Introduction to Optimization
Introduction to Continuous Optimization II

November 28, 2016

École Centrale Paris, Châtenay-Malabry, France

Dimo Brockhoff
INRIA Saclay – Ile-de-France

Course Overview

Date		Topic
Fri, 7.10.2016		Introduction
Fri, 28.10.2016	D	Introduction to Discrete Optimization + Greedy algorithms I
Fri, 4.11.2016	D	Greedy algorithms II + Branch and bound
Fri, 18.11.2016	D	Dynamic programming
Mon, 21.11.2016 in S103-S105	D	Approximation algorithms ~~and heuristics~~
Fri, 25.11.2016 in S103-S105	C	Randomized Search Heuristics + Introduction to Continuous Optimization I
Mon, 28.11.2016	C	Introduction to Continuous Optimization II
Mon, 5.12.2016	C	Gradient-based Algorithms
Fri, 9.12.2016	C	Stochastic Optimization and Derivative Free Optimization I
Mon, 12.12.2016	C	Stochastic Optimization and Derivative Free Optimization II
Fri, 16.12.2016	C	Benchmarking Optimizers with the COCO platform
Wed, 4.1.2017		Exam

all classes last 3h15 and take place in S115-S117 (see exceptions)

Course Overview

Date		Topic
Fri, 7.10.2016		Introduction
Fri, 28.10.2016	D	Introduction to Discrete Optimization + Greedy algorithms I
Fri, 4.11.2016	D	Greedy algorithms II + Branch and bound
Fri, 18.11.2016	D	Dynamic programming
Mon, 21.11.2016 in S103-S105	D	Approximation algorithms ~~and heuristics~~
Fri, 25.11.2016 in S103-S105	C	Randomized Search Heuristics + Intro. to Continuous Opt. I
Mon, 28.11.2016 in S103-S105	C	Introduction to Continuous Optimization II
Mon, 5.12.2016 in S103-S105	C	Introduction to Continuous Optimization III
Fri, 9.12.2016	C	Constrained Optimization + Descent Methods
Mon, 12.12.2016 in S103-S105	C	Derivative Free Optimization I: CMA-ES
Fri, 16.12.2016	C	Derivative Free Optimization II: Benchmarking Optimizers with the COCO platform
Wed, 4.1.2017		Exam

if not indicated otherwise, classes take place in S115-S117

Overview Continuous Optimization Part

Introduction to Continuous Optimization

- examples (from ML / black-box problems)
- typical difficulties in optimization (e.g. constraints)

Mathematical Tools to Characterize Optima

- reminders about differentiability, gradient, Hessian matrix
- unconstraint optimization
  - first and second order conditions
  - convexity
- constrained optimization

Gradient-based Algorithms

- gradient descent
- quasi-Newton method (BFGS)

Derivative Free Optimization

- stochastic adaptive algorithms (CMA-ES)
- Benchmarking Numerical Blackbox Optimizers

Example Problems

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 simple models

  only affordable (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:

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

Supervised Learning:

Predict $y \in \mathcal{Y}$ from $\boldsymbol{x} \in \mathcal{X}$, given a set of observations (examples) $\{y_i, \boldsymbol{x}_i\}_{i=1,\ldots,N}$

(Simple) Linear regression

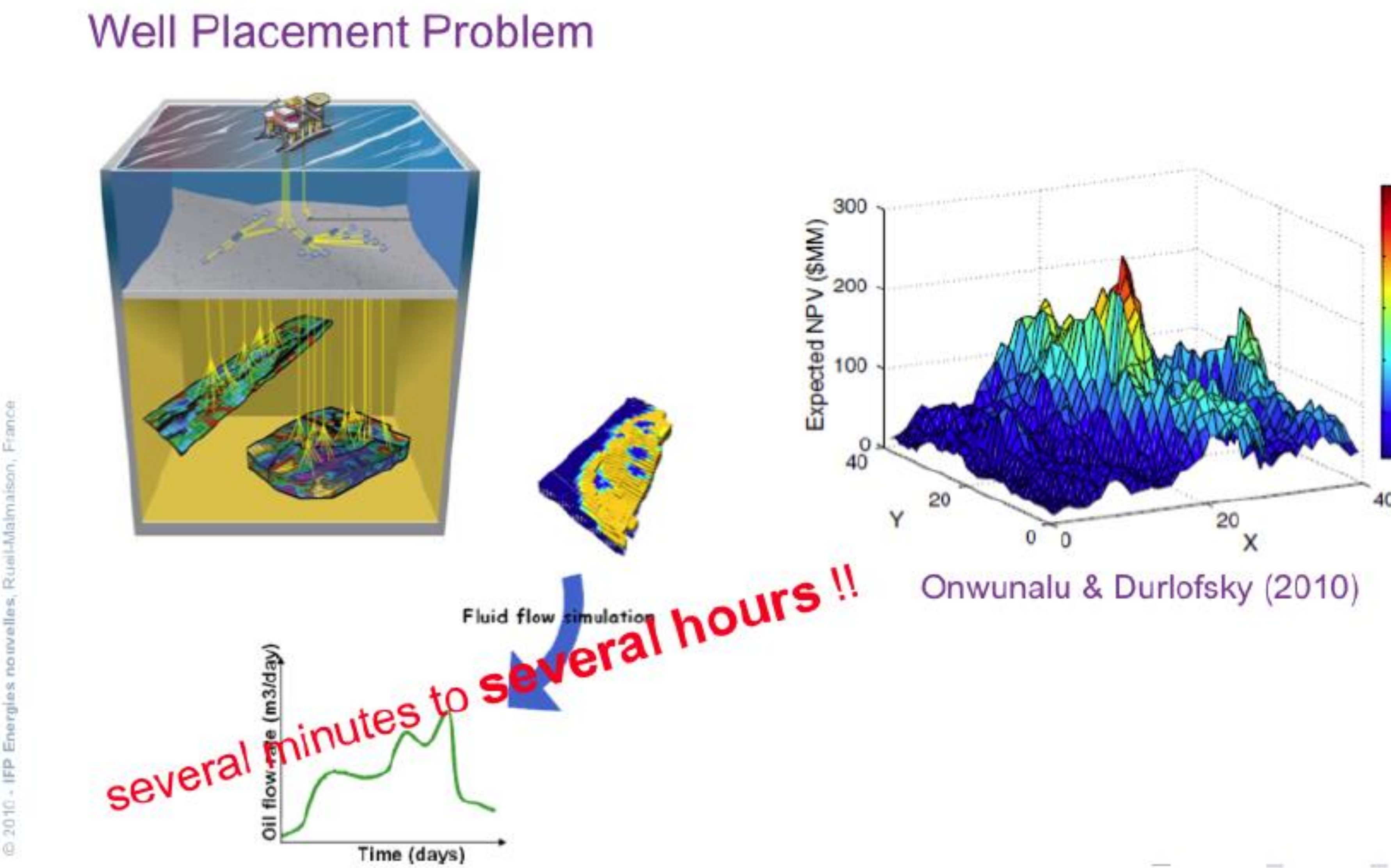
Given a set of data: $\{y_i, \underbrace{x_i^1, \ldots, x_i^p}_{\boldsymbol{x}_i^T}\}_{i=1\ldots N}$

$$\min_{\boldsymbol{w}\in\mathbb{R}^p, \beta\in\mathbb{R}} \underbrace{\sum_{i=1}^{N} |\boldsymbol{w}^T\boldsymbol{x}_i + \beta - y_i|^2}_{||\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{w}} - \mathbf{y}||^2}$$

$\widetilde{\boldsymbol{X}} \in \mathbb{R}^{N\times(p+1)}$, $\widetilde{\boldsymbol{w}} \in \mathbb{R}^{p+1}$

same as data fitting with linear model, i.e. $f_{(\boldsymbol{w},\beta)}(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + \beta$,
$\theta \in \mathbb{R}^{p+1}$

A Real-World Problem in Petroleum Engineering



Function Difficulties

What Makes a Function Difficult to Solve?

- dimensionality

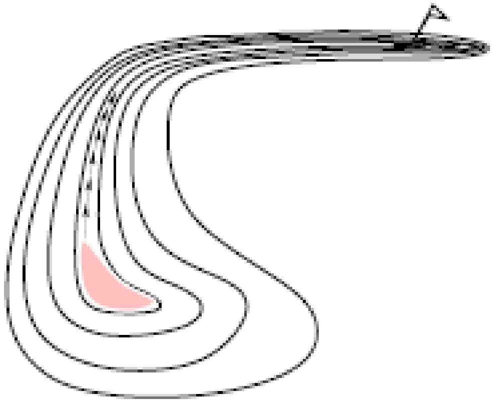
(considerably) larger than three

- non-separability

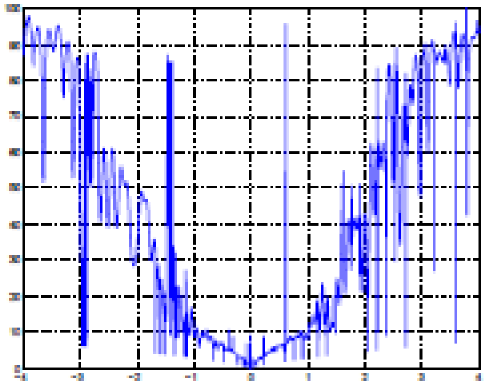
dependencies between the objective variables

- 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

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

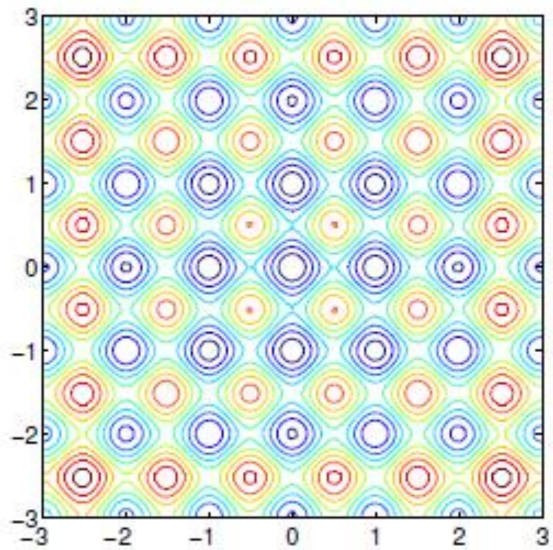
$\Longrightarrow$ it follows that f can be optimized in a sequence of n independent 1-D optimization processes

Example:

Additively decomposable functions

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

Rastrigin function



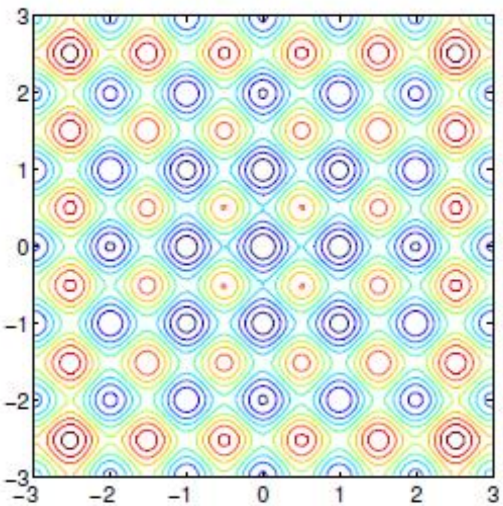
Non-Separable Problems

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

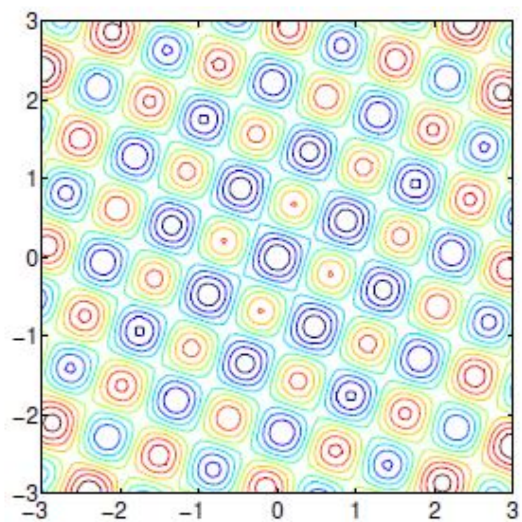
Rotating the coordinate system

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

R rotation matrix



$\xrightarrow{R}$



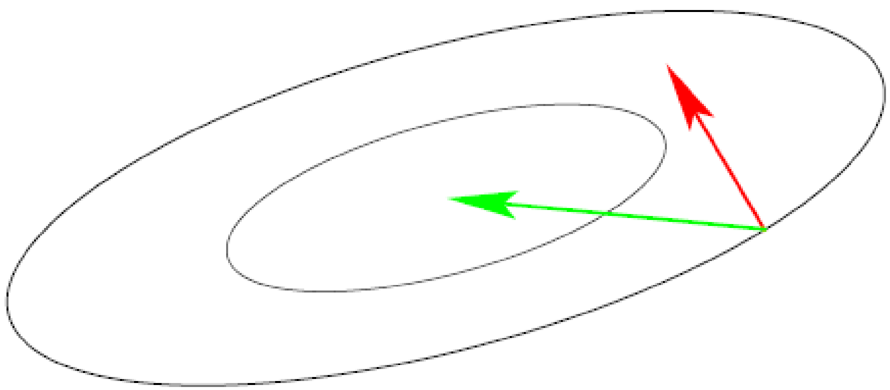
[1] N. Hansen, A. Ostermeier, A. Gawelczyk (1995). "On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation". Sixth ICGA, pp. 57-64, Morgan Kaufmann

[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 of SPD matrix A = ratio between largest and smallest eigenvalue

Condition number equals nine here (kind of well-conditioned). Condition numbers up to 10^{10} are not unusual in real-world problems.

Mathematical Tools to Characterize Optima

Different Notions of Optimum

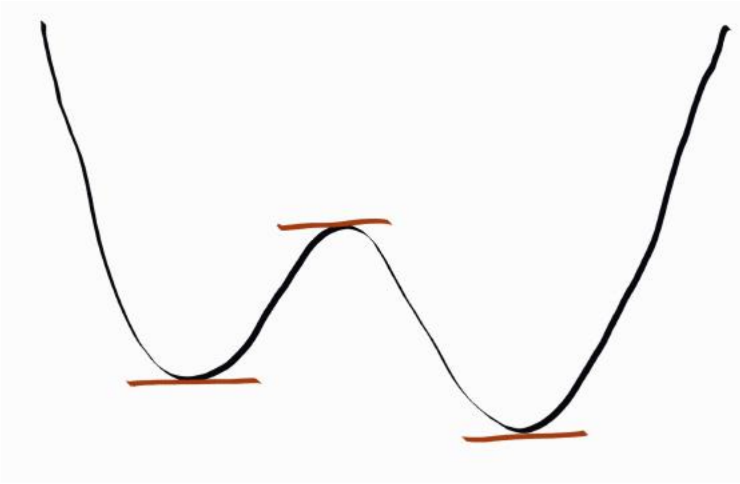
Unconstrained case

- 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

Mathematical Characterization of Optima

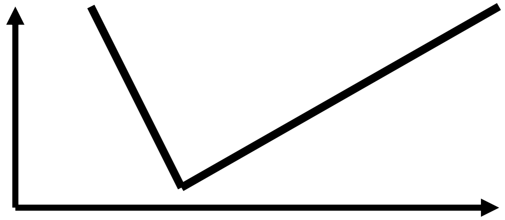
Objective: Derive general characterization of optima

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 differentiability

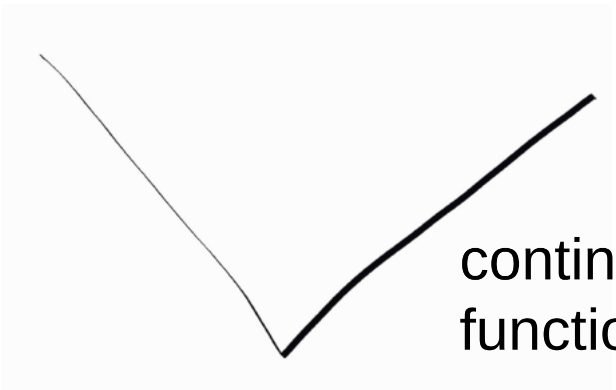


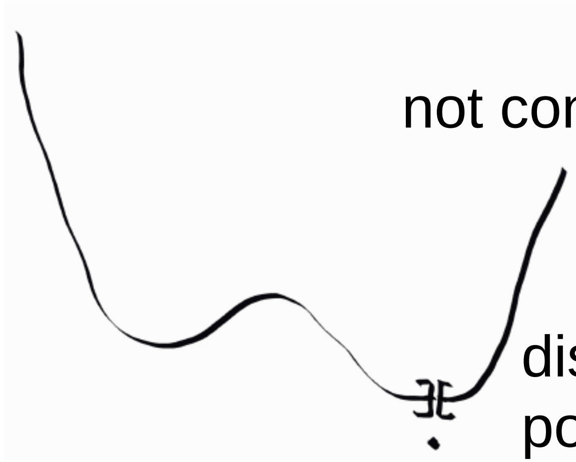
optima of such function can be easily approached by certain type of methods

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 $\forall y \in V$: $||x - y||_V \leq \eta$; $||f(x) - f(y)||_W \leq \epsilon$



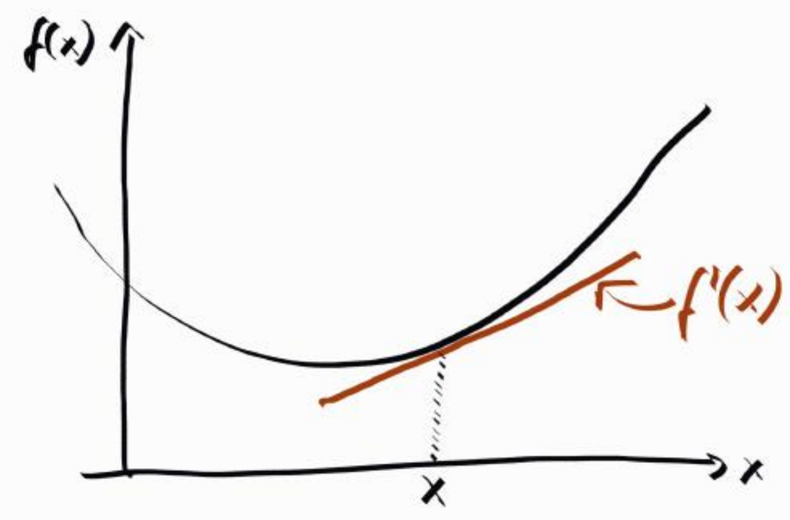


$f: \mathbb{R} \to \mathbb{R}$ is differentiable in $x \in \mathbb{R}$ if

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

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)

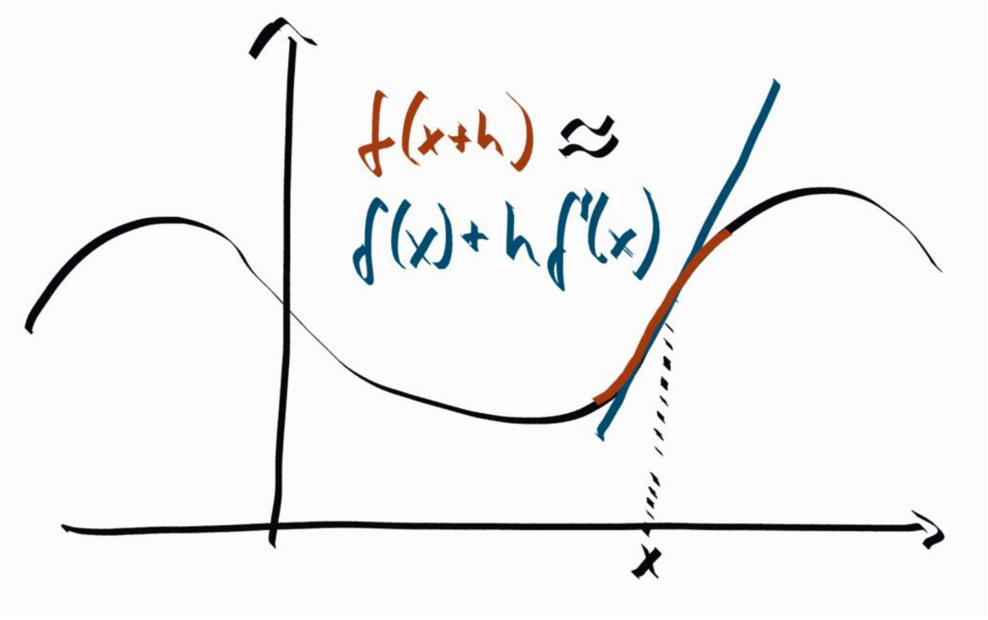
If f is differentiable in x then

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

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.

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

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

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

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 $\mathbb{R}$, $\nabla f(\boldsymbol{x}) = f'(\boldsymbol{x})$

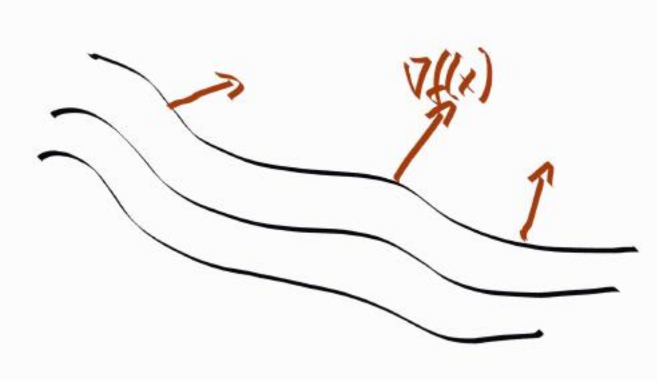
Exercise:

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.



Level Sets and Gradients are Orthogonal

Question: How do we prove in general that the gradient is orthogonal to the level sets?

Answer:

- similar to what we did for two variables
- take any curve within the level set, parametrized by $t \longmapsto c(t)$
- clear: $f(c(t)) = c$ for all t
- derivative wrt to t: $\frac{d}{dt} f\left(c(t)\right) = 0$
- but also $\frac{d}{dt} f(c(t)) = \nabla(f\left(c(t)\right)^T \frac{d}{dt} c(t)$
  [via chain rule, $\frac{d}{dt} c(t)$ is a vector, tangent to the curve in t]

Differentiability in $\mathbb{R}^n$

Taylor Formula – Order One

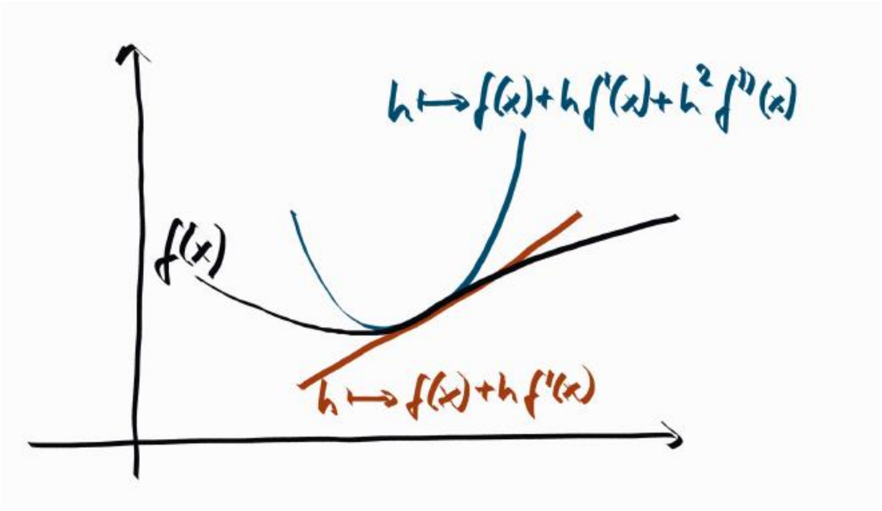
$$f(\boldsymbol{x}+\boldsymbol{h}) = f(\boldsymbol{x}) + \left(\nabla f(\boldsymbol{x})\right)^T \boldsymbol{h} + o(||\boldsymbol{h}||)$$

Reminder: Second Order Differentiability in 1D

- Let $f: D \subseteq \mathbb{R} \to \mathbb{R}$ 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 $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\, \boldsymbol{x}$, $\boldsymbol{x} \in \mathbb{R}^n$, and $A \in \mathbb{R}^{n\times n}$ symmetric.

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 $\mathbb{R}^n$

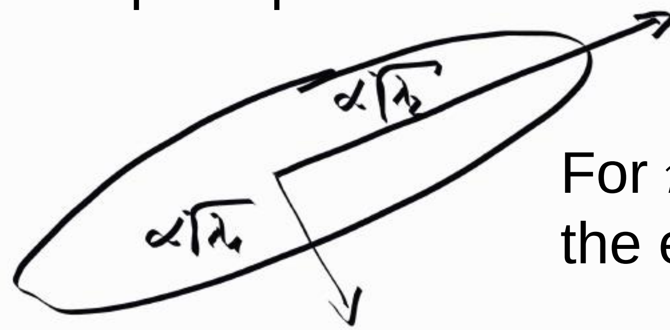
Taylor Formula – Order Two

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

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.

Exercise: Gradients and Level Sets of Convex Quadratic Functions

http://researchers.lille.inria.fr/~brockhof/introoptimization/

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 $x \in \mathbb{R}^2$ of your choice (which should not be on a coordinate axis) into the same plot with the level sets, we created before.

Optimality Conditions for Unconstrained Problems

For 1-dimensional optimization problems $\boldsymbol{f}$: $\mathbb{R} \to \mathbb{R}$

Assume f is differentiable

- $\boldsymbol{x}^*$ is a local optimum $\Longrightarrow f'(\boldsymbol{x}^*) = 0$

  not a sufficient condition: consider $f(\boldsymbol{x}) = \boldsymbol{x}^3$

  proof via Taylor formula: $f(\boldsymbol{x}^* + \boldsymbol{h}) = f(\boldsymbol{x}^*) + f'(\boldsymbol{x}^*)\boldsymbol{h} + o(||\boldsymbol{h}||)$

- points $\boldsymbol{y}$ such that $f'(\boldsymbol{y}) = 0$ are called critical or stationary points

Generalization to $\boldsymbol{n}$-dimensional functions

If $f: U \subset \mathbb{R}^n \longmapsto \mathbb{R}$ is differentiable

- necessary condition: If $\boldsymbol{x}^*$ is a local optimum of f, then $\nabla f(\boldsymbol{x}^*) = 0$

  proof via Taylor formula

Second Order Necessary and Sufficient Opt. Cond.

If f is twice continuously differentiable

- Necessary condition: if $\boldsymbol{x}^*$ is a local minimum, then $\nabla f(\boldsymbol{x}^*) = 0$ and $\nabla^2 f(\boldsymbol{x}^*)$ is positive semi-definite

proof via Taylor formula at order 2

- Sufficient condition: if $\nabla f(\boldsymbol{x}^*) = 0$ and $\nabla^2 f(\boldsymbol{x}^*)$ is positive definite, then $\boldsymbol{x}^*$ is a strict local minimum

Proof of Sufficient Condition:

- Let $\lambda > 0$ be the smallest eigenvalue of $\nabla^2 f(\boldsymbol{x}^*)$, using a second order Taylor expansion, we have for all $\boldsymbol{h}$:
- $f(\boldsymbol{x}^* + \boldsymbol{h}) - f(\boldsymbol{x}^*) = \nabla f(\boldsymbol{x}^*)^T \boldsymbol{h} + \frac{1}{2} \boldsymbol{h}^T \nabla^2 f(\boldsymbol{x}^*) \boldsymbol{h} + o(||\boldsymbol{h}||^2)$

$$> \frac{\lambda}{2} ||\boldsymbol{h}||^2 + \mathrm{o}(||\boldsymbol{h}||^2) = \left(\frac{\lambda}{2} + \frac{o(||\boldsymbol{h}||^2)}{||\boldsymbol{h}||^2} \right) ||\boldsymbol{h}||^2$$

Convex Functions

Let U be a convex open set of $\mathbb{R}^n$ and $f: U \to \mathbb{R}$. The function f is said to be convex if for all $\boldsymbol{x}, \boldsymbol{y} \in U$ and for all $t \in [0,1]$

$$f\big((1-t)\boldsymbol{x} + t\boldsymbol{y}\big) \leq (1-t)f(\boldsymbol{x}) + tf(\boldsymbol{y})$$

Theorem

If f is differentiable, then f is convex if and only if for all $\boldsymbol{x}, \boldsymbol{y}$

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) \geq \big(\nabla f(x)\big)^T (\boldsymbol{y} - \boldsymbol{x})$$

if $n = 1$, the curve is on top of the tangent

If f is twice continuously differentiable, then f is convex if and only if $\nabla^2 f(\boldsymbol{x})$ is positive semi-definite for all $\boldsymbol{x}$.

Convex Functions: Why Convexity?

Examples of Convex Functions:

- $f(\boldsymbol{x}) = a^T\boldsymbol{x} + b$
- $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} + a^T\boldsymbol{x} + b$, A symmetric positive definite
- the negative of the entropy function ($\text{i.e.}\ f(\boldsymbol{x}) = -\sum_{i=1}^{n} \boldsymbol{x}_i \ln(\boldsymbol{x}_i)$)

Exercise:

Let $f: U \to \mathbb{R}$ be a convex and differentiable function on a convex open U.
Show that if $\nabla f(\boldsymbol{x}^*) = 0$, then $\boldsymbol{x}^*$ is a global minimum of f

Constrained Optimization

Equality Constraint

Objective:

Generalize the necessary condition of $\nabla f(x) = 0$ at the optima of f

when f is in $\mathcal{C}^1$, i.e. is differentiable and its derivative is continuous

Theorem:

Be U an open set of $(E, ||\quad||)$, and $f: U \to \mathbb{R}$, $g: U \to \mathbb{R}$ in $\mathcal{C}^1$.

Let $a \in E$ satisfy

$$\begin{cases} f(a) = \inf \{f(x) \mid x \in \mathbb{R}^n, g(x) = 0\} \\ g(a) = 0 \end{cases}$$

i.e. a is optimum of the problem

If $\nabla g(a) \neq 0$, then there exists a constant $\lambda \in \mathbb{R}$ called *Lagrange multiplier*, such that

$$\underbrace{\nabla f(a) + \lambda \nabla g(a) = 0}_{\text{i.e. gradients of } f \text{ and } g \text{ in } a \text{ are colinear}}$$

Note: a need not be a global minimum but a local one

Geometrical Interpretation Using an Example

Exercise:

Consider the problem

$$\inf \{ f(x,y) \mid (x,y) \in \mathbb{R}^2, g(x,y) = 0\}$$

$f(x,y) = y - x^2 \qquad g(x,y) = x^2 + y^2 - 1$

1) Plot the level sets of f, plot $g = 0$
2) Compute ∇f and ∇g
3) Find the solutions with $\nabla f + \lambda \nabla g = 0$

   equation solving with 3 unknowns (x, y, λ)
4) Plot the solutions of 3) on top of the level set graph of 1)

Interpretation of Euler-Lagrange Equation

Intuitive way to retrieve the Euler-Lagrange equation:

- In a local minimum a of a constrained problem, the hypersurfaces (or level sets) $f = f(a)$ and $g = 0$ are necessarily tangent (otherwise we could decrease f by moving along $g = 0$).
- Since the gradients $\nabla f(a)$ and $\nabla g(a)$ are orthogonal to the level sets $f = f(a)$ and $g = 0$, it follows that $\nabla f(a)$ and $\nabla g(a)$ are colinear.

Theorem

- Assume $f: U \to \mathbb{R}$ and $g_k: U \to \mathbb{R}$ $(1 \leq k \leq p)$ are $\mathcal{C}^1$.
- Let a be such that

$$\begin{cases} f(a) = \inf \{f(x) \mid x \in \mathbb{R}^n, \quad g_k(x) = 0, \quad 1 \leq k \leq p\} \\ g_k(a) = 0 \text{ for all } 1 \leq k \leq p \end{cases}$$

- If $\left(\nabla g_k(a)\right)_{1 \leq k \leq p}$ are linearly independent, then there exist p real constants $(\lambda_k)_{1 \leq k \leq p}$ such that

$$\nabla f(a) + \sum_{k=1}^{p} \underbrace{\lambda_k}_{\text{Lagrange multiplier}} \nabla g_k(a) = 0$$

again: a does not need to be global but local minimum

The Lagrangian

- Define the Lagrangian on $\mathbb{R}^n \times \mathbb{R}^p$ as

$$\mathcal{L}(x, \{\lambda_k\}) = f(x) + \sum_{k=1}^{p} \lambda_k g_k(x)$$

- To find optimal solutions, we can solve the optimality system

$$\begin{cases} \text{Find } (x, \{\lambda_k\}) \in \mathbb{R}^n \times \mathbb{R}^p \text{ such that } \nabla f(x) + \sum_{k=1}^{p} \lambda_k \nabla g_k(x) = 0 \\ g_k(x) = 0 \ \text{ for all } 1 \leq k \leq p \end{cases}$$

$$\Leftrightarrow \begin{cases} \text{Find } (x, \{\lambda_k\}) \in \mathbb{R}^n \times \mathbb{R}^p \text{ such that } \nabla_x \mathcal{L}(x, \{\lambda_k\}) = 0 \\ \nabla_{\lambda_k} \mathcal{L}(x, \{\lambda_k\})(x) = 0 \ \text{ for all } 1 \leq k \leq p \end{cases}$$

Inequality Constraints: Definitions

Let $\mathcal{U} = \{x \in \mathbb{R}^n \mid g_k(x) = 0 \text{ (for } k \in E),\ g_k(x) \leq 0 \text{ (for } k \in I)\}$.

Definition:

The points in $\mathbb{R}^n$ that satisfy the constraints are also called *feasible* points.

Definition:

Let $a \in \mathcal{U}$, we say that the constraint $g_k(x) \leq 0$ (for $k \in I$) is *active* in a if $g_k(a) = 0$.

Inequality Constraint: Karush-Kuhn-Tucker Theorem

Theorem (Karush-Kuhn-Tucker, KKT):

Let U be an open set of $(E, || \, ||)$ and $f: U \to \mathbb{R}$, $g_k: U \to \mathbb{R}$, all $\mathcal{C}^1$

Furthermore, let $a \in U$ satisfy

$$\begin{cases} f(a) = \inf(f(x) \mid x \in \mathbb{R}^n, g_k(x) = 0 \ (\text{for } k \in E), g_k(x) \leq 0 \ (\text{for } k \in \mathrm{I}) \\ g_k(a) = 0 \ (\text{for } k \in E) \\ g_k(a) \leq 0 \ (\text{for } k \in I) \end{cases}$$

also works again for a being a local minimum

Let I_a^0 be the set of constraints that are active in a. Assume that $\left(\nabla g_k(a)\right)_{k \in E \cup I_a^0}$ are linearly independent.

Then there exist $(\lambda_k)_{1 \leq k \leq p}$ that satisfy

$$\begin{cases} \nabla f(a) + \sum_{k=1}^{p} \lambda_k \nabla g_k(a) = 0 \\ g_k(a) = 0 \ (\text{for } k \in E) \\ g_k(a) \leq 0 \ (\text{for } k \in I) \\ \lambda_k \geq 0 \ (\text{for } k \in I_a^0) \\ \lambda_k g_k(a) = 0 \ (\text{for } k \in E \cup I) \end{cases}$$

Inequality Constraint: Karush-Kuhn-Tucker Theorem

Theorem (Karush-Kuhn-Tucker, KKT):

Let U be an open set of $(E, || \, ||)$ and $f: U \to \mathbb{R}$, $g_k: U \to \mathbb{R}$, all $\mathcal{C}^1$

Furthermore, let $a \in U$ satisfy

$$\begin{cases} f(a) = \inf(f(x) \mid x \in \mathbb{R}^n, g_k(x) = 0 \;(\text{for } k \in E), g_k(x) \leq 0 \;(\text{for } k \in \mathrm{I}) \\ g_k(a) = 0 \;(\text{for } k \in E) \\ g_k(a) \leq 0 \;(\text{for } k \in I) \end{cases}$$

Let I_a^0 be the set of constraints that are active in a. Assume that $\left(\nabla g_k(a)\right)_{k \in E \cup I_a^0}$ are linearly independent.

Then there exist $(\lambda_k)_{1 \leq k \leq p}$ that satisfy

$$\begin{cases} \nabla f(a) + \sum_{k=1}^{p} \lambda_k \nabla g_k(a) = 0 \\ g_k(a) = 0 \;(\text{for } k \in E) \\ g_k(a) \leq 0 \;(\text{for } k \in I) \\ \lambda_k \geq 0 \;(\text{for } k \in I_a^0) \\ \lambda_k g_k(a) = 0 \;(\text{for } k \in E \cup I) \end{cases}$$

either active constraint or $\lambda_k = 0$

Descent Methods

General principle

1. choose an initial point $\boldsymbol{x}_0$, set $t = 1$
2. while not happy
   - choose a descent direction $\boldsymbol{d}_t \neq 0$
   - line search:
     - choose a step size $\sigma_t > 0$
     - set $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \sigma_t \boldsymbol{d}_t$
   - set $t = t + 1$

Remaining questions

- how to choose $\boldsymbol{d}_t$?
- how to choose σ_t?

Gradient Descent

Rationale: $\boldsymbol{d}_t = -\nabla f(\boldsymbol{x}_t)$ is a descent direction

indeed for f differentiable

$$f\big(x - \sigma\nabla f(x)\big) = f(x) - \sigma||\nabla f(x)||^2 + o(\sigma||\nabla f(x)||)$$
$$< f(x) \text{ for } \sigma \text{ small enough}$$

Step-size

- optimal step-size: $\sigma_t = \underset{\sigma}{\operatorname{argmin}}\, f(\boldsymbol{x}_t - \sigma\nabla f(\boldsymbol{x}_t))$
- **Line Search:** total or partial optimization w.r.t. σ
  Total is however often too "expensive" (needs to be performed at each iteration step)
  Partial optimization: execute a limited number of trial steps until a loose approximation of the optimum is found. Typical rule for partial optimization: Armijo rule

see next slide and exercise

Stopping criteria:

norm of gradient smaller than ϵ

The Armijo-Goldstein Rule

Choosing the step size:

- Only to decrease f-value not enough to converge (quickly)
- Want to have a reasonably large decrease in f

Armijo-Goldstein rule:

- also known as backtracking line search
- starts with a (too) large estimate of σ and reduces it until f is reduced enough
- what is enough?
  - assuming a linear f e.g. $m_k(x) = f(x_k) + \nabla f(x_k)^T(x - x_k)$
  - expected decrease if step of σ_k is done in direction $\boldsymbol{d}$: $\sigma_k \nabla f(x_k)^T \boldsymbol{d}$
  - actual decrease: $f(x_k) - f(x_k + \sigma_k \boldsymbol{d})$
  - stop if actual decrease is at least constant times expected decrease (constant typically chosen in [0, 1])

The Armijo-Goldstein Rule

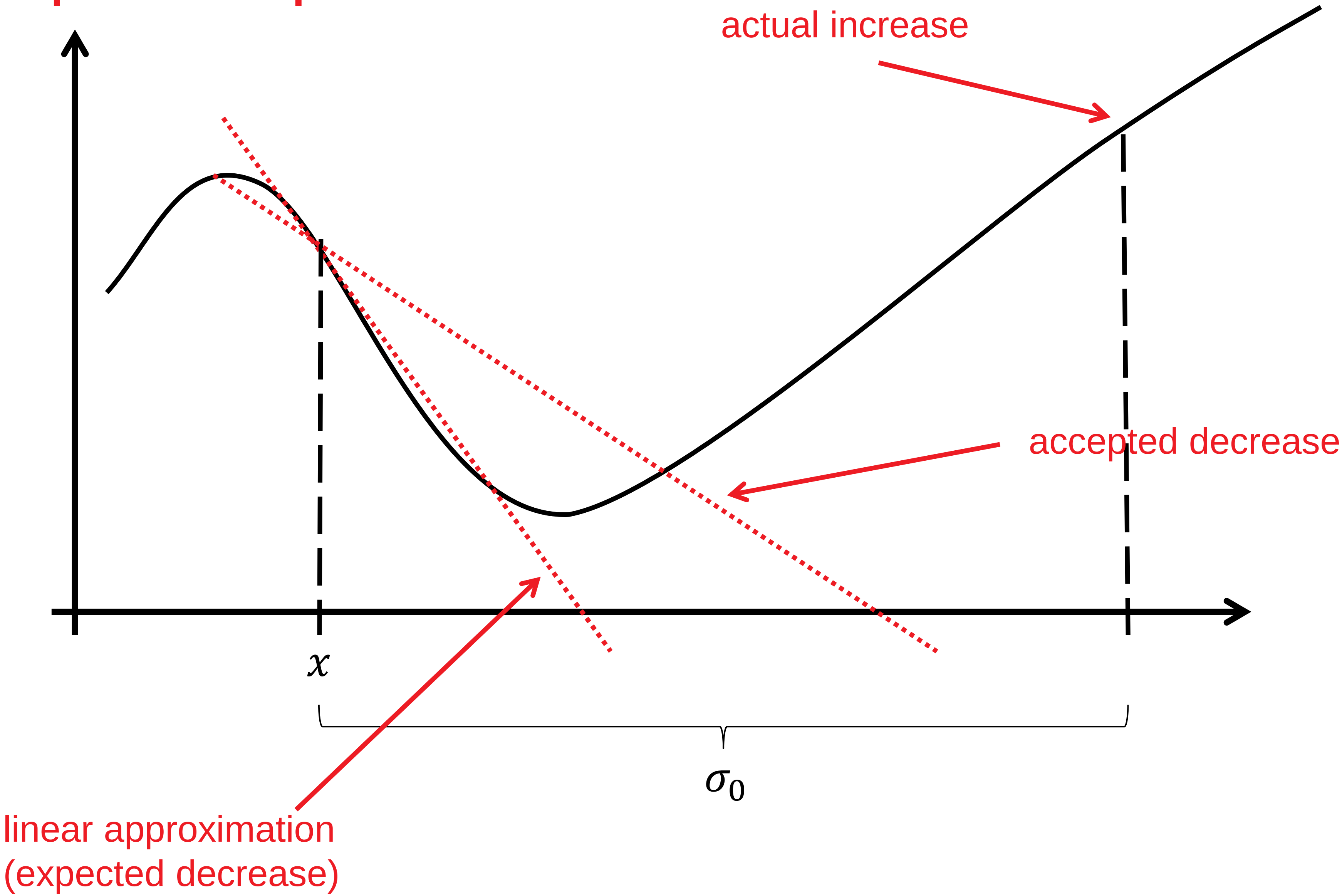
The Actual Algorithm:

Input: descent direction $\mathbf{d}$, point $\mathbf{x}$, objective function $f(\mathbf{x})$ and its gradient $\nabla f(\mathbf{x})$, parameters $\sigma_0 = 10$, $\theta \in [0,1]$ and $\beta \in (0,1)$
Output: step-size σ

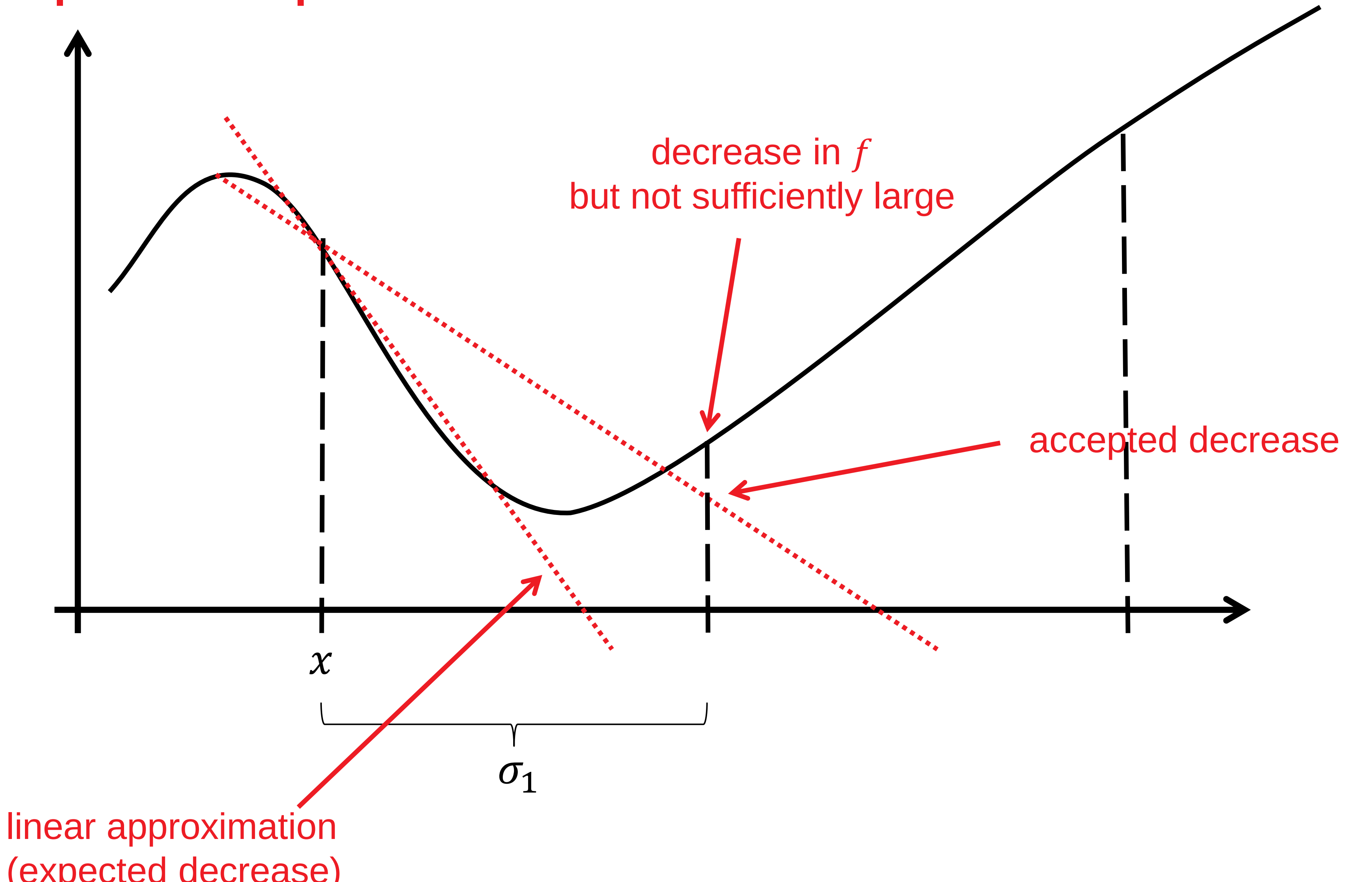
Initialize σ: $\sigma \leftarrow \sigma_0$
while $f(\mathbf{x} + \sigma \mathbf{d}) > f(\mathbf{x}) + \theta\sigma\nabla f(\mathbf{x})^T\mathbf{d}$ **do**
    $\sigma \leftarrow \beta\sigma$
end while

Armijo, in his original publication chose $\beta = \theta = 0.5$.
Choosing $\theta = 0$ means the algorithm accepts any decrease.

Graphical Interpretation

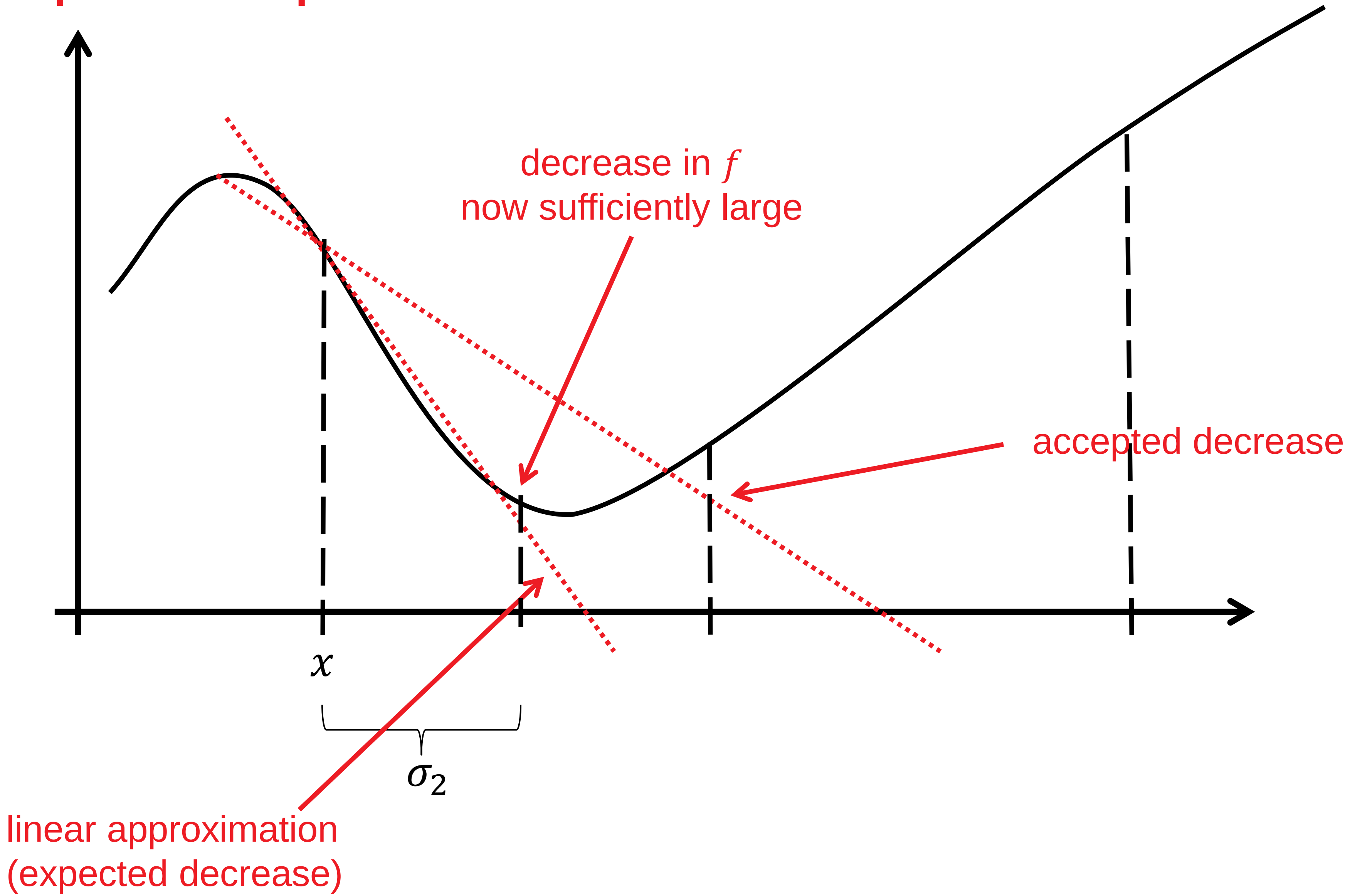


Graphical Interpretation



The Armijo-Goldstein Rule

Graphical Interpretation



Gradient Descent: Simple Theoretical Analysis

Assume f is twice continuously differentiable, convex and that $\mu I_d \preccurlyeq \nabla^2 f(x) \preccurlyeq L I_d$ with $\mu > 0$ holds, assume a fixed step-size $\sigma_t = \frac{1}{L}$

Note: $A \preccurlyeq B$ means $x^T A x \leq x^T B x$ for all x

$$x_{t+1} - x^* = x_t - x^* - \sigma_t \nabla^2 f(y_t)(x_t - x^*) \text{ for some } y_t \in [x_t, x^*]$$

$$x_{t+1} - x^* = \left(I_d - \frac{1}{L} \nabla^2 f(y_t) \right) (x_t - x^*)$$

$$\text{Hence } ||x_{t+1} - x^*||^2 \leq |||I_d - \tfrac{1}{L} \nabla^2 f(y_t)|||^2 \; ||x_t - x^*||^2$$

$$\leq \left(1 - \frac{\mu}{L}\right)^2 ||x_t - x^*||^2$$

Linear convergence: $||x_{t+1} - x^*|| \leq \left(1 - \frac{\mu}{L}\right) ||x_t - x^*||$

algorithm slower and slower with increasing condition number

Non-convex setting: convergence towards stationary point

Newton Method

- descent direction: $-[\nabla^2 f(x_k)]^{-1}\nabla f(x_k)$ [so-called Newton direction]
- The Newton direction:
  - minimizes the best (locally) quadratic approximation of f: $\tilde{f}(x+\Delta x) = f(x) + \nabla f(x)^T \Delta x + \frac{1}{2}(\Delta x)^T \nabla^2 f(x) \Delta \mathrm{x}$
  - points towards the optimum on $f(x) = (x - x^*)^T A (x - x^*)$
- however, Hessian matrix is expensive to compute in general and its inversion is also not easy

quadratic convergence

(i.e. $\lim_{k \to \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^2} = \mu > 0$)

Remark: Affine Invariance

Affine Invariance: same behavior on $f(x)$ and $f(Ax+b)$ for $A \in \mathrm{GLn}(\mathbb{R})$

- Newton method is affine invariant

  see **http://users.ece.utexas.edu/~cmcaram/EE381V_2012F/Lecture_6_Scribe_Notes.final.pdf**

- same convergence rate on all convex-quadratic functions
- Gradient method not affine invariant

Quasi-Newton Method: BFGS

$x_{t+1} = x_t - \sigma_t H_t \nabla f(x_t)$ where H_t is an approximation of the inverse Hessian

Key idea of Quasi Newton:

successive iterates x_t, x_{t+1} and gradients $\nabla f(x_t)$, $\nabla f(x_{t+1})$ yield second order information

$$q_t \approx \nabla^2 f(x_{t+1}) p_t$$

where $p_t = x_{t+1} - x_t$ and $q_t = \nabla f(x_{t+1}) - \nabla f(x_t)$

Most popular implementation of this idea: Broyden-Fletcher-Goldfarb-Shanno (BFGS)

- default in MATLAB's **`fminunc`** and python's **`scipy.optimize.minimize`**

Conclusions

I hope it became clear...

...what are the difficulties to cope with when solving numerical optimization problems

in particular dimensionality, non-separability and ill-conditioning

...what are gradient and Hessian

...what is the difference between gradient and Newton direction

...and that adapting the step size in descent algorithms is crucial.