: some stopping criterion for $\phi_k(\alpha)$ is not fulfilled:

set: $\alpha = \rho \cdot \alpha$.

In Practice

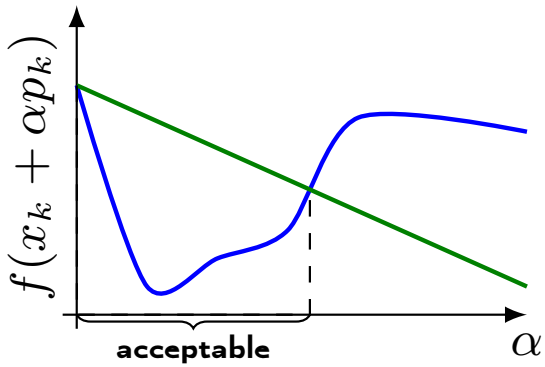
- Also set a maximum number of iterations
- Sometimes: Re-use α from previous run of algorithm
- Improvements based on 1d equation solvers (regula falsi, interpolation etc.)
- $\rho \in (0.4, 0.7)$

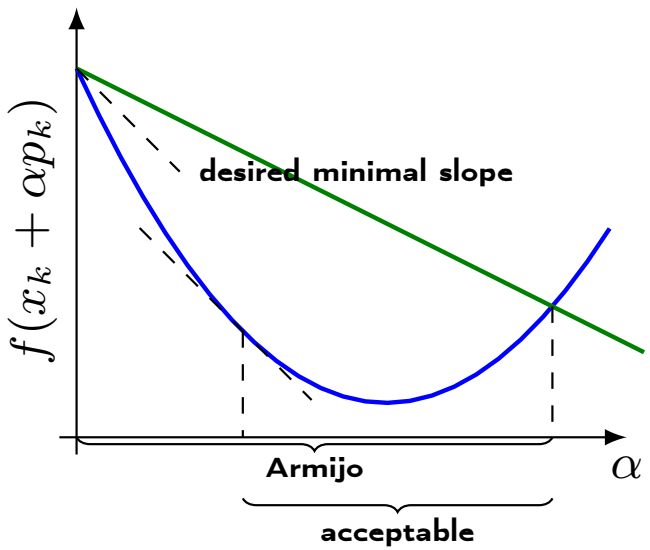
Line Search (II)

Armijo (Sufficient Decrease) Condition

Given $c_1 \in (0,1)$, we require

$$\underbrace{f(x_k + \alpha_k p_k)}_{\Phi_k(\alpha_k)} \leq \underbrace{f(x_k) + c_1 \cdot \alpha_k \cdot \nabla f(x_k)^\top \cdot p_k}_{\Phi_k(0)+\alpha_k \cdot c_1 \cdot \Phi_k'(0) =: l(\alpha_k)}$$





Wolfe (Curvature) Condition(s)

Given, furthermore, a constant $c_2 \in (c_1, 1)$, we require additionally

$$\underbrace{\nabla f(x_k + \alpha_k \cdot p_k)^\top \cdot p_k}_{\Phi_k'(\alpha_k)} \geq \underbrace{c_2 \cdot \nabla f(x_k)^\top \cdot p_k}_{c_2 \cdot \Phi_k'(0)}$$

Strong Wolfe condition: Also restrict positive slope

Existence of such α_k given if f smooth and bounded from below along the search line.

Line Search (III)

Fundamental Result for Line Search Methods

Theorem (Zoutendijk)

Consider any iteration of the form

$$x_{k+1} = x_k + \alpha_k \cdot p_k$$

where p_k is a descent direction and α_k satisfies the Wolfe conditions for $0 < c_1 < c_2 < 1$. Suppose that f is bounded below, differentiable with L-continuous gradient on any open set containing $\Omega_{f(x_0)} f$.
Then

$$\sum_{k \geq 0} (\cos \theta_k)^2 \cdot \|\nabla f(x_k)\|^2 < \infty \,,$$

where $\theta_k := \sphericalangle(-\nabla f(x_k), p_k)$

Result

$$\cos \theta_k \geq \delta > 0 \qquad \Rightarrow \qquad \lim_{k \to \infty} \|\nabla f(x_k)\| \to 0$$

Gradient Descent

Given a (continuously) differentiable objective $f\colon \mathbb{R}^n \to \mathbb{R}$.

We have information concerning the slope(s) of f.

Algorithm: Gradient Descent

Use a descent method (with or without line search).
Use the search direction

$$\forall k : p_k := -\nabla f(x_k)$$

Locally (i.e. $\bar{\alpha} \to 0$), this is the best possible choice

Gradient Descent: Convergence

Gradient Descent for quadratic test problem

$$f(x) := \frac{1}{2} x^\top \cdot Q \cdot x - b^\top \cdot x$$

with positive definite matrix Q.

Gradient Descent with Exact Line Search

Update formula

$$x_{k+1} = x_k - \frac{\nabla f(x_k)^\top \cdot \nabla f(x_k)}{\nabla f(x_k)^\top \cdot Q \cdot \nabla f(x_k)} \cdot \nabla f(x_k)$$

Theorem

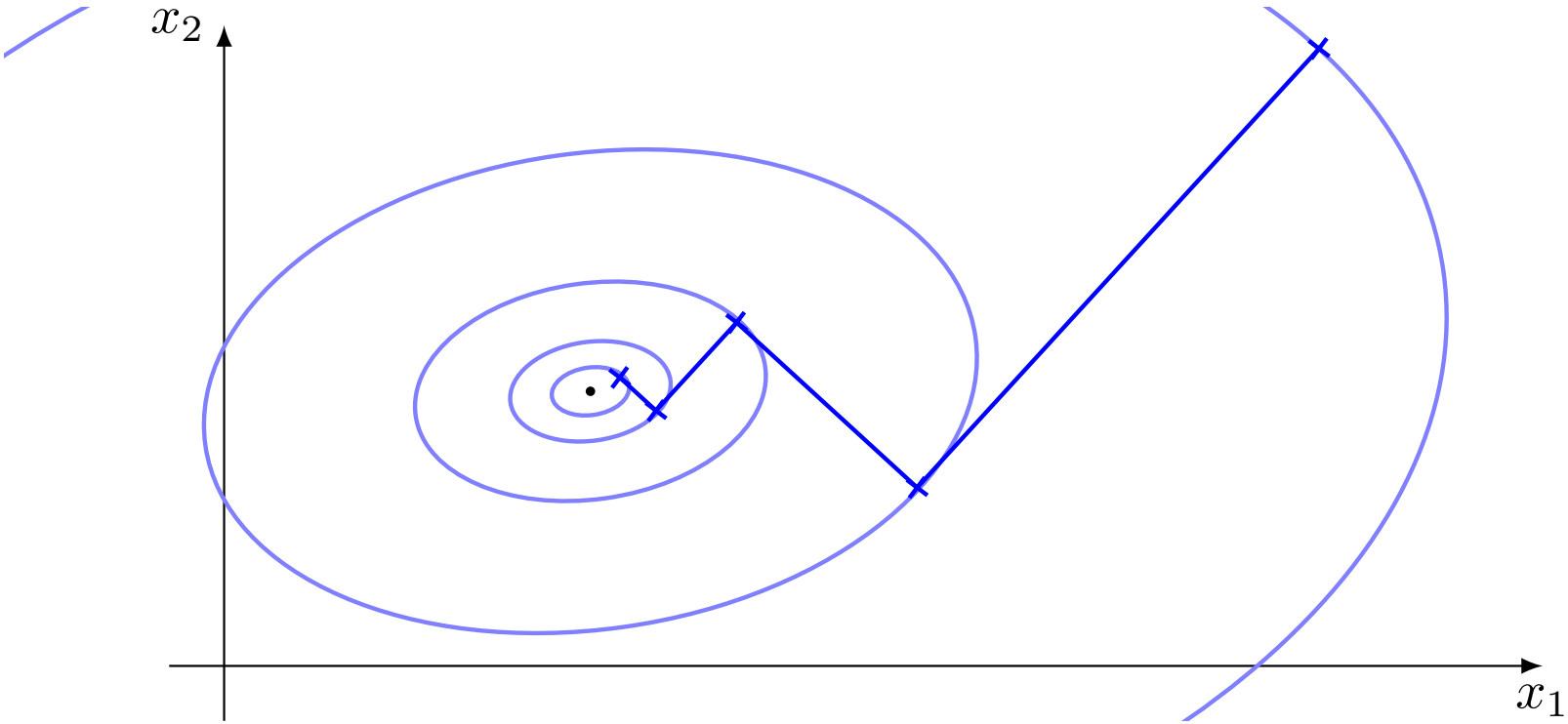
When gradient descent with exact line search is applied to f as above, then

$$\|x_{k+1} - x^*\|_Q^2 \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 \cdot \|x_k - x^*\|_Q^2 \ ,$$

where $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ are the eigenvalues of Q. (linear convergence)

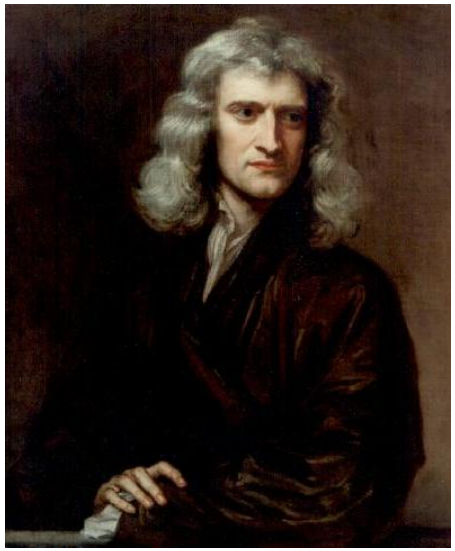
Similar result for nonlinear case. ($Q \leftrightarrow \nabla^2 f(x^*)$, $(.)^2 \leftrightarrow r^2$ with $\kappa(Q) < r < 1$)

Gradient Descent: Convergence



Perfect (convergence in one step) if all level curves equally spaced along all axes.

Newton's Method



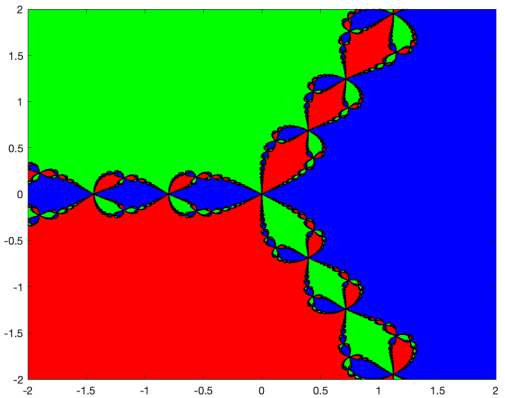
Sir Isaac Newton, 1643 - 1727

Motivation

- Instant method for $f(x) = x^\top \cdot Q \cdot x - b^\top \cdot x$ with s.p.d. matrix Q
- Solve $\nabla f = \mathbf{0}$ with the 'classical' Newton's method
- (Sequential) approximate of f by quadratic functions

Global Convergence Properties

- Oscillations
- Divergence
- Stuck in local minima
- Often not even a descent direction
- Not defined if $\nabla^2 f(x_k)$ singular



Source: By Godfrey Kneller - http://www.phys.uu.nl/ vgent/astrology/images/newton1689.jpg], Public Domain, https://commons.wikimedia.org/w/index.php?curid=146431

Newton's Method

Newton's Method

Apply the descent method algorithm with the update

$$x_{k+1} := x_k - \underbrace{\alpha_k}_{\substack{=1 \\ \text{or}\to 1}} \cdot \underbrace{\left(\nabla^2 f(x_k)\right)^{-1}}_{\substack{\text{not always} \\ \text{s.p.d.}}} \cdot \nabla f(x_k)$$

(Additional) Costs:

- approximation/calculation of Hessian $\nabla^2 f(x_k)$, storage thereof
- Inversion of $\nabla^2 f(x_k)$ (never invert(!))

Why all this trouble?

Theorem

If f is twice differentiable with a L.-continuous Hessian and $\nabla^2 f(x^*)$ is s.p.d. in a local minimum x^* and x_0 is **sufficiently close** to x^*, then Newton's method **convergences quadratically** in x and ∇f towards x^*.

Modifications of Newton's Method *different names in literature

Modified Newton's Method

Replace

$$\nabla^2 f(x_k) \rightsquigarrow \nabla^2 f(x_{0/m}) \qquad (m < k)$$

loose quadratic convergence, fewer evaluations/inversions

Simplified Newton

Replace Hessian by (good and easy to invert) approximation

$$\nabla^2 f(x_k) \rightsquigarrow Q_k \approx \nabla^2 f(x_k)$$

Damped Newton

Use a reduced step length $\alpha_k < 1$ in the Newton updates.
loose quadratic convergence, (often:) increase domain of convergence

In Practice

Newton's method only when Hessian cheap and/or to finalize iteration.

(Linear) Conjugated Gradient Method (I)

Back to quadratic problem:

$$f(x) = \frac{1}{2} \cdot x^\top \cdot A \cdot x - b^\top \cdot x$$

with a s.p.d. matrix $A \in \mathbb{R}^{n \times n}$.
Alternatively: Solve the (large) linear system $A \cdot x = b$. with the residual $r(x) := A \cdot x - b = \nabla f(x)$

Conjugate Direction Methods

Use a set of conjugate directions $\{p_0, p_1, \ldots, p_{n-1}\}$ (always lin. indep.) as search directions in the descent algorithm. i.e. $p_i^\top \cdot A \cdot p_j = 0$, $(i \neq j)$

- Exact line search like for the gradient method possible
- Convergence after n steps guaranteed
- After k steps: Minimization of f over a k-dimensional subspace (a so-called Krylov space ($\rightarrow$ Cayley-Hamilton))

(Linear) Conjugated Gradient Method (II)

CG method: Basic idea

- Use only previous vector p_{k-1} and residual $r(x_k)$ to construct next conjugate direction p_k

$p_k = -r(x_k) + \beta_k \cdot p_{k-1}$ (β_k uniquely det. from being conj. direction)

Linear CG method

$r_0 = Ax_0 - b$, $p_0 = -r_0$

While $r_k \neq \mathbf{0}$:

$$\alpha_k = -\frac{r_k^\top \cdot p_k}{p_k^\top \cdot A \cdot p_k}$$
$$x_{k+1} = x_k + \alpha_k \cdot p_k$$
$$r_{k+1} = A \cdot x_{k+1} - b$$
$$\beta_{k+1} = \frac{r_{k+1}^\top \cdot A \cdot p_k}{p_k^\top \cdot A \cdot p_k}$$
$$p_{k+1} = -r_{k+1} + \beta_{k+1} \cdot p_k$$

Convergence and Preconditioned CG

Theorem

If A has only r distinct eigenvalues, then the (linear) conjugated gradient method terminates at the solution after r steps.

Theorem

The (theoretical) convergence rate is similar to the gradient method

$$\|x_{k+1} - x^*\|_A^2 \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1}\right)^2 \cdot \|x_0 - x^*\|_A^2$$

PCG

Improve convergence behavior by linearly transforming the search space

$\hat{x} := C \cdot x$

$$\hat{f}(\hat{x}) = \frac{1}{2} \cdot \hat{x}^\top \cdot (C^{-\top} A C^{-1})\hat{x} - (C^{-\top} b)^\top \cdot \hat{x}$$

NB: Extension (and convergence analysis) to the nonlinear case (relatively) easy.

Least Squares Problems

Form of the objective function:

$$f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x)$$

where each $r_j : \mathbb{R}^n \to \mathbb{R}$ is a real valued smooth function called *residual*. We assume that $m \geq n$.

Remark

- Least-squares problems arise in many areas of applications, and may in fact be the largest source of unconstrained optimization problems
- measure the discrepancy between the model and the observed behavior of the system.
- by minimizing f, we select values for the parameters that best match the model to the data.
- we show how to devise efficient, robust minimization algorithms by exploiting the special structure of the function f and its derivatives.

Least Squares Problems (cont.)

Residual vector $r\colon\ \mathbb{R}^n \to \mathbb{R}^m$

$$r(x) := (r_1(x), \ldots, r_m(x))^\top$$

Then $f = \frac{1}{2}\|r(x)\|_2^2$. The *Jacobian* ($m \times n$-matrix) is defined as

$$J(x) := \left[\frac{\partial r_j}{\partial x_i}\right]_{j=1,\ldots,m,i=1,\ldots,n} = \begin{pmatrix} \nabla r_1(x)^\top \\ \vdots \\ \nabla r_m(x)^\top \end{pmatrix}.$$

$$\nabla f(x) = \sum_{j=1}^m r_j(x)\nabla r_j(x) = J(x)^\top r(x),$$

$$\underbrace{\nabla^2 f(x)}_{\text{“for free”}} = \sum_{j=1}^m \nabla r_j(x)\nabla r_j(x)^\top + \sum_{j=1}^m r_j(x)\nabla^2 r_j(x)^\top$$

$$= J(x)^\top J(x) + \sum_{j=1}^m r_j(x)\nabla^2 r_j(x). \tag{1}$$

Gauss Newton Methods

The Gauss-Newton Method

The standard Newton equations $\nabla^2 f(x_k)p = -\nabla f(x_k)$ are replaced by the following system to obtain the search direction p_k^{GN}:

$$J_k^\top J_k p_k^{\mathrm{GN}} = -J_k^\top r_k.$$

Advantages

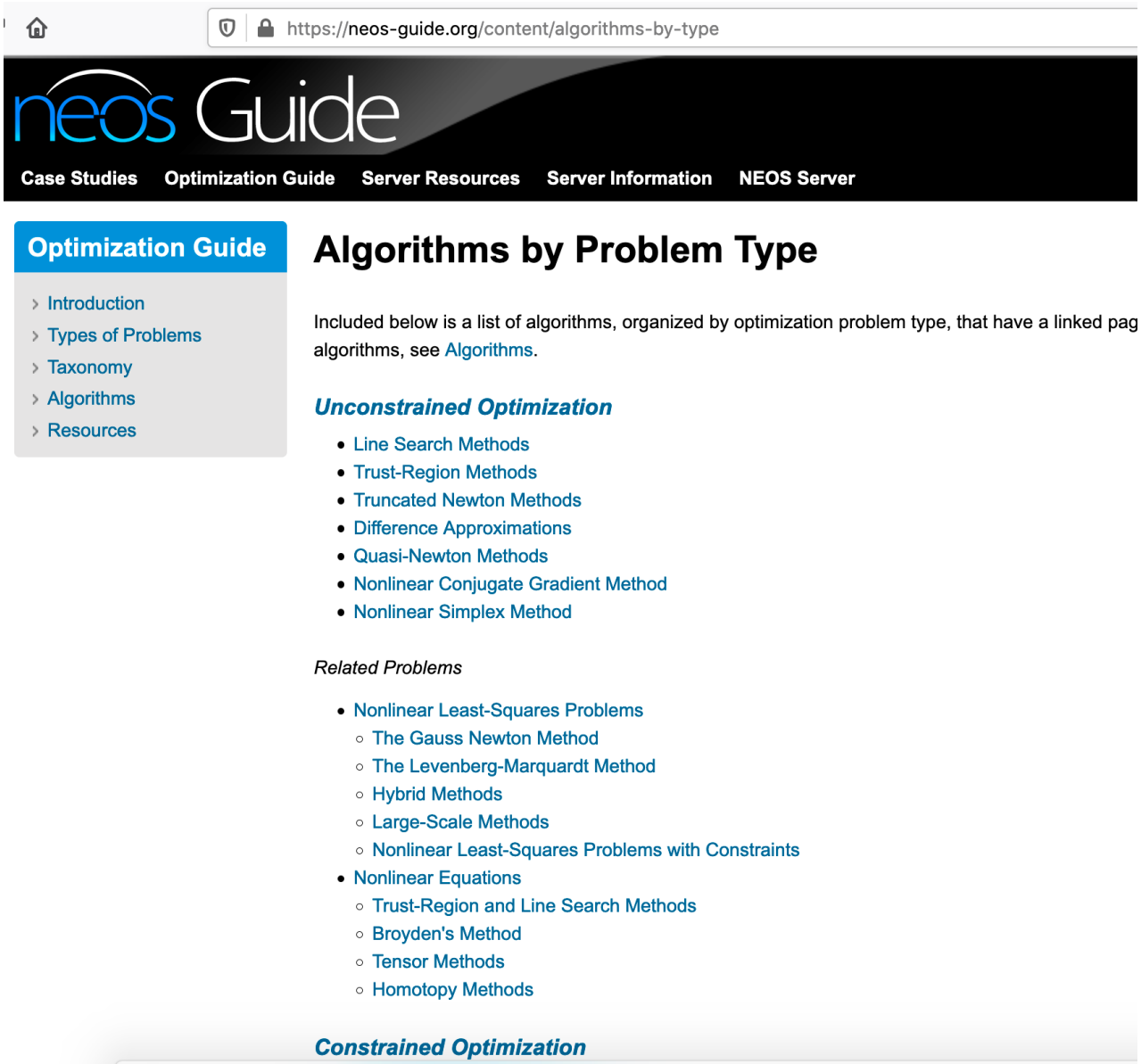
Using the approximation $\nabla^2 f(x_k) \approx J_k^T J_k$ saves the trouble of computing the individual residual Hessians $\nabla^2 r_j$ from (1). Often, the first term in (1) dominates the second one, such that $J_k^\top J_k$ is a good approximation of $\nabla^2 f(x_k)$. In practice, many least-squares problems have small residuals at the solution, leading to rapid local convergence of Gauss-Newton.

Convergence

In theory: (Linear) convergence under mild regularity and full-rank condition on system matrices.

In practice: Good global and fast local convergence

What Next?


https://neos-guide.org/content/algorithms-by-type
neos Guide
Case Studies
Optimization Guide
Server Resources
Server Information
NEOS Server
Optimization Guide
Introduction
Types of Problems
Taxonomy
Algorithms
Resources
Algorithms by Problem Type
Included below is a list of algorithms, organized by optimization problem type, that have a linked pag
algorithms, see Algorithms.
Unconstrained Optimization
Line Search Methods
Trust-Region Methods
Truncated Newton Methods
Difference Approximations
Quasi-Newton Methods
Nonlinear Conjugate Gradient Method
Nonlinear Simplex Method
Related Problems
Nonlinear Least-Squares Problems
The Gauss Newton Method
The Levenberg-Marquardt Method
Hybrid Methods
Large-Scale Methods
Nonlinear Least-Squares Problems with Constraints
Nonlinear Equations
Trust-Region and Line Search Methods
Broyden's Method
Tensor Methods
Homotopy Methods
Constrained Optimization