Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Basic solution methods for unconstrained problems

Solution Methods for Unconstrained Optimization $\min_{\boldsymbol{x} \in R^n} f(\boldsymbol{x})$

- **Unconstrained optimization methods**
  - **Direct methods**
  - **Indirect methods**

Indirect Methods – Concept

- First-order necessary conditions

$$\nabla f(\boldsymbol{x}) = \mathbf{0} \Leftrightarrow \begin{cases} \left.\dfrac{\partial f}{\partial x_1}\right|_{\boldsymbol{x}} = 0 = g_1(\boldsymbol{x}) \\ \left.\dfrac{\partial f}{\partial x_2}\right|_{\boldsymbol{x}} = 0 = g_2(\boldsymbol{x}) \\ \vdots \\ \left.\dfrac{\partial f}{\partial x_1}\right|_{\boldsymbol{x}} = 0 = g_n(\boldsymbol{x}) \end{cases} \Leftrightarrow \quad \boldsymbol{g}(\boldsymbol{x}) = \mathbf{0}$$

nonlinear system of equations

- The optimal solution is found by solving the system of equations analytically or numerically (e.g., by Newton's method).
- Differentiation and solution of the system of equations is challenging for complex problems!

Solution Methods for Unconstrained Optimization

Unconstrained optimization methods

Direct methods

Indirect methods

Optimal solution is found by solving the system of equations (**optimality conditions**):

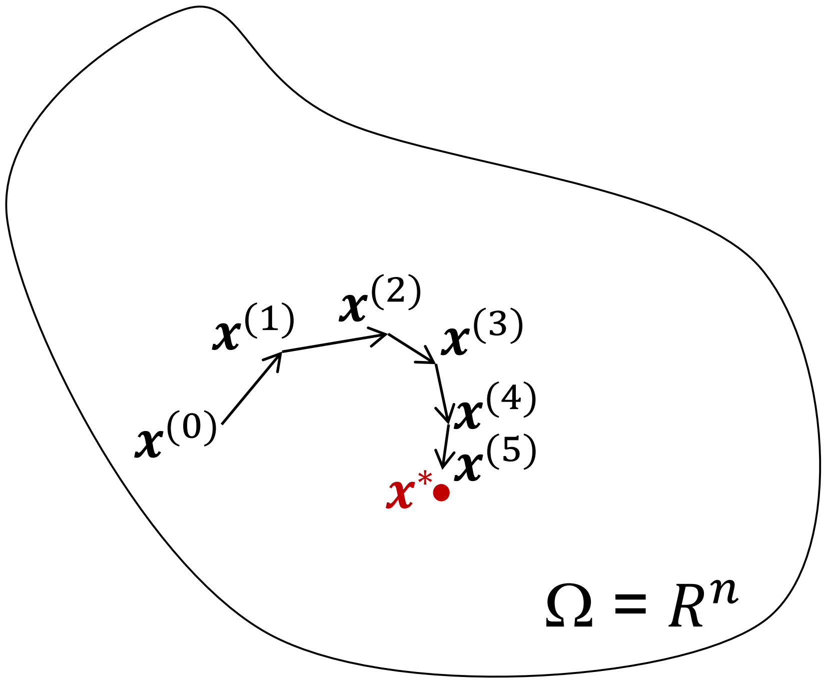
$$\nabla f(\boldsymbol{x}) = \boldsymbol{0}$$

analytically or numerically

Direct Methods – Concept

Idea: Construct a convergent sequence of $\{\boldsymbol{x}^{(k)}\}_{k=1}^{\infty}$, which fulfills the following conditions:

$$\exists \bar{k} \geq 0:\ f\left(\boldsymbol{x}^{(k+1)}\right) < f\left(\boldsymbol{x}^{(k)}\right)\ \forall k > \bar{k} \qquad \text{and} \qquad \lim_{k\to\infty} \boldsymbol{x}^{(k)} = \boldsymbol{x}^* \in R^n$$



Definition: Rate of Convergence

Idea: Construct a convergent sequence of $\{\boldsymbol{x}^{(k)}\}_{k=1}^{\infty}$, which fulfills the following conditions:

$$\exists \bar{k} \geq 0\colon\ f\left(\boldsymbol{x}^{(k+1)}\right) < f\left(\boldsymbol{x}^{(k)}\right)\ \forall k > \bar{k} \qquad \text{and} \qquad \lim_{k\to\infty} \boldsymbol{x}^{(k)} = \boldsymbol{x}^* \in R^n$$

Rate of convergence:

- Linear: if there exists a constant $C \in (0,1)$, such that for sufficiently large k :

$$\left\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\right\| \leq C\left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\right\|$$

- Order p (often $p = 2$): if there exists a constant $M > 0$, such that

$$\left\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\right\| \leq M\left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\right\|^p$$

- Superlinear: if there exists a sequence c_k converging to zero, i.e., $\lim_{k\to\infty} c_k = 0$, such that

$$\left\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\right\| \leq c_k\left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\right\|$$

Solution Methods for Unconstrained Optimization

Unconstrained optimization methods

Direct methods

Optimal solution is found by directly improving the objective function via **iterative descent**.

Indirect methods

Optimal solution is found by solving the system of equations (**optimality conditions**):

$$\boldsymbol{\nabla} f(\boldsymbol{x}) = \mathbf{0}$$

analytically or numerically.

Direct vs Indirect: Nomenclature not consistent in Literature

- Throughout class we use "direct" and "indirect":
  - "indirect methods": 1. set up optimality conditions and then 2. try to solve the system of equations (or equations and inequalities)
  - "direct methods": directly aim to improve objective function (or objective function and constraints). These methods hope to converge to optimality conditions.
- In the literature there are many alternative uses of the word, including
  - exactly the opposite than ours
  - "direct": without the use of derivatives, "indirect": using derivatives
  - only in the context of dynamic optimization problems:
    - "direct": first convert to nonlinear program
    - "indirect": first set up optimality conditions
  - only in the context of constrained problems
    - "direct": only feasible iterates
    - "indirect": infeasible iterates are allowed

Solution Methods for Unconstrained Optimization

Unconstrained optimization methods

- **Direct methods**

  Iterative descent

  - **Line-search approach**
  - **Trust-region approach**

- **Indirect methods**

  Optimal solution is found by solving the system of equations (**optimality conditions**):

  $$\nabla f(\boldsymbol{x}) = \mathbf{0}$$

  analytically or numerically.

Check Yourself

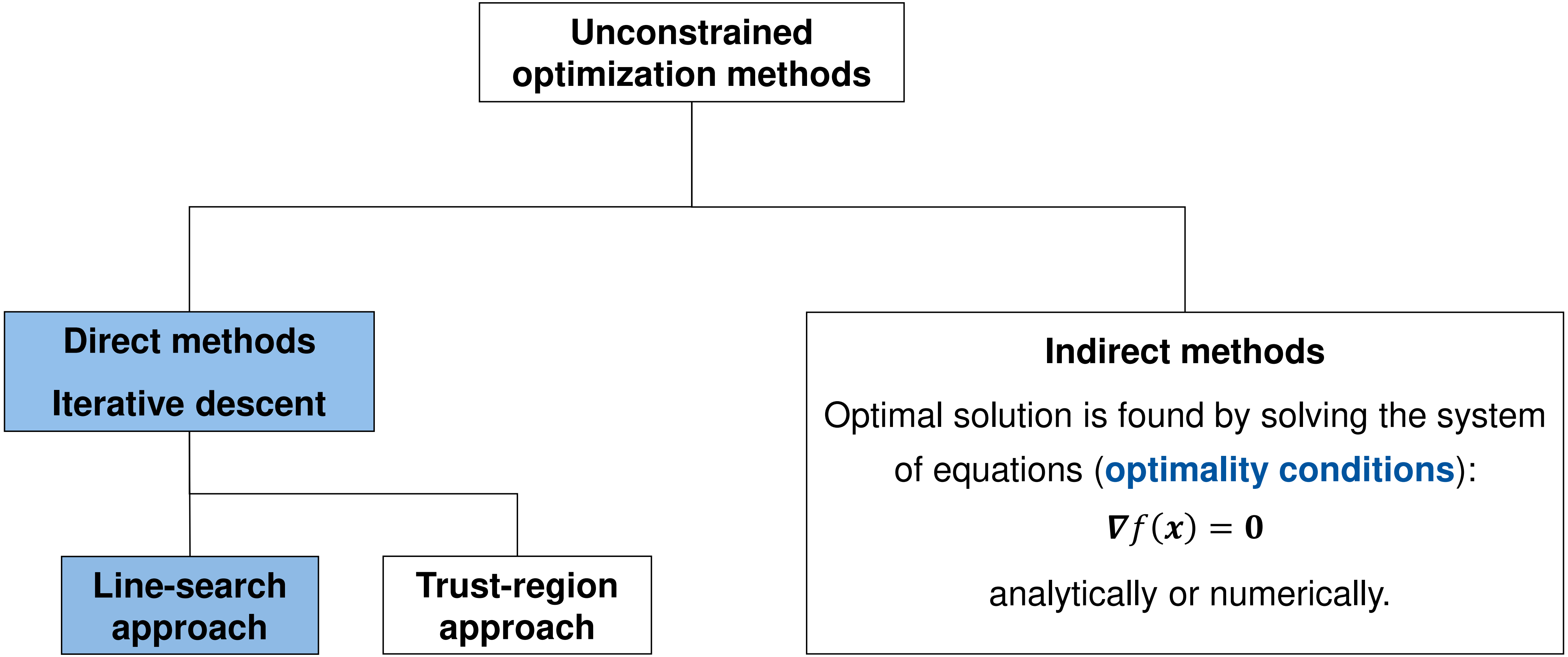
- What are direct vs indirect methods?
- Which direct methods did we learn?
- Which convergence rates exist? Why is the convergence rate important?

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Line search: basic idea and step length

Solution Methods for Unconstrained Optimization



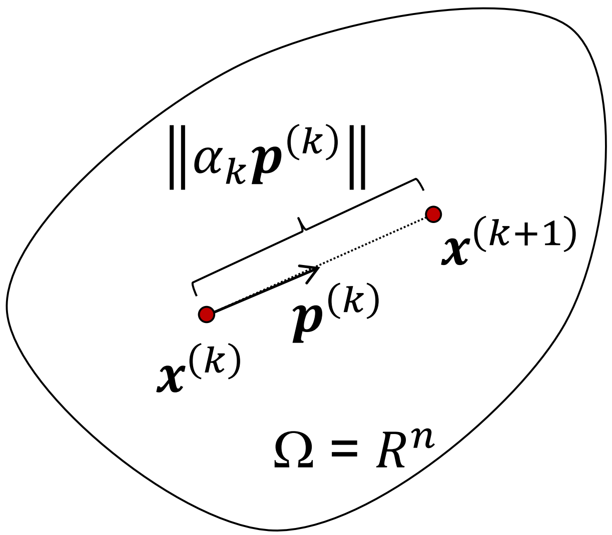
Direct Methods – Line-Search Approach

Definition (descent direction):

A vector $\boldsymbol{p}$ is called descent direction at $\boldsymbol{x}^{(k)}$, if $\nabla f\left(\boldsymbol{x}^{(k)}\right)^T \boldsymbol{p} < 0$ holds.

Basic algorithm (line-search):

1. Choose a descent direction, $\boldsymbol{p}^{(k)}$, such that
   $\nabla f\left(\boldsymbol{x}^{(k)}\right)^T \boldsymbol{p}^{(k)} < 0$
2. Determine a step length α_k
3. Set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)}$



Open issues:

- Determination of the descent direction $\boldsymbol{p}^{(k)}$?
- Calculation of the step length α_k?

Calculation of Step Length α_k

The exact line search strategy:

1. Define the one-dimensional function along the descent direction $\boldsymbol{p}^{(k)}$:

$$\phi(\alpha) = f(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{p}^{(k)})$$

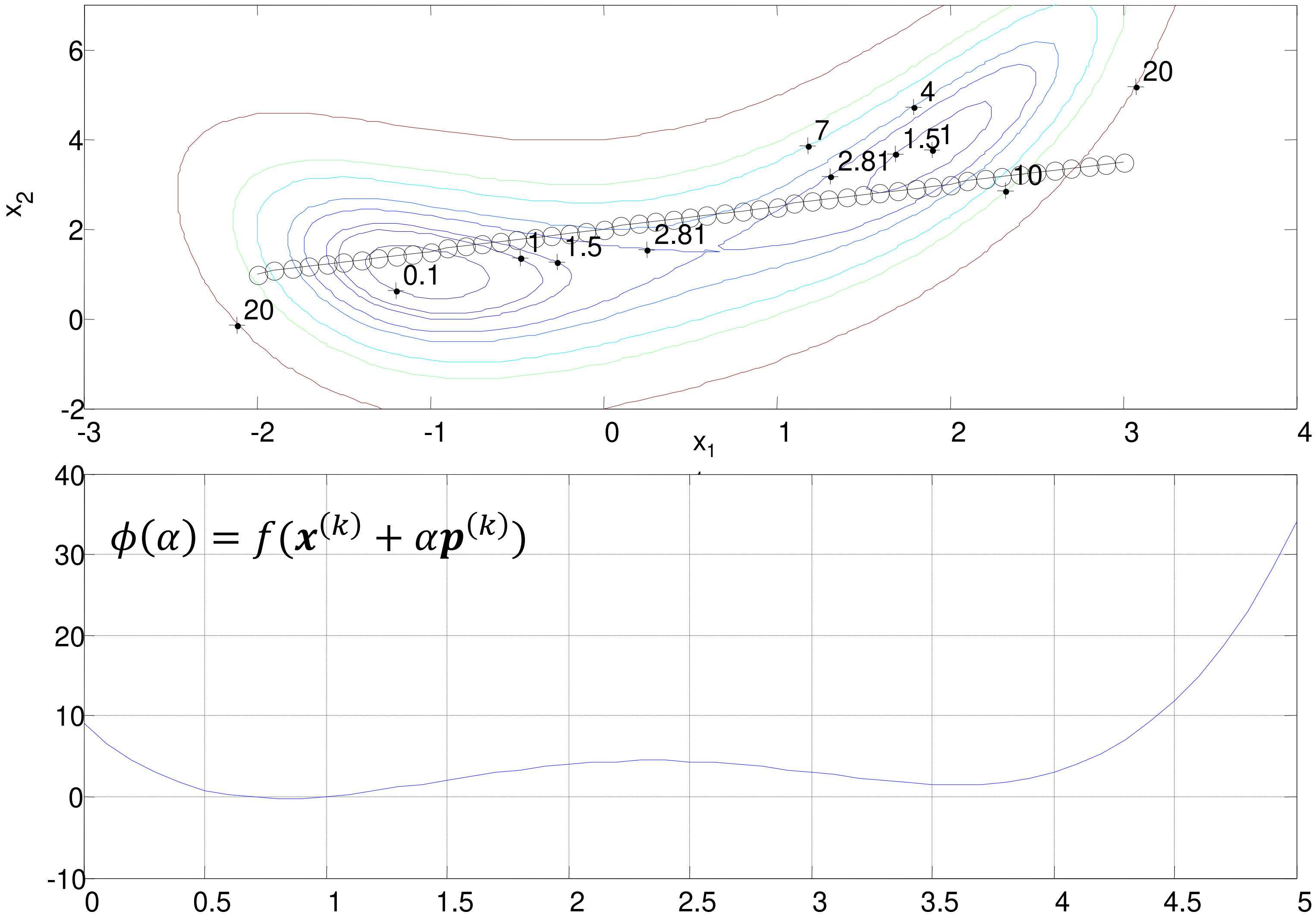
2. Solve the one-dimensional minimization problem

$$\min_{\alpha > 0} \phi(\alpha)$$

Remarks

1. Naively speaking it would be ideal to globally minimize $\phi(\alpha)$. Generally, it is very expensive to find this solution. It is not necessarily a good idea since the search is one-dimensional
2. One could also search for some local solution. But this is often also too expensive (need function and/or gradient evaluations at a number of points).
3. Practical strategies (so-called non-exact LS): find α such that $f(\boldsymbol{x}^{(k+1)})$ becomes as small as possible with minimal effort.

Practical Line-Search Strategies

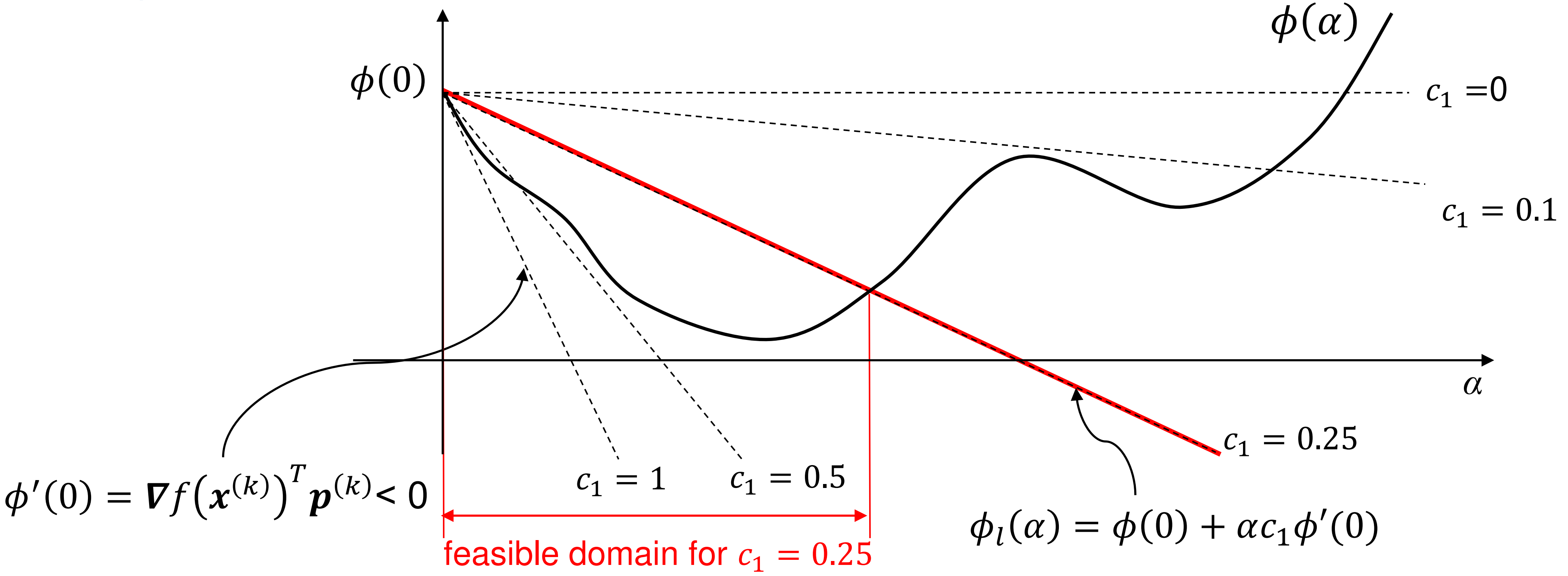


Armijo Condition

Theorem[1]:

Let f be continuously differentiable, $\boldsymbol{p}^{(k)}$ a descent direction, and let $c_1 \in (0,1)$ be given. Then there exists an $\alpha > 0$, such that for $\phi(\alpha) \coloneqq f(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{p}^{(k)})$, the condition $\phi(\alpha) \leq \phi(0) + \alpha c_1 \phi'(0)$ holds.

Geometrical interpretation:



[1] Nocedal J., Wright S. J., Numerical Optimization, 2nd Edition, Springer, 2006

Simple Line-Search Algorithm

Remarks:

1. The choice of a step length, which fulfills the Armijo condition guarantees the descent of f:

$$\phi'(0) = \nabla f\left(\boldsymbol{x}^{(k)}\right)^T \boldsymbol{p}^{(k)} < 0 \text{ (}\boldsymbol{p}^{(k)} \text{ is a descent direction)}$$

$$\phi(\alpha) \leq \phi(0) + \alpha c_1 \phi'(0)$$

$\Rightarrow \phi(\alpha) < \phi(0)$
$\Rightarrow$ a descent is guaranteed!

2. The choice of c_1 is crucial:
   - Large c_1 leads to small values of α, such that $\boldsymbol{x}^{(k+1)} \approx \boldsymbol{x}^{(k)}$.
   - Small c_1 potentially results in small reduction of f and therefore slower convergence

Simple line-search algorithm:

choose $\alpha_1 > 0;\ \rho, c_1 \in (0,1)$

set $\alpha = \alpha_1$

repeat $\alpha \leftarrow \rho\alpha$ until $\phi(\alpha) \leq \phi(0) + \alpha c_1 \phi'(0)$

Improved Line-Search Algorithm

choose $\alpha_0 > 0$ and $c_1 \in (0,1)$

if $\phi(\alpha_0) \leq \phi(0) + \alpha_0 c_1 \phi'(0)$ STOP, else

find a better $\alpha \in (0, \alpha_0)$ through *quadratic interpolation* of available data:

$$\alpha_1 = -\frac{\phi'(0)\alpha_0^2}{2[\phi(\alpha_0) - \phi(0) - \phi'(0)\alpha_0]}$$

if $\phi(\alpha_1) \leq \phi(0) + \alpha_1 c_1 \phi'(0)$ STOP, else

find a better $\alpha \in (0, \alpha_1)$ through *cubic interpolation* of available data (how ?)

repeat the procedure of *cubic interpolation*, until the condition is fulfilled

Wolfe Conditions

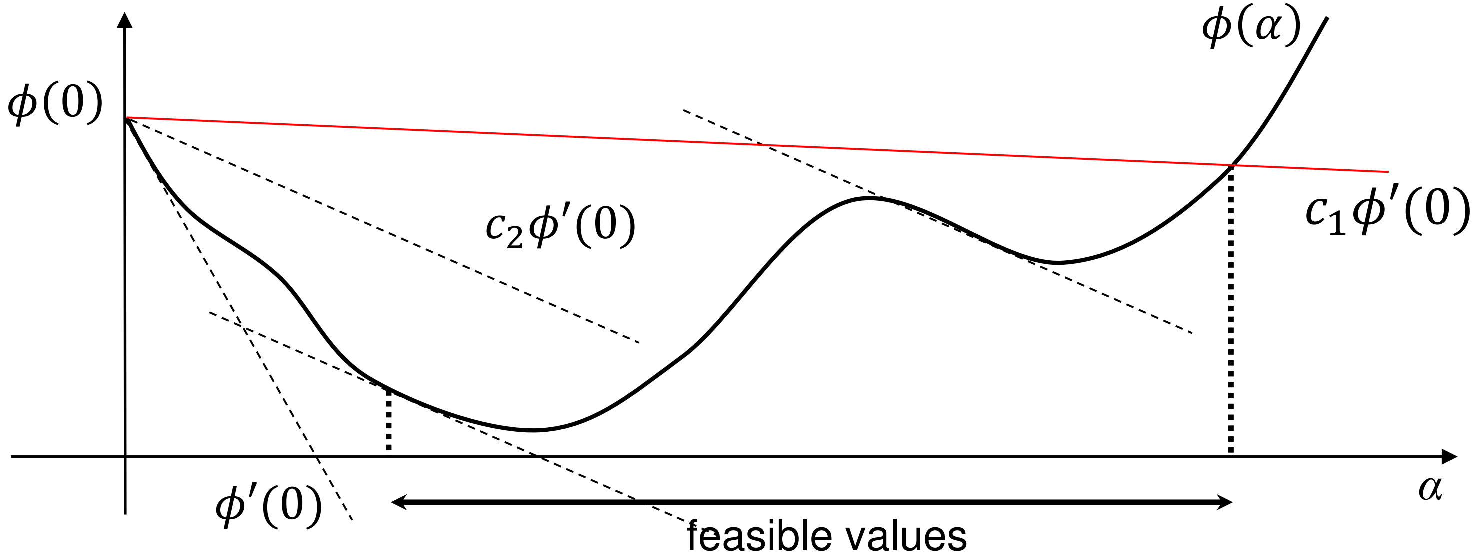
Theorem[1]:

Let f be continuously differentiable, $\boldsymbol{p}^{(k)}$ a descent direction and $c_1 \in (0,1)$, $c_2 \in (c_1, 1)$. Then, there exists an $\alpha > 0$, such that $\phi(\alpha) \le \phi(0) + \alpha c_1 \phi'(0)$

$\phi'(\alpha) \ge c_2 \phi'(0)$ (slope condition)

Geometric interpretation:

→ guarantee minimum step length!



Relevance:

Wolfe Conditions promote convergence to a stationary point[1]

[1] Nocedal J., Wright S. J., Numerical Optimization, 2nd Edition, Springer, 2006

Check Yourself

- Explain the basic ideas of the line-search method.
- What is a descent direction? How it is defined?
- Explain the Armijo-rule and its potential drawbacks?
- Explain the Wolfe conditions and the advantage compared to Armijo's rule.

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Line search: simple directions

Aachener Verfahrenstechnik

RWTH AACHEN UNIVERSITY

Solution Methods for Unconstrained Optimization

Unconstrained optimization methods

- **Direct methods**

  Iterative descent

  - **Line-search approach**
  - **Trust-region approach**

- **Indirect methods**

  Optimal solution is found by solving the system of equations (**optimality conditions**):

  $$\boldsymbol{\nabla} f(\boldsymbol{x}) = \mathbf{0}$$

  analytically or numerically.

Determination of a Descent Direction: A Toolbox

Line-search approaches differ from each other with respect to the determination of descent direction and step length.

Line-search approach

Determine the direction $\boldsymbol{p}^{(k)}$

- Steepest-descent direction
- Conjugate directions
- Newton's-step direction
- … special cases

Many gradient methods use a symmetric positive definite matrix $\boldsymbol{D}^{(k)}$ and calculate

$$\boxed{\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{D}^{(k)} \nabla f(\boldsymbol{x}^{(k)})}$$

Extra work: *prove that it guarantees descent!*

Determination of a Descent Direction: A Toolbox

Line-search approaches differ from each other with respect to the determination of descent direction and step length.

Line-search approach

Determine the direction $\boldsymbol{p}^{(k)}$

- Steepest-descent direction
- Conjugate directions
- Newton's-step direction
- … special cases

Many gradient methods use a symmetric positive definite matrix $\boldsymbol{D}^{(k)}$ and calculate

$$\boxed{\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{D}^{(k)} \boldsymbol{\nabla} f(\boldsymbol{x}^{(k)})}$$

Steepest-Descent Direction (1)

Taylor series: $f\left(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{p}^{(k)}\right) = f\left(\boldsymbol{x}^{(k)}\right) + \boxed{\alpha \boldsymbol{\nabla} f\left(\boldsymbol{x}^{(k)}\right)^T \boldsymbol{p}^{(k)}} + O(\alpha^2)$

The rate of change of f at $\boldsymbol{x}^{(k)}$ along the direction $\boldsymbol{p}^{(k)}$ is the coefficient in the linear term:

$$\boldsymbol{\nabla} f\left(\boldsymbol{x}^{(k)}\right)^T \boldsymbol{p}^{(k)}$$

The unit direction $\boldsymbol{p}^{(k)}$ with the **highest rate of change** is the solution of the following problem

$$\min_{\boldsymbol{p}^{(k)} \in R^n} \boldsymbol{\nabla} f\left(\boldsymbol{x}^{(k)}\right)^T \boldsymbol{p}^{(k)} \quad \text{s.t.} \left\|\boldsymbol{p}^{(k)}\right\| = 1$$

Note that $\boldsymbol{\nabla} f(\boldsymbol{x}^{(k)})^T \boldsymbol{p}^{(k)} = \left\|\boldsymbol{\nabla} f(\boldsymbol{x}^{(k)})\right\| \left\|\boldsymbol{p}^{(k)}\right\| \cos(\theta)$

The solution of the problem is achieved for $\cos(\theta) = -1 \Rightarrow \theta = \pi$

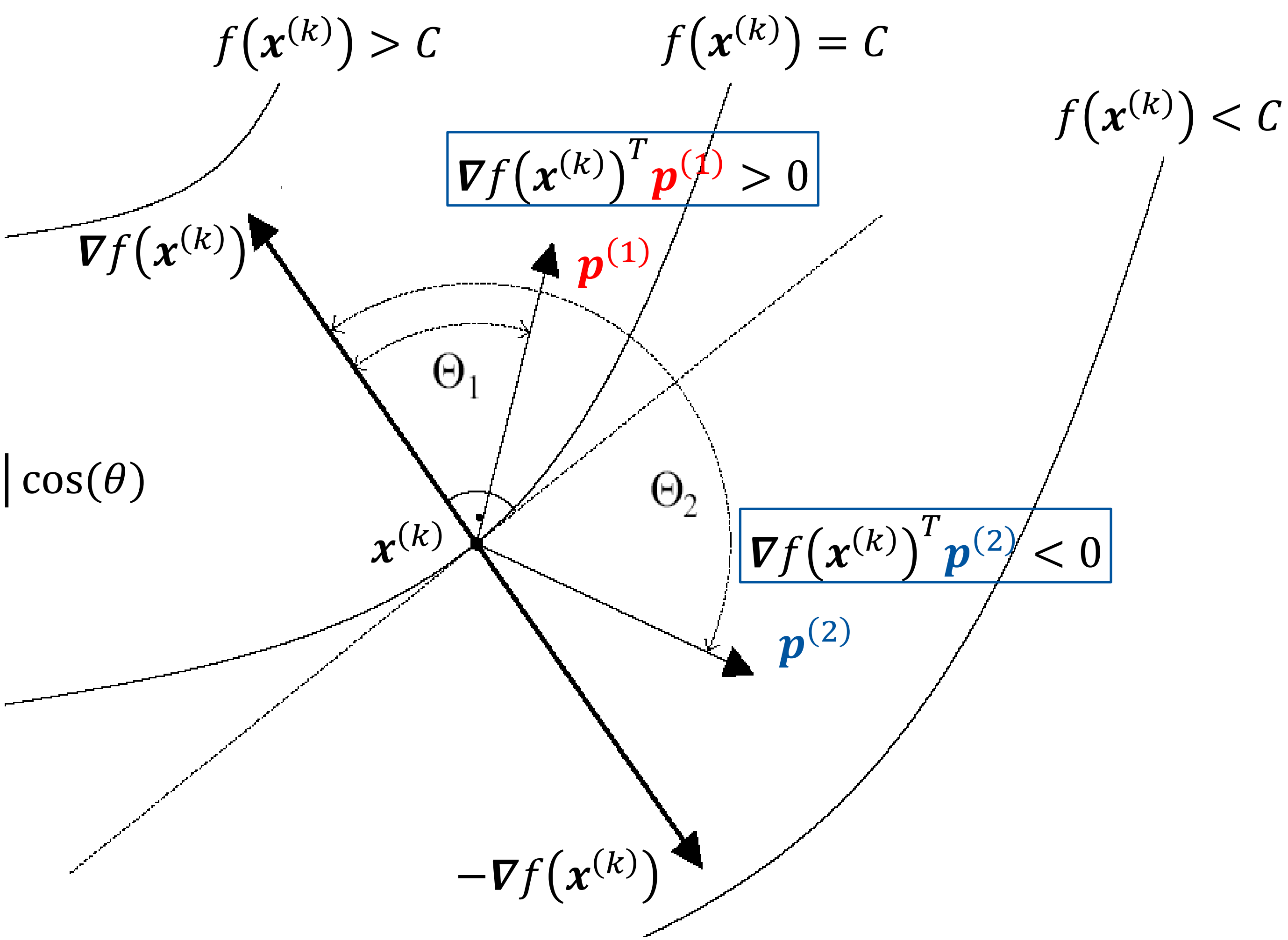
$$\Rightarrow \boldsymbol{p}^{(k)} = -\boldsymbol{\nabla} f(\boldsymbol{x}^{(k)}) / \left\|\boldsymbol{\nabla} f(\boldsymbol{x}^{(k)})\right\|$$

The choice of $\boldsymbol{D}^{(k)}$ is the identity matrix $\boldsymbol{I}$.

Steepest-Descent Direction (2)

descent direction: $\nabla f\left(\boldsymbol{x}^{(k)}\right)^T \boldsymbol{p}^{(k)} < 0$

$$\nabla f(\boldsymbol{x}^{(k)})^T \boldsymbol{p}^{(k)} = \left\|\nabla f(\boldsymbol{x}^{(k)})\right\| \left\|\boldsymbol{p}^{(k)}\right\| \cos(\theta)$$



Method of Steepest-Descent

Algorithm:

choose $\boldsymbol{x}^{(0)}$

for k=0,1,…

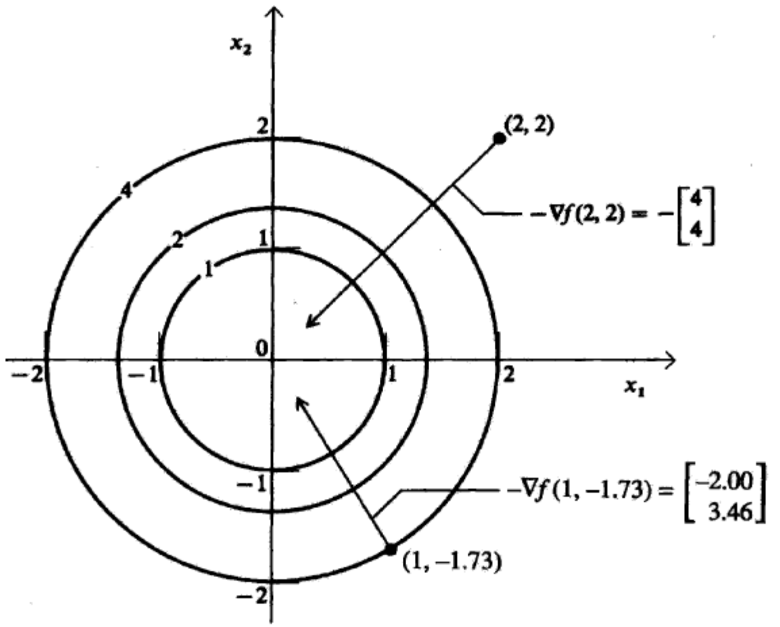
if $\left\|\boldsymbol{\nabla} f\left(\boldsymbol{x}^{(k)}\right)\right\| \leq \varepsilon$ stop, else

set $\boldsymbol{p}^{(k)} = -\boldsymbol{\nabla} f\left(\boldsymbol{x}^{(k)}\right)$

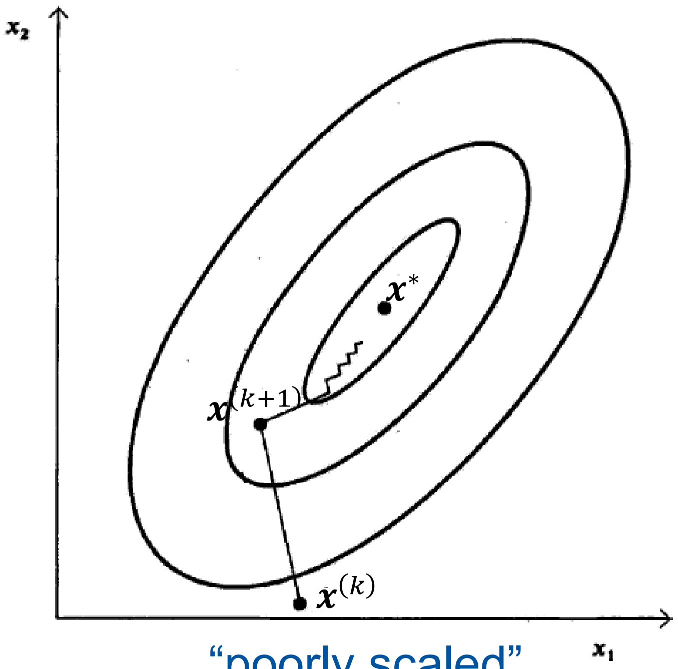
determine the step length α_k (e.g. using the Armijo rule)

set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)}$

end for



"well scaled"



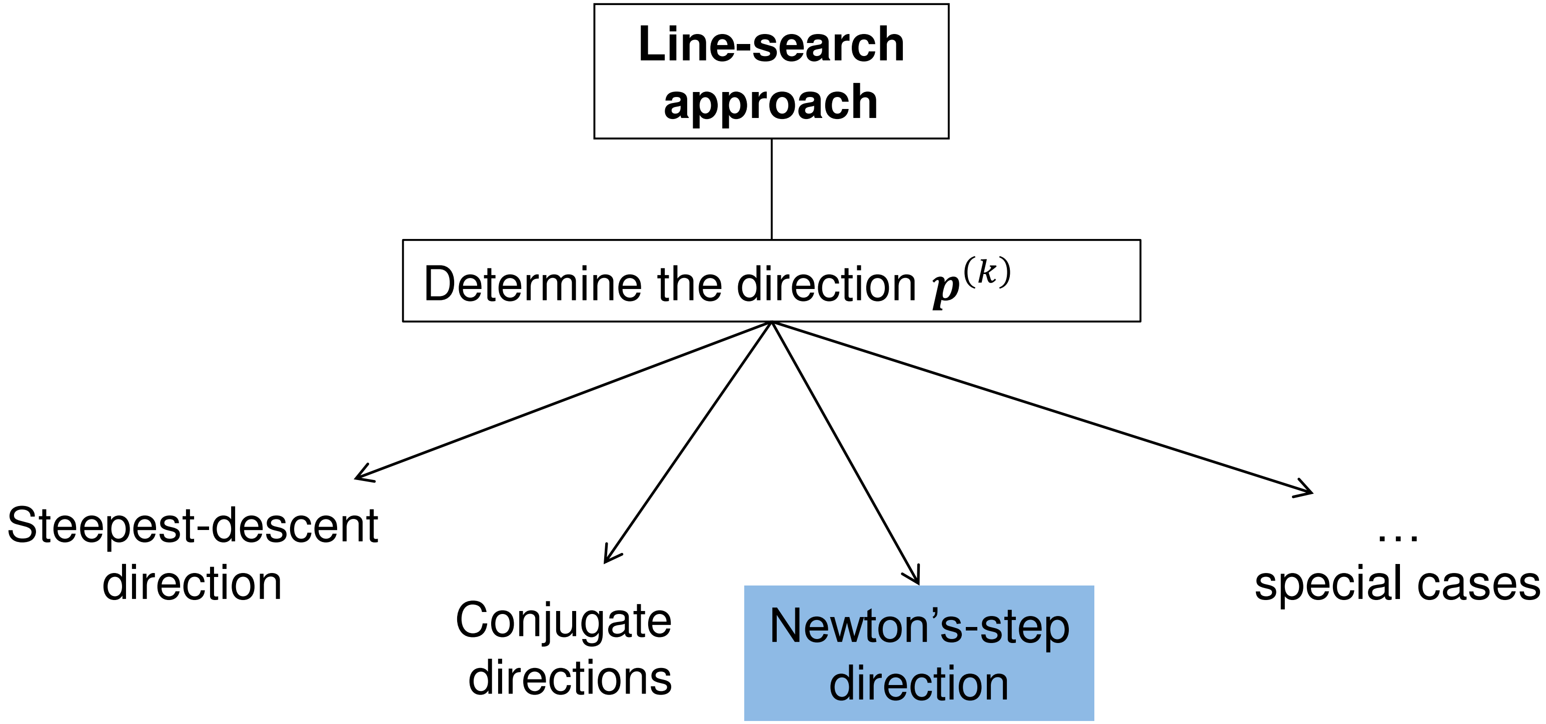
"poorly scaled"

Directions become perpendicular

Edgar T. F., Himmelblau D. M., Optimization of Chemical Processes, 2nd Edition, McGraw-Hill, 2001

Determination of a Descent Direction: A Toolbox

Line-search approaches differ from each other with respect to the determination of descent direction and step length.



Many gradient methods use a symmetric positive definite matrix $\boldsymbol{D}^{(k)}$ and calculate

$$\boxed{\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{D}^{(k)} \nabla f(\boldsymbol{x}^{(k)})}$$

Newton's Descent Direction

Quadratic approximation of f at $\boldsymbol{x}^{(k+1)}$

$$m\left(\boldsymbol{x}^{(k+1)}\right) = f\left(\boldsymbol{x}^{(k)}\right) + \nabla f\left(\boldsymbol{x}^{(k)}\right)^T\left(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\right) + \frac{1}{2}\left(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\right)^T \nabla^2 f\left(\boldsymbol{x}^{(k)}\right)\left(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\right)$$

(1st nec. opt. cond. for m)

$$0 = \nabla m\left(\boldsymbol{x}^{(k+1)}\right) = \nabla f\left(\boldsymbol{x}^{(k)}\right) + \nabla^2 f\left(\boldsymbol{x}^{(k)}\right)\left(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\right)$$

$$\Rightarrow \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \left[\nabla^2 f\left(\boldsymbol{x}^{(k)}\right)\right]^{-1} \nabla f\left(\boldsymbol{x}^{(k)}\right)$$

$$\Rightarrow \boldsymbol{p}^{(k)} = -\left[\nabla^2 f\left(\boldsymbol{x}^{(k)}\right)\right]^{-1} \nabla f\left(\boldsymbol{x}^{(k)}\right)$$

$$\Rightarrow \alpha_k = 1$$

The choice of $\boldsymbol{D}^{(k)}$ is the inverse of the Hessian

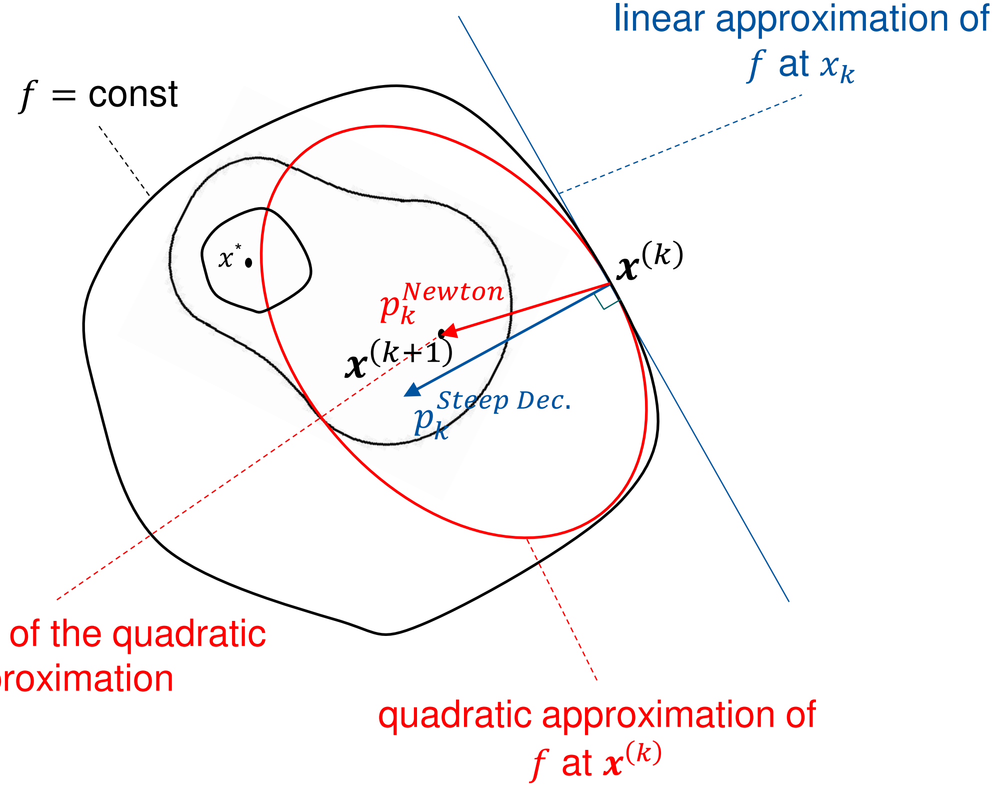


Fig.: Comparison of steepest-descent with Newton's method from viewpoint of objective function approximation

Newton's Method

Algorithm:

choose $\boldsymbol{x}^{(0)}$

for k=0,1,…

if $\|\nabla f(\boldsymbol{x}^{(k)})\| \leq \varepsilon$ stop, else

set $\boldsymbol{p}^{(k)} = -\left[\nabla^2 f\left(\boldsymbol{x}^{(k)}\right)\right]^{-1} \nabla f\left(\boldsymbol{x}^{(k)}\right)$

set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{p}^{(k)}$

end for

Remarks:

1. line-search?

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)}$$
$$\boldsymbol{p}^{(k)} = -\left[\nabla^2 f\left(\boldsymbol{x}^{(k)}\right)\right]^{-1} \nabla f\left(\boldsymbol{x}^{(k)}\right)$$
$$\alpha_k = 1$$

2. (+) locally quadratic convergence, if $\boldsymbol{x}^{(k)}$ close to $\boldsymbol{x}^*$
(−) 2nd derivatives & inversion (expensive for large system of equations)

3. If f is quadratic, the algorithm converges in one iteration.

4. Convergence to a minimum is not guaranteed! Why?

Check Yourself

- Explain the basic ideas of the line-search method.
- Explain the steepest descent method.
- What additional requirements puts Newton‘s method on the objective function?
- Explain the Newton direction. Is it better than other descent directions? Why is the Newton step-length equal to one?

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Line search: complexity and examples

Aachener Verfahrenstechnik

RWTH AACHEN UNIVERSITY

Solution Methods for Unconstrained Optimization

Unconstrained optimization methods

- **Direct methods**

  Iterative descent

  - **Line-search approach**
  - **Trust-region approach**

- **Indirect methods**

  Optimal solution is found by solving the system of equations (**optimality conditions**):

  $$\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{0}$$

  analytically or numerically.

Complexity Analysis

Nesterov (2004) proves: “In **general**, optimization problems are **unsolvable**” *

Let F denote a class of problems, e.g., Lipschitz-continuous functions with Lipschitz-constant L, i.e.,
$|f(\boldsymbol{x}) - f(\boldsymbol{y})| < L||\boldsymbol{x} - \boldsymbol{y}||$, L is assumed to be fixed for all $P \in F$.

“Performance of a method M on a problem $P \in F$ is the total amount of computational effort that is required by M to solve P.” *

“To solve the problem means to find an approximate solution to P with an accuracy ε > 0.” *

For unconstrained problems, the accuracy ε > 0 can be defined as the **norm of the objective’s gradient.**

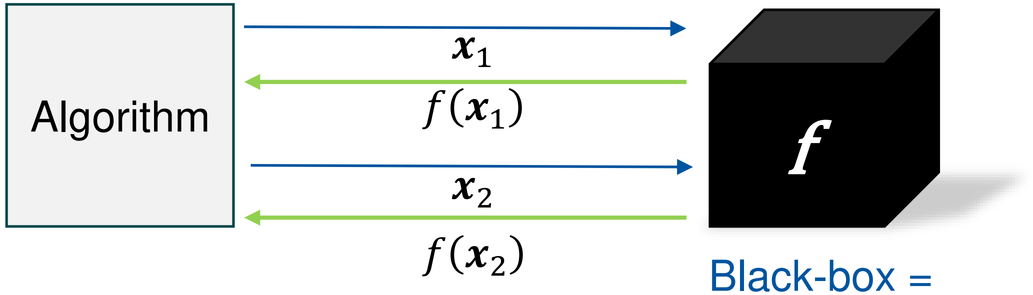
* Yurii Nesterov, *Introductory Lectures on Convex Optimization – A Basic Course*, Kluwer Academic Publishers, (2004)

Complexity Analysis – Measuring Computational Effort

Unit of measurement: *Query to an oracle*

It is assumed that the objective function is unknown and that the algorithm solves the optimization problem by *querying an oracle* for local information about the unknown objective function. An oracle is simply a “black box” capable of answering any query of the form:

- Given $\boldsymbol{x}$ return the value $f(\boldsymbol{x})$ (Zeroth-order oracle)
- Given $\boldsymbol{x}$ return $f(\boldsymbol{x})$ and gradient $\boldsymbol{\nabla} f(\boldsymbol{x})$ (First-order oracle)
- Given $\boldsymbol{x}$ return $f(\boldsymbol{x})$, $\boldsymbol{\nabla} f(\boldsymbol{x})$ and Hessian $\boldsymbol{\nabla}^2 f(\boldsymbol{x})$ (Second-order oracle)



Black-box = “Oracle”

Analytical Complexity: The smallest number of queries to an oracle to solve Problem P to accuracy ε. [1]

Arithmetical Complexity: The smallest number of arithmetic operations (including work of the oracle and work of method), required to solve problem P up to accuracy ε. [1]

[1] Yurii Nesterov, *Introductory Lectures on Convex Optimization – A Basic Course*, Kluwer Academic Publishers, 2004.

Analytical Complexity of Steepest Descent Method

Algorithm:

choose $\boldsymbol{x}^{(0)}$

for k=0,1,…

if $\left\|\boldsymbol{\nabla} f(\boldsymbol{x}^{(k)})\right\| \leq \varepsilon$ stop, else

set $\boldsymbol{p}^{(k)} = -\boldsymbol{\nabla} f\left(\boldsymbol{x}^{(k)}\right)$

determine the step length α_k (e.g. using the Armijo rule)

set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)}$

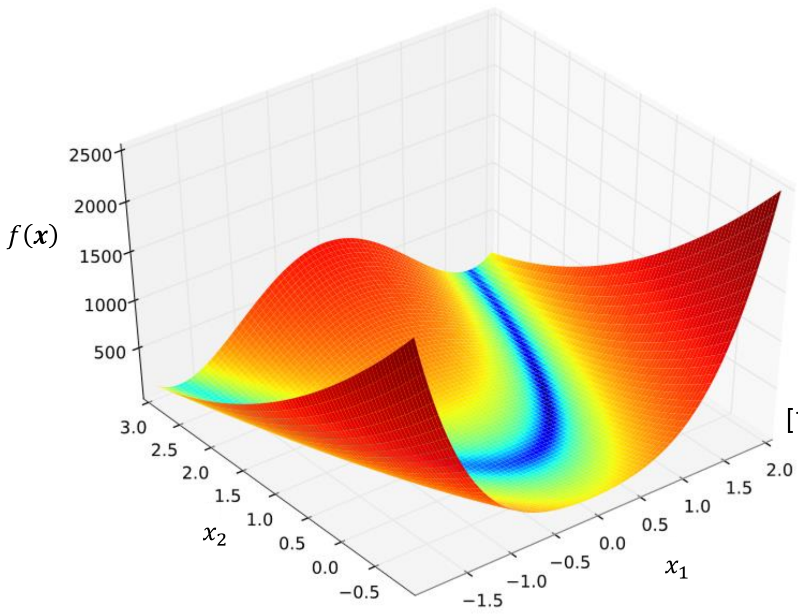
end for

- Problem class: f is continuously differentiable and $\boldsymbol{\nabla} f(\boldsymbol{x})$ is Lipschitz-continuous with fixed Lipschitz constant L, i.e., $\|\boldsymbol{\nabla} f(\boldsymbol{x}) - \boldsymbol{\nabla} f(\boldsymbol{y})\| < L\|\boldsymbol{x} - \boldsymbol{y}\|$
- First-order oracle: returns $f(\boldsymbol{x})$ and gradient $\boldsymbol{\nabla} f(\boldsymbol{x})$
- Worst-case analytical complexity (queries to oracle): $O\left(\frac{1}{\varepsilon^2}\right)$

Rosenbrock Function

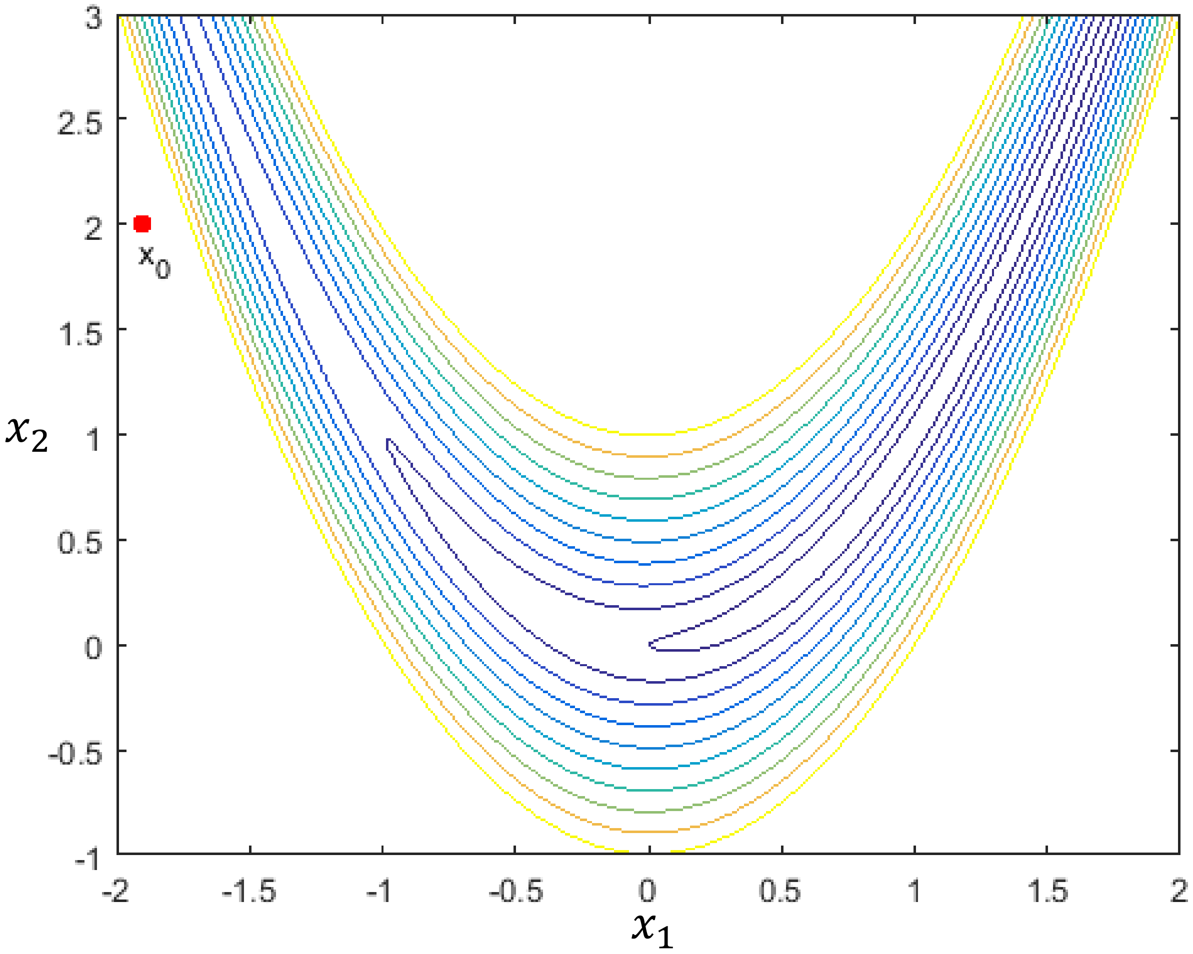
$$\min_{\boldsymbol{x}\in R^2} f(\boldsymbol{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- Solution point is $\boldsymbol{x}^* = (1,1)^T$ - why?



[1]

Contour lines of $f(\mathbf{x})$

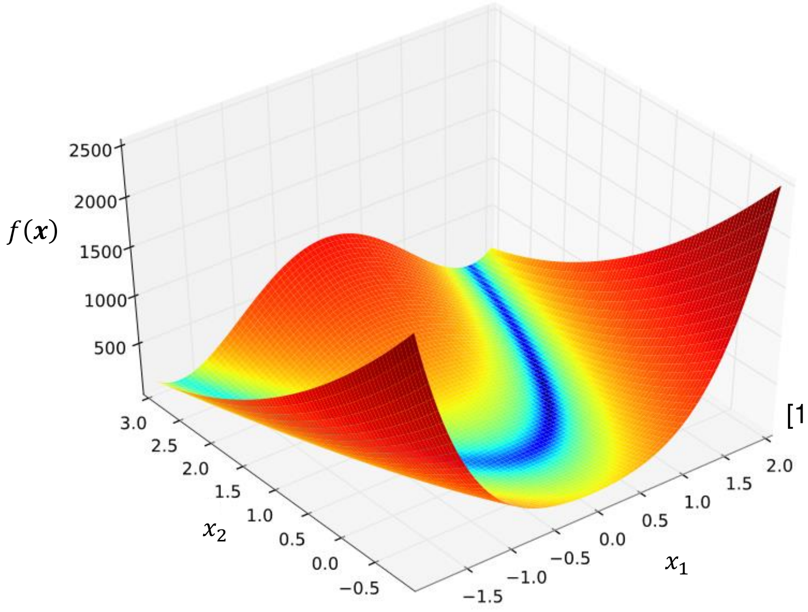


[1] https://commons.wikimedia.org/w/index.php?curid=9941741

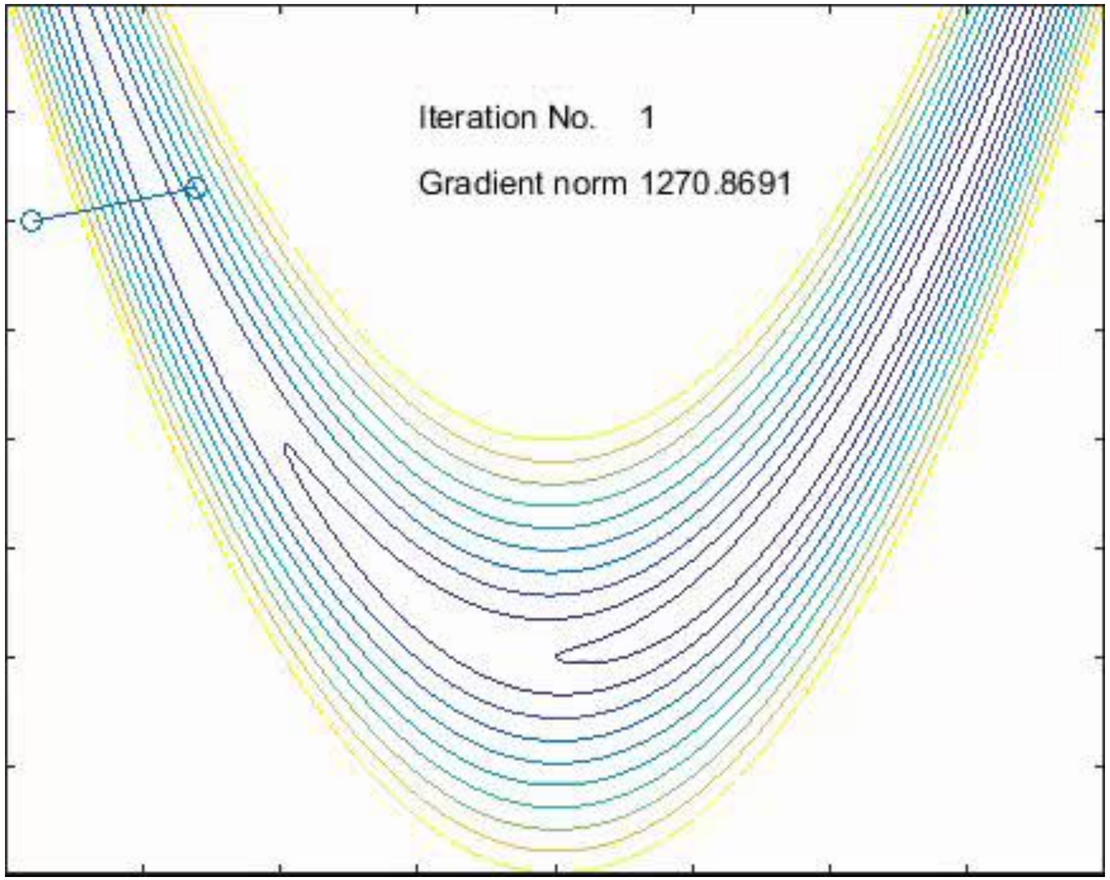
Illustration of Convergence (1)

$$\min_{\boldsymbol{x} \in R^2} f(\boldsymbol{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- Steepest descent with **Armijo** line-search



[1]

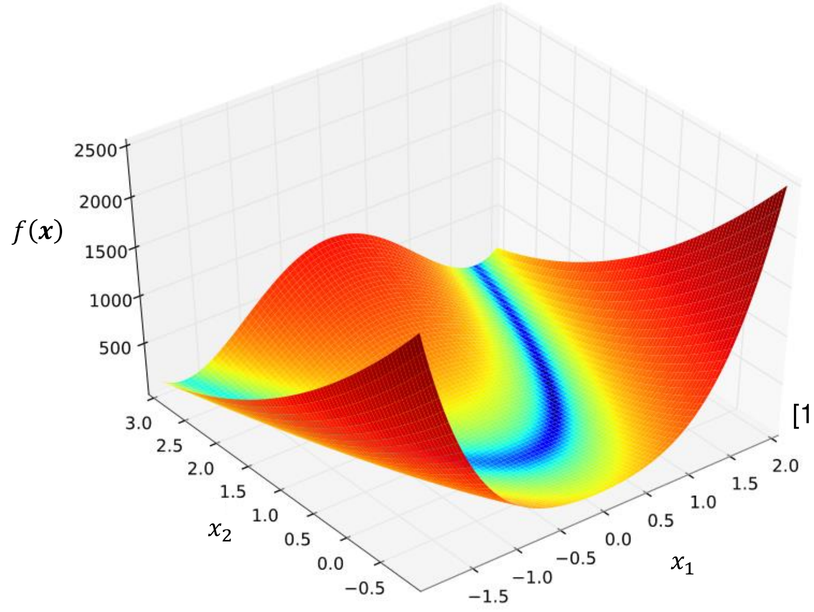


[1] https://commons.wikimedia.org/w/index.php?curid=9941741

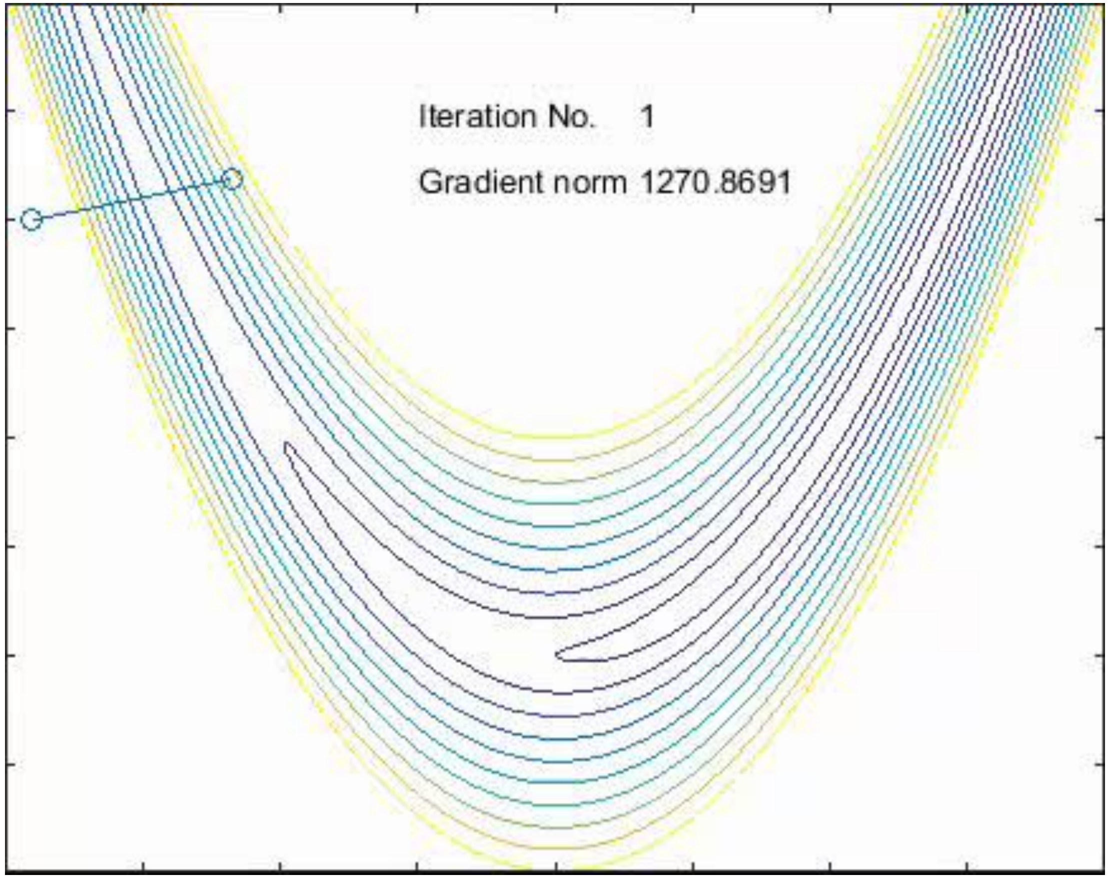
Illustration of Convergence (2)

$$\min_{\boldsymbol{x}\in R^2} f(\boldsymbol{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- Steepest descent with **Wolfe** line-search



[1]



[1] https://commons.wikimedia.org/w/index.php?curid=9941741

Analytical Complexity of Newton's Method

- Problem class: f is twice continuously differentiable and $\boldsymbol{\nabla}^2 f(\boldsymbol{x})$ is Lipschitz-continuous with fixed Lipschitz constant L, i.e., $\|\boldsymbol{\nabla}^2 f(\boldsymbol{x}) - \boldsymbol{\nabla}^2 f(\boldsymbol{y})\| < L\|\boldsymbol{x} - \boldsymbol{y}\|$
- Second-order oracle: returns $f(\boldsymbol{x})$, $\boldsymbol{\nabla} f(\boldsymbol{x})$ and Hessian $\boldsymbol{\nabla}^2 f(\boldsymbol{x})$

- Quadratic approximation of f around $\boldsymbol{x}^{(k)}$, line search

$$m(\boldsymbol{x}^{(k+1)}) = f(\boldsymbol{x}^{(k)}) + \boldsymbol{\nabla} f(\boldsymbol{x}^{(k)})^T(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}) + \frac{1}{2}(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)})^T \boldsymbol{\nabla}^2 f(\boldsymbol{x}^{(k)})(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)})$$

- Worst-case analytical complexity: $O\left(\frac{1}{\varepsilon^{2-\tau}}\right)$, $1 > \tau > 0$, arbitrary but fixed for a given problem

- Quadratic approximation of f around $\boldsymbol{x}^{(k)}$ with cubic regularization, line search

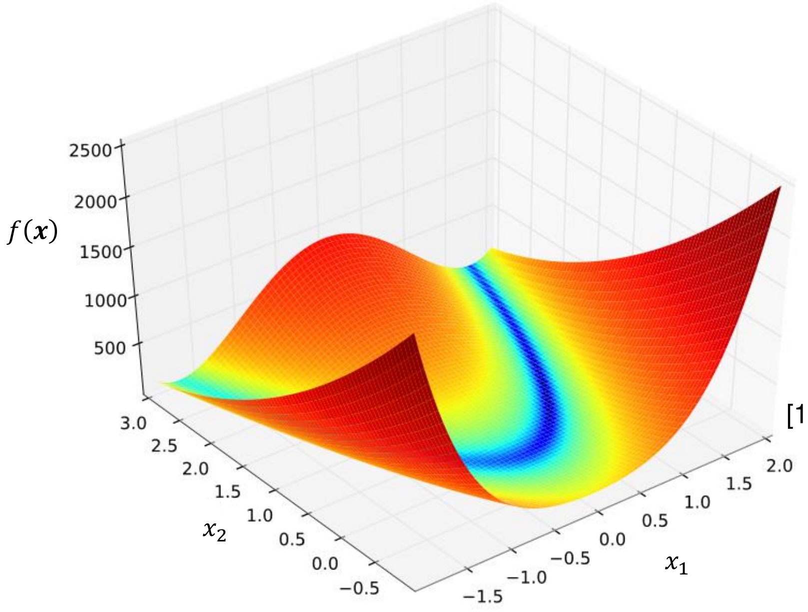
$$m_{regularized}(\boldsymbol{x}^{(k+1)}) = m(\boldsymbol{x}^{(k+1)}) + \frac{1}{3}\sigma_k \|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|^3$$

- Worst-case analytical complexity: $O\left(\frac{1}{\varepsilon^{3/2}}\right)$

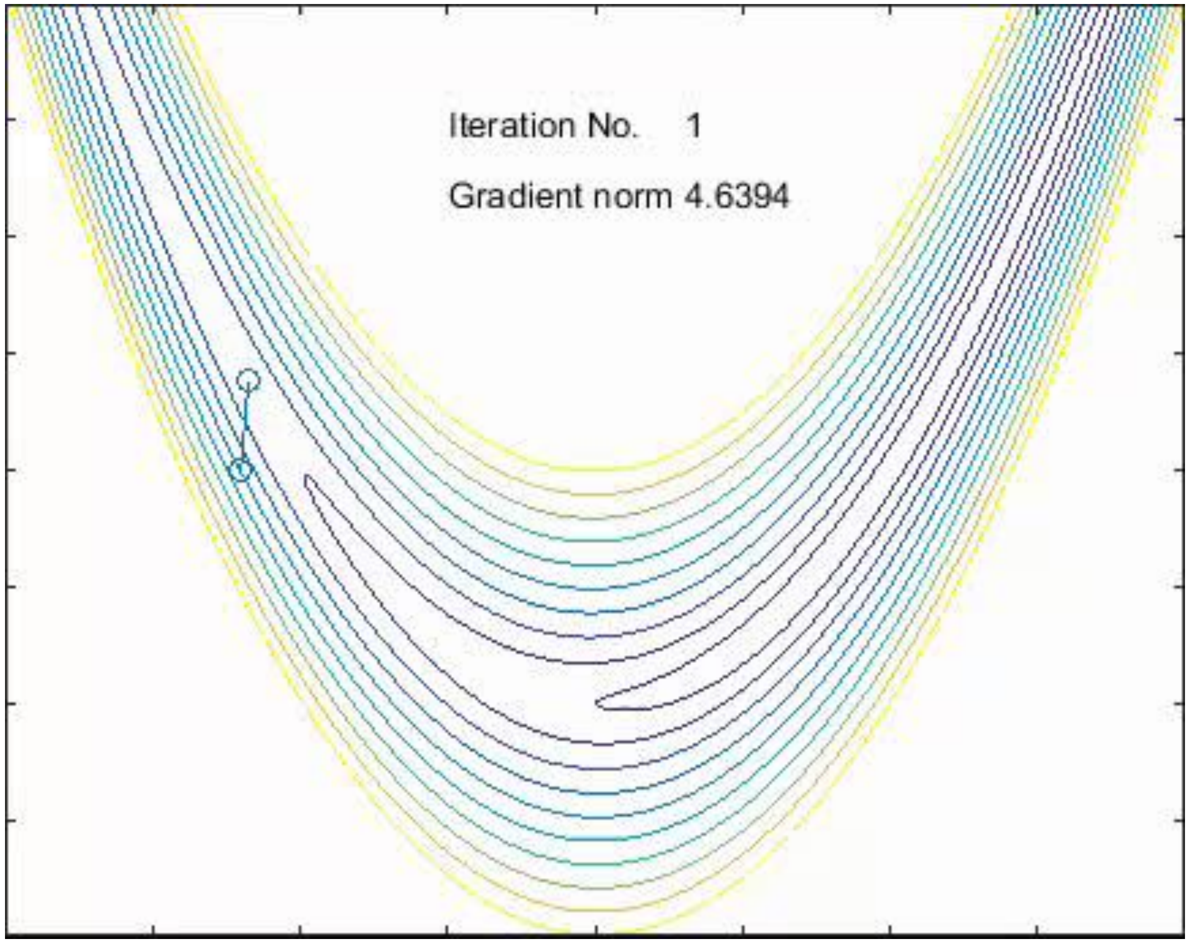
Illustration of Convergence (3)

$$\min_{\boldsymbol{x} \in R^2} f(\boldsymbol{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- **Modified Newton** (Armijo line-search; if Hessian is < 0 switch to steepest descent)



[1]



[1] https://commons.wikimedia.org/w/index.php?curid=9941741

Check Yourself

- What does the term complexity analysis refer to?
- What is the difference of analytical and arithmetic complexity
- Which method has better analytical complexity: Newton vs. steepest descent?

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Line search: advanced directions

Aachener Verfahrenstechnik

RWTH AACHEN UNIVERSITY

Solution Methods for Unconstrained Optimization

Unconstrained optimization methods

- **Direct methods**

  Iterative descent

  - **Line-search approach**
  - **Trust-region approach**

- **Indirect methods**

  Optimal solution is found by solving the system of equations (**optimality conditions**):

  $$\boldsymbol{\nabla} f(\boldsymbol{x}) = \mathbf{0}$$

  analytically or numerically.

Determination of a Descent Direction: A Toolbox

Line-search approaches differ from each other with respect to the determination of descent direction and step length.

Line-search approach

Determine the direction $\boldsymbol{p}^{(k)}$

- Steepest-descent direction
- Conjugate directions
- Newton's-step direction → variants
- … special cases

Aachener Verfahrenstechnik

RWTH AACHEN UNIVERSITY

Inexact Newton Method (1)

Define: $f^{(k)} = f\left(\boldsymbol{x}^{(k)}\right)$ and $\boldsymbol{g}^{(k)} \coloneqq \boldsymbol{\nabla} f(\boldsymbol{x}^{(k)})$ and $\boldsymbol{H}^{(k)} \coloneqq \boldsymbol{\nabla}^2 f\left(\boldsymbol{x}^{(k)}\right)$

From Newton's method:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{p}^{(k)}$$

$$\left[\boldsymbol{\nabla}^2 f\left(\boldsymbol{x}^{(k)}\right)\right]\boldsymbol{p}^{(k)} = -\boldsymbol{\nabla} f(\boldsymbol{x}^{(k)}) \Rightarrow \boldsymbol{H}^{(k)}\boldsymbol{p}^{(k)} = -\boldsymbol{g}^{(k)}$$

Idea:

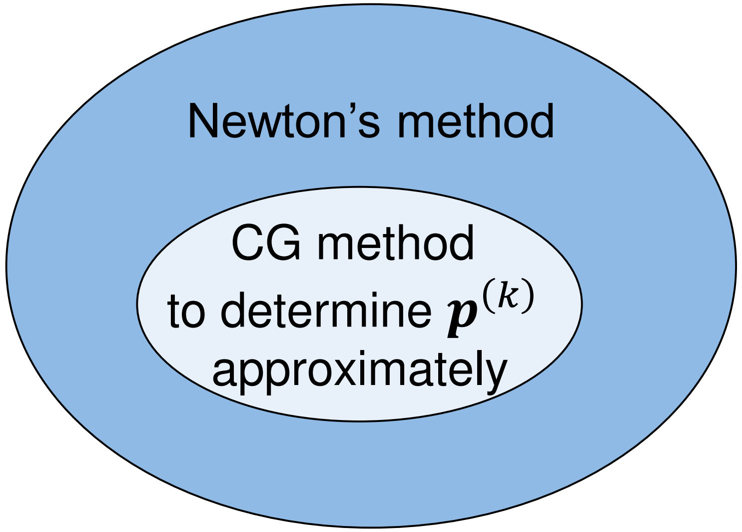
- The linear equation system, $\boldsymbol{H}^{(k)}\boldsymbol{p}^{(k)} = -\boldsymbol{g}^{(k)}$, is solved approximately by an iterative method, e.g., by CG (conjugate gradients) if $\boldsymbol{H}^{(k)}$ is positive definite.

Comments:

- LU- or Cholesky-decomposition – very high computational effort!
- Large errors occur for ill-conditioned problems.
- The exact solution is not needed.

Inexact Newton Method (2)

Newton-CG method:



Algorithm:

choose $\boldsymbol{x}^{(0)}$

for $k=0,1,\ldots$

  if $\left\|\boldsymbol{\nabla} f\left(\boldsymbol{x}^{(k)}\right)\right\| \leq \varepsilon$ stop, else

    calculate $\boldsymbol{g}^{(k)} \coloneqq \boldsymbol{\nabla} f\left(\boldsymbol{x}^{(k)}\right)$ and $\boldsymbol{H}^{(k)} \coloneqq \boldsymbol{\nabla}^2 f\left(\boldsymbol{x}^{(k)}\right)$

    solve $\boldsymbol{H}^{(k)}\boldsymbol{p}^{(k)} = -\boldsymbol{g}^{(k)}$ for $\boldsymbol{p}^{(k)}$ with CG method

    set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)}$

end for

Some line search strategy is needed. (Why?)

Modified Newton Method

Motivation: What if

- $\boldsymbol{H}^{(k)}$ is singular or almost singular (poorly conditioned)?
- $\boldsymbol{H}^{(k)}$ is not positive definite?

$$\boxed{\boldsymbol{H}^{(k)} := \nabla^2 f\left(\boldsymbol{x}^{(k)}\right)}$$

Idea	Approximations
replace $\boldsymbol{H}^{(k)}$ by the approximation $\boldsymbol{B}^{(k)} \approx \boldsymbol{H}^{(k)}$ $\Rightarrow$ $\boldsymbol{B}^{(k)}\boldsymbol{p}^{(k)} = -\boldsymbol{g}^{(k)}$ $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)}$, ($\alpha_k$ from the line-search)	$\boldsymbol{B}^{(k)} = \boldsymbol{H}^{(k)} + \boldsymbol{E}^{(k)}$ with $\boldsymbol{E}^{(k)} = \tau_k \boldsymbol{I}$, $\tau_k \geq 0$ smartly chosen converges to steepest descent for $\tau_k \to \infty$

Alternatives exist, e.g., see [1]

[1] Nocedal J., Wright S. J., Numerical Optimization, 2nd Edition, Springer, 2006, Chapter 3

Quasi-Newton Methods (1)

Idea: Reduce complexity by simplified calculation of $\boldsymbol{H}^{(k)}$ (Davidon):

- replace $\boldsymbol{H}^{(k)}$ by an approximation $\boldsymbol{B}^{(k)}$.
- instead of calculating $\boldsymbol{B}^{(k)}$, we look for a simple update using information from the last iterations.

$\boxed{\boldsymbol{H}^{(k)} := \boldsymbol{\nabla}^2 f\left(\boldsymbol{x}^{(k)}\right)}$

$\boxed{\boldsymbol{g}^{(k)} := \boldsymbol{\nabla} f\left(\boldsymbol{x}^{(k)}\right)}$

$\boxed{f^{(k)} := f\left(\boldsymbol{x}^{(k)}\right)}$

Approach:

- Consider quadratic approximation of f at $\boldsymbol{x}^{(k)}$, $m^{(k)}(\boldsymbol{p}) = f^{(k)} + \boldsymbol{g}^{(k)^T}\boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^T\boldsymbol{B}^{(k)}\boldsymbol{p}$.

symmetric positive definite

- First order optimality condition: $\boldsymbol{p}^{(k)} = -\boldsymbol{B}^{(k)^{-1}}\boldsymbol{g}^{(k)}$
- By convexity necessary and sufficient for minimization of $m^{(k)}(\boldsymbol{p})$.

- Construct the quadratic approximation at $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k\boldsymbol{p}^{(k)}$,

$$m^{(k+1)}(\boldsymbol{p}) = f^{(k+1)} + \boldsymbol{g}^{(k+1)^T}\boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^T\boldsymbol{B}^{(k+1)}\boldsymbol{p}$$

- What conditions must $\boldsymbol{B}^{(k+1)}$ satisfy?

Quasi-Newton Methods (2)

Conditions on $\boldsymbol{B}^{(k+1)}$:

1. Gradient of $m^{(k+1)}$ at $\boldsymbol{x}^{(k)}$ and $\boldsymbol{x}^{(k+1)}$ must be equal to gradient of f.

<table>
<tr><td colspan="2">$\nabla m^{(k+1)}(\boldsymbol{p}) = \boldsymbol{g}^{(k+1)} + \boldsymbol{B}^{(k+1)}\boldsymbol{p}$</td></tr>
<tr><td>At $\boldsymbol{x} = \boldsymbol{x}^{(k+1)},\ \boldsymbol{p} = 0$
We want $\nabla m^{(k+1)}(0) = \boldsymbol{g}^{(k+1)}$</td><td>At $\boldsymbol{x} = \boldsymbol{x}^{(k)},\ \boldsymbol{p} = -\alpha_k \boldsymbol{p}^{(k)}$
We want $\nabla m^{(k+1)}\left(-\alpha_k \boldsymbol{p}^{(k)}\right) = \boldsymbol{g}^{(k)}$</td></tr>
<tr><td>Automatically satisfied</td><td>$\Rightarrow \boldsymbol{g}^{(k+1)} - \alpha_k \boldsymbol{B}^{(k+1)}\boldsymbol{p}^{(k)} = \boldsymbol{g}^{(k)}$
$\Rightarrow \boldsymbol{B}^{(k+1)} \underbrace{\alpha_k \boldsymbol{p}^{(k)}}_{=\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)}} = \boldsymbol{g}^{(k+1)} - \boldsymbol{g}^{(k)}$
$\Rightarrow \boxed{\boldsymbol{B}^{(k+1)}\boldsymbol{s}^{(k)} = \boldsymbol{y}^{(k)}}$, where $\boldsymbol{s}^{(k)} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$ *and* $\boldsymbol{y}^{(k)} = \boldsymbol{g}^{(k+1)} - \boldsymbol{g}^{(k)}$</td></tr>
</table>

2. Since, $\boldsymbol{B}^{(k+1)}$ is symmetric positive definite: $\boldsymbol{s}^{(k)^T}\boldsymbol{B}^{(k+1)}\boldsymbol{s}^{(k)} > 0,\ \forall \boldsymbol{s}^{(k)} \neq 0 \Rightarrow \boxed{\boldsymbol{s}^{(k)^T}\boldsymbol{y}^{(k)} > 0}$

★ Wolfe conditions (line-search) guarantee these constraints for all f, even when f is non-convex.

Quasi-Newton Methods (3)

Conditions on $\boldsymbol{B}^{(k+1)}$:

$\boldsymbol{B}^{(k+1)}\boldsymbol{s}^{(k)} = \boldsymbol{y}^{(k)}$ gives many solutions for $\boldsymbol{B}^{(k+1)}$

- Unique solution: $\boldsymbol{B}^{(k+1)}$ should be close to $\boldsymbol{B}^{(k)}$

$$\min_{\boldsymbol{B}} \left\| \boldsymbol{B} - \boldsymbol{B}^{(k)} \right\|_W \quad \leftarrow \text{weighted Frobenius-Norm}$$

$$\text{s.t.} \quad \boldsymbol{B}^T = \boldsymbol{B}$$

$$\boldsymbol{B}\boldsymbol{s}^{(k)} = \boldsymbol{y}^{(k)}$$

$\|A\|_W = \left\|W^{1/2} A W^{1/2}\right\|_F$, for any W s.t. $W y_k = s_k$

$$\|\cdot\|_F^2 : R^{n\times n} \rightarrow R_{\geq 0}, \|C\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}^2$$

$$\Rightarrow \boldsymbol{B}^{(k+1)} = \left(\boldsymbol{I} - \frac{1}{\boldsymbol{y}^{(k)^T}\boldsymbol{s}^{(k)}}\boldsymbol{y}^{(k)}\boldsymbol{s}^{(k)^T}\right)\boldsymbol{B}^{(k)}\left(\boldsymbol{I} - \frac{1}{\boldsymbol{y}^{(k)^T}\boldsymbol{s}^{(k)}}\boldsymbol{s}^{(k)}\boldsymbol{y}^{(k)^T}\right) + \frac{1}{\boldsymbol{y}^{(k)^T}\boldsymbol{s}^{(k)}}\boldsymbol{y}^{(k)}\boldsymbol{y}^{(k)^T} \quad \rightarrow \text{DFP formula}$$

$$\Rightarrow \boldsymbol{B}^{(k+1)^{-1}} = \left(\boldsymbol{I} - \frac{1}{\boldsymbol{y}^{(k)^T}\boldsymbol{s}^{(k)}}\boldsymbol{s}^{(k)}\boldsymbol{y}^{(k)^T}\right)\boldsymbol{B}^{(k)^{-1}}\left(\boldsymbol{I} - \frac{1}{\boldsymbol{y}^{(k)^T}\boldsymbol{s}^{(k)}}\boldsymbol{y}^{(k)}\boldsymbol{s}^{(k)^T}\right) + \frac{1}{\boldsymbol{y}^{(k)^T}\boldsymbol{s}^{(k)}}\boldsymbol{s}^{(k)}\boldsymbol{s}^{(k)^T} \quad \rightarrow \text{BFGS formula}$$

Check Yourself

- Explain the inexact Newton method.
- What is the main idea of the modified and quasi-Newton methods? Why are these methods advantageous?
- Why is it necessary to introduce a step-length control mechanism (line-search) into modified and quasi-Newton methods?

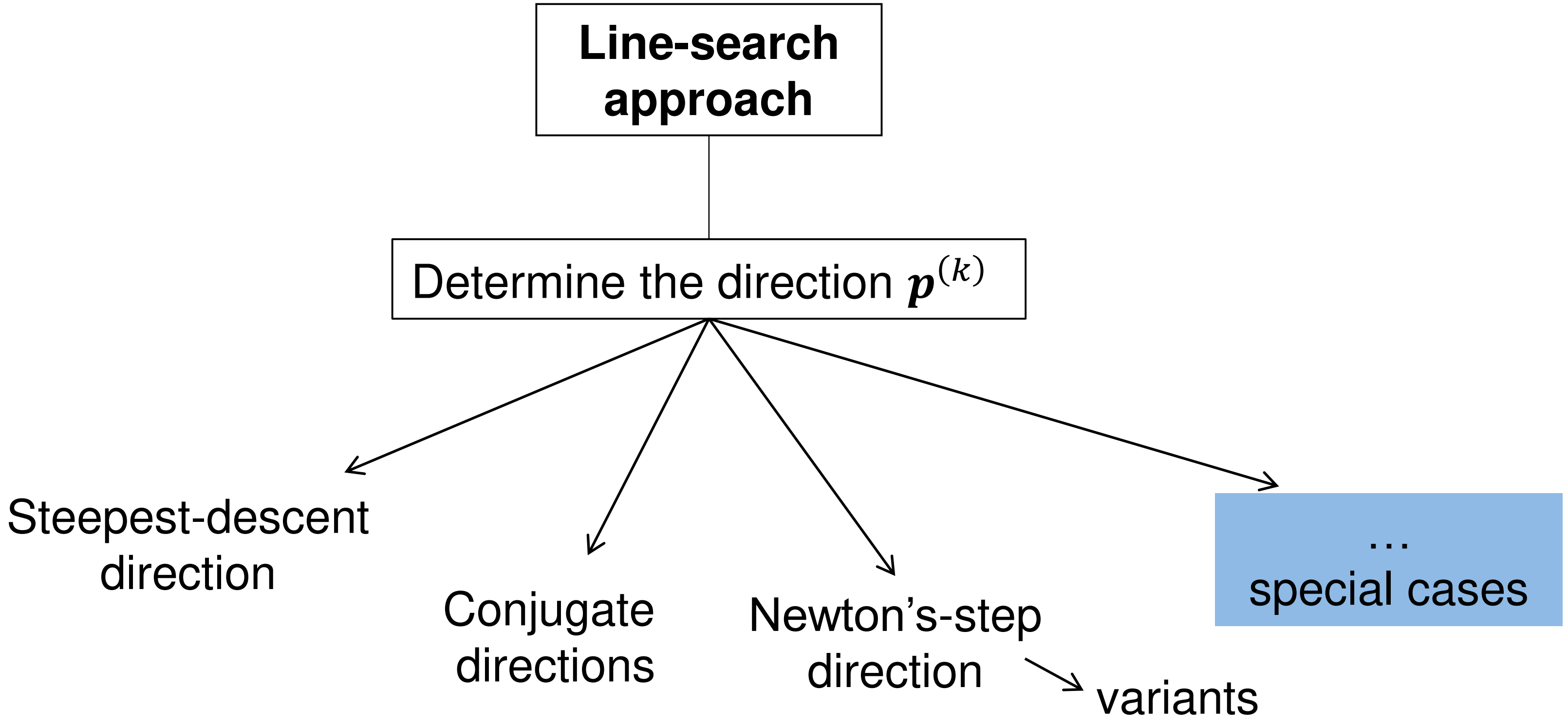
Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Parameter estimation

Determination of a Descent Direction: A Toolbox

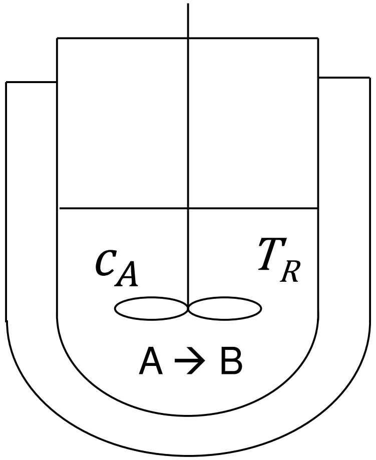
Line-search approaches differ from each other with respect to the determination of descent direction and step length.



Regression Problems: Least-Squares Formulation

Example:

Consider a batch reactor with the reaction $A \rightarrow B$ at constant temperature T_R. The reagent concentration c_A is measured at time instants t_j.
The reaction is of first order, therefore we can write the analytic solution:



$$\left.\frac{dc_A}{dt}\right|_t = -k \cdot c_A(t) \quad \rightarrow \quad c_A(t) = c_A|_{t=0} \cdot e^{-kt}$$

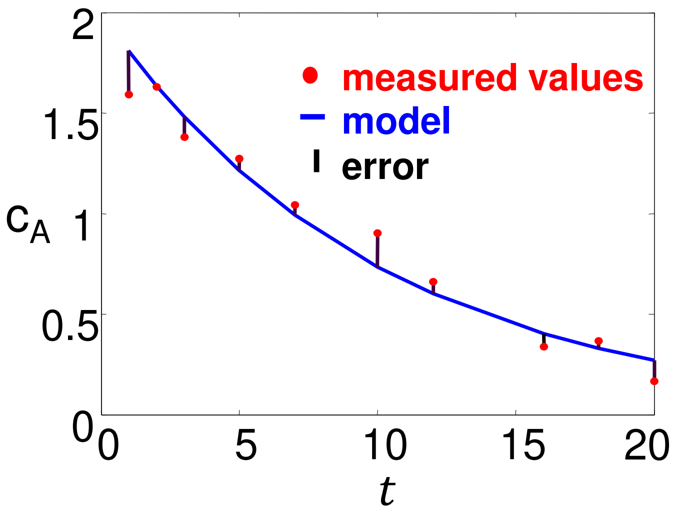
The reaction constant k and the reagent concentration at initial time $c_A(t=0)$ are unknown and should be determined from the measurements.
Optimization formulation uses $x_1 = c_A(t=0), x_2 = k$:

$$c_{A,\,\text{theoretical}}(t_j) = \varphi(\boldsymbol{x}, t_j) = x_1 \cdot e^{x_2 \cdot t_j} \qquad \text{model}$$

$$c_{A,\,\text{measured}}(t_j) = y_j \qquad \text{measurement}$$

$$\varepsilon_j = y_j - \varphi(\boldsymbol{x}, t_j), \forall j = 1, \ldots, m \qquad \text{residual (error)}$$

$$\boxed{\min_{\boldsymbol{x} \in R^2} \frac{1}{2} \|\boldsymbol{\varepsilon}(\boldsymbol{x})\|_2^2 = \frac{1}{2} \Sigma_j \left(y_j - \phi(\boldsymbol{x}, t_j)\right)^2}$$



Gauss-Newton Method

- $\min_{\boldsymbol{x}\in R^2} f(\boldsymbol{x}) = \min_{\boldsymbol{x}\in R^2} \frac{1}{2}\|\boldsymbol{\varepsilon}(\boldsymbol{x})\|_2^2 = \min_{\boldsymbol{x}\in R^2} \frac{1}{2}\boldsymbol{\varepsilon}(\boldsymbol{x})^T\boldsymbol{\varepsilon}(\boldsymbol{x})$

- Define: $\boldsymbol{J}(\boldsymbol{x}) \coloneqq \boldsymbol{\nabla}\boldsymbol{\varepsilon}(\boldsymbol{x}) \in R^{m\times 2}$

$\Rightarrow \boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{J}(\boldsymbol{x})^T\boldsymbol{\varepsilon}(\boldsymbol{x})$

$\Rightarrow \boldsymbol{\nabla}^2 f(\boldsymbol{x}) = \boldsymbol{J}(\boldsymbol{x})^T\boldsymbol{J}(\boldsymbol{x}) + \cancel{\sum_{j=1}^{m} \varepsilon_j(\boldsymbol{x})\boldsymbol{\nabla}^2\varepsilon_j(\boldsymbol{x})}$

- The Hessian can be approximated by the first term in case of almost linear problems (i.e., $\boldsymbol{\nabla}^2\varepsilon_j(\boldsymbol{x}) = 0$) or good starting values (i.e., small $\varepsilon_j(\boldsymbol{x})$)

- Newton's direction: $\boldsymbol{\nabla}^2 f\left(\boldsymbol{x}^{(k)}\right)\boldsymbol{p}_k = -\boldsymbol{\nabla} f(\boldsymbol{x}^{(k)})$

- With Hessian approximation: $\boxed{\boldsymbol{J}^{(k)^T}\boldsymbol{J}^{(k)}\boldsymbol{p}^{(k)} = -\boldsymbol{J}^{(k)^T}\boldsymbol{\varepsilon}^{(k)}}$

Remarks on Gauss-Newton Method

- If $\boldsymbol{J}^{(k)}$ has full-rank, $\boldsymbol{p}^{(k)}$ is always a descent direction

$$\boldsymbol{p}^{(k)^T} \cdot \nabla f\left(\boldsymbol{x}^{(k)}\right) = \boldsymbol{p}^{(k)^T} \cdot \boldsymbol{J}^{(k)^T} \boldsymbol{\varepsilon}^{(k)} = -\boldsymbol{p}^{(k)^T} \cdot \boldsymbol{J}^{(k)^T} \cdot \boldsymbol{J}^{(k)} \cdot \boldsymbol{p}^{(k)} = -\left\|\boldsymbol{J}^{(k)} \cdot \boldsymbol{p}^{(k)}\right\|_2^2 < 0,$$

  The inequality is strict unless $\boldsymbol{J}^{(k)} \cdot \boldsymbol{p}^{(k)} = 0 \Leftrightarrow \boldsymbol{J}^{(k)^T} \boldsymbol{\varepsilon}^{(k)} = \nabla f_k = 0.$ ← Optimum

- In descent-direction $\boldsymbol{p}^{(k)}$, the step-length is determined as per the Wolfe-conditions
- For linear models the Jacobian $\boldsymbol{J}$ matrix is constant.
- The condition of minimization corresponds to the normal equations
- If $\boldsymbol{J}^{(k)}$ is singular or almost singular, the descent direction $\boldsymbol{p}^{(k)}$ is, usually, not reliable. The method converges very poorly. Quasi-Newton methods are therefore more efficient.
- It is a local method

Check Yourself

- Where can least-squares problems be applied?
- When is a least-squares problem linear or nonlinear?
- What is the key idea of the Gauss-Newton method?

Applied Numerical Optimization

Prof. Alexander Mitsos, Ph.D.

Trust region method

Solution Methods for Unconstrained Optimization

Unconstrained optimization methods

- **Direct methods**

  Iterative descent

  - **Line-search approach**
  - **Trust-region approach**

- **Indirect methods**

  Optimal solution is found by solving the system of equations (**optimality conditions**):

  $$\boldsymbol{\nabla} f(\boldsymbol{x}) = \mathbf{0}$$

  analytically or numerically.

Trust Region Method (1)

Idea:

- Approximate f at $\boldsymbol{x}^{(k)}$ by the quadratic model function $m^{(k)}$:

  $$m^{(k)}(\boldsymbol{p}) = f^{(k)} + \boldsymbol{g}^{(k)^T}\boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^T\boldsymbol{B}^{(k)}\boldsymbol{p}$$

  where $f^{(k)} = f(\boldsymbol{x}^{(k)})$, $\boldsymbol{g}^{(k)} = \boldsymbol{\nabla} f\left(\boldsymbol{x}^{(k)}\right)$ and $\boldsymbol{B}^{(k)}$ is symmetric
- Taylor-series: the approximation error is small for small $\boldsymbol{p}$
- For each iteration $k = 0,1,\ldots$ choose a trust region radius $\Delta^{(k)}$
- Solve the minimization problem: $\min_{\boldsymbol{p}} m^{(k)}(\boldsymbol{p}) \quad \text{s.t.} \|\boldsymbol{p}\| \leq \Delta^{(k)}$ and set $\boldsymbol{p}^{(k)}$ to the solution found
- Set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{p}^{(k)}$

Trust Region Method (2)

How to update the radius $\Delta^{(k)}$?

- Compare the agreement between the model function m_k and the objective function f at the previous iterations. Define contraction rate ρ_k as:

$$\rho_k = \frac{f\left(\boldsymbol{x}^{(k)}\right) - f\left(\boldsymbol{x}^{(k)} + \boldsymbol{p}^{(k)}\right)}{m^{(k)}(\boldsymbol{0}) - m^{(k)}\left(\boldsymbol{p}^{(k)}\right)} = \frac{\text{actual reduction}}{\text{predicted reduction}}$$

As m_k is minimized over a domain containing **0**:

$$m^{(k)}(\boldsymbol{0}) - m^{(k)}\left(\boldsymbol{p}^{(k)}\right) > 0$$

If $\rho_k < 0$! reject this step (ascent)

If $\rho_k \approx 1$! increase the radius: good agreement

If $\rho_k \approx 0$! decrease the radius: poor agreement

Trust Region Method (3)

Basic Algorithm:

choose $\Delta^{(max)} > 0, \Delta^{(0)} \in (0, \Delta^{(max)})$ and $\eta \in [0, \frac{1}{4})$

for $k = 0,1, \ldots$

calculate direction $\boldsymbol{p}^{(k)}$, contraction rate ρ_k

if $\rho_k < \frac{1}{4}$, $\Delta^{(k+1)} = \frac{||\boldsymbol{p}^{(k)}||}{4}$

else if $\rho_k > \frac{3}{4}$ and $||\boldsymbol{p}^{(k)}|| = \Delta^{(k)}$, $\Delta^{(k+1)} = \min(2\Delta^{(k)}, \Delta^{(max)})$

else $\Delta^{(k+1)} = \Delta^{(k)}$

if $\rho_k > \eta$, $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{p}^{(k)}$

else $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)}$

Trust Region Method (3)

Remarks:

- $\Delta^{(k)}$ is increased only if $||\boldsymbol{p}^{(k)}||$ reaches the boundary of the domain.
- Strategies for the efficient solution of the minimization problem for $\boldsymbol{p}^{(k)}$:

  - The Cauchy point: minimum along the steepest descent direction $(-\boldsymbol{g}^{(k)})$, slow

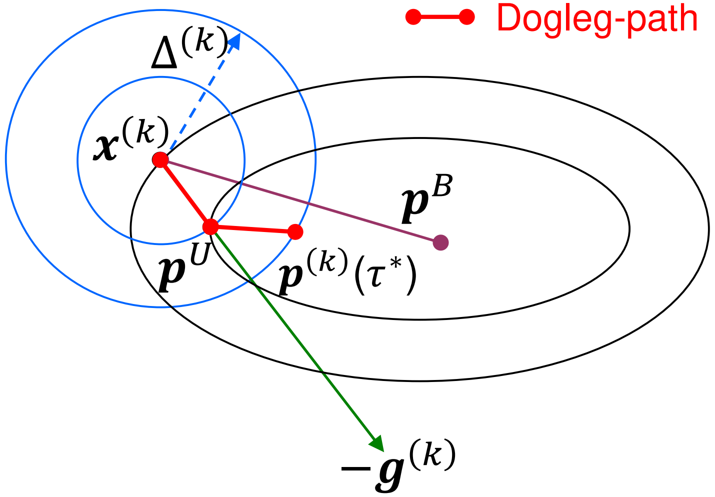
  - The *Dogleg method*: applicable when $\boldsymbol{B}^{(k)}$ is positive definite, fast (superlinear)

  - Steihaug's approach for large sparse matrices

The Dogleg Method

Idea:

- For a large $\Delta^{(k)}$: Newton step, $\boldsymbol{p}^{(k)} = \boldsymbol{p}^B = -\boldsymbol{B}^{(k)^{-1}}\boldsymbol{g}^{(k)}$. Where $\boldsymbol{p}^B$ is the unconstrained minimum of m_k, $\|\boldsymbol{p}^B\| \leq \Delta^{(k)}$.



- For a small $\Delta^{(k)}$: search the solution along the direction $-\boldsymbol{g}^{(k)}$

- For an intermediate $\Delta^{(k)}$: additionally calculate $\boldsymbol{p}^U = -\dfrac{\boldsymbol{g}^{(k)^T}\boldsymbol{g}^{(k)}}{\boldsymbol{g}^{(k)^T}\boldsymbol{B}^{(k)}\boldsymbol{g}^{(k)}}\boldsymbol{g}^{(k)}$

  Where $\boldsymbol{p}^U$ is the unconstrained minimum of $m^{(k)}$ in the steepest descent direction.

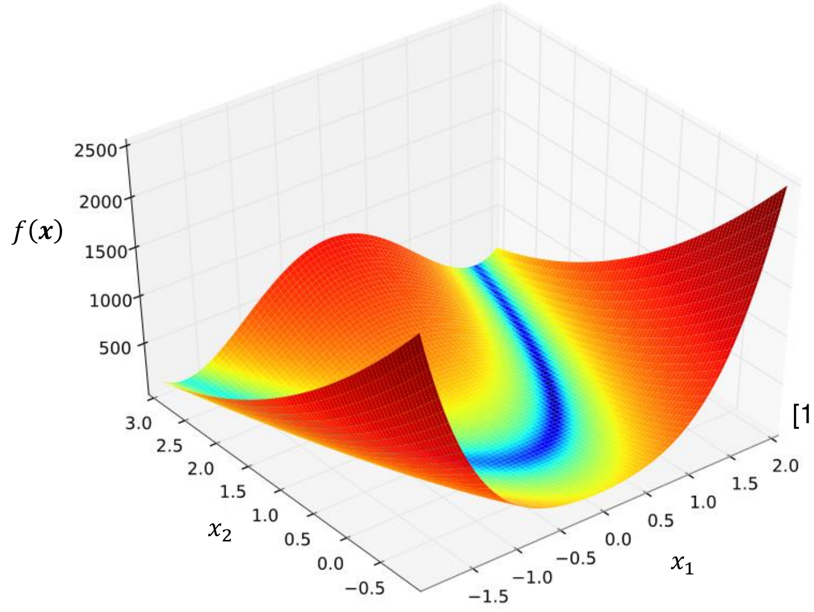
- Then calculate the search-direction using a linear combination of $\boldsymbol{p}^U$ and $\boldsymbol{p}^B$: $\boldsymbol{p}^{(k)}(\tau) = \begin{cases} \tau\boldsymbol{p}^U & 0 \leq \tau \leq 1 \\ \boldsymbol{p}^U + (\tau-1)(\boldsymbol{p}^B - \boldsymbol{p}^U) & 1 \leq \tau \leq 2 \end{cases}$

  with $\left\|\boldsymbol{p}^{(k)}(\tau^*)\right\| = \Delta^{(k)}$.

Illustration of Convergence – Rosenbrock Function

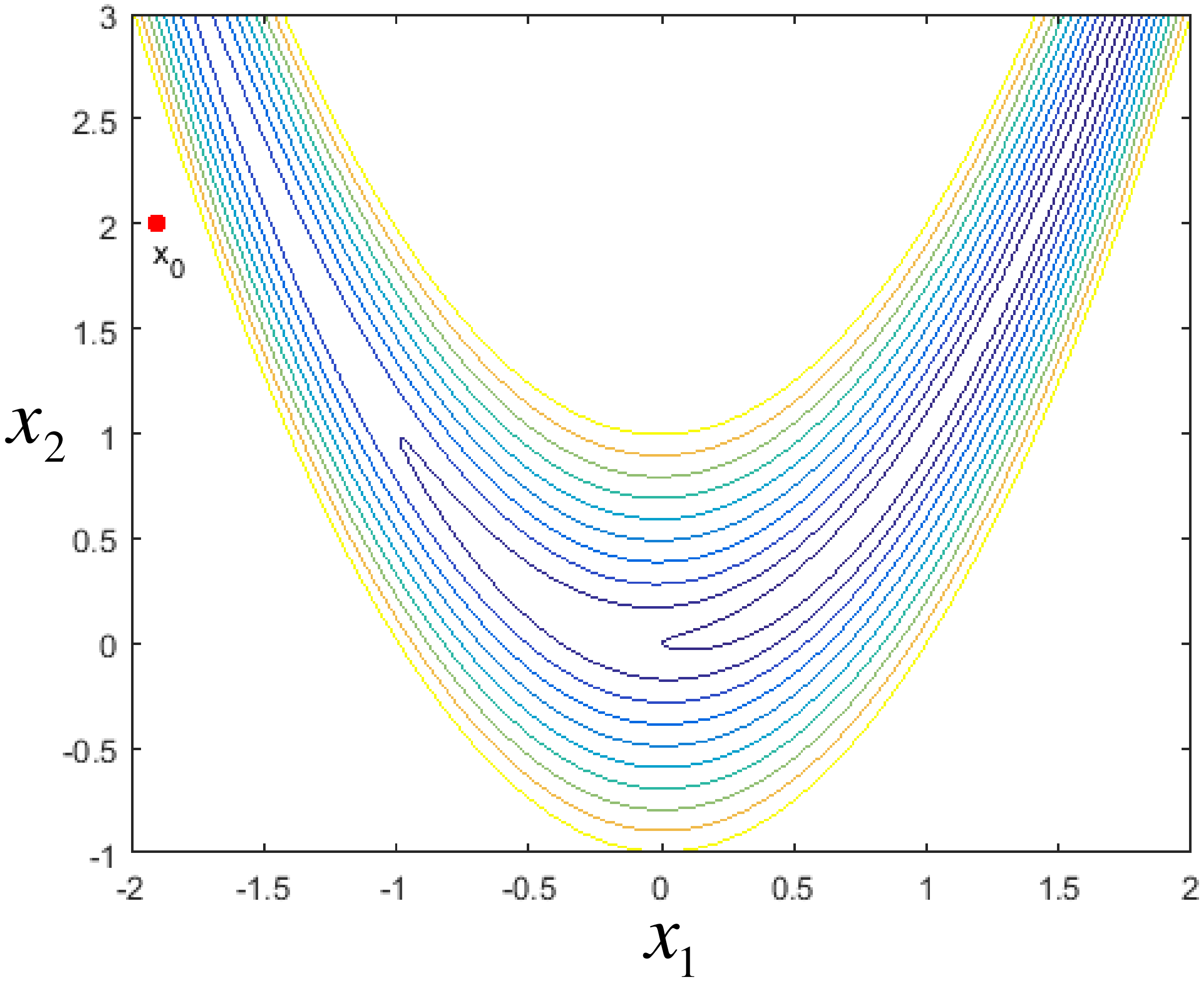
$$\min_{\boldsymbol{x}\in R^2} f(\boldsymbol{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- Solution point is $\boldsymbol{x}^* = (1,1)^T$ - why?



[1]

Contour lines of $f(\mathbf{x})$

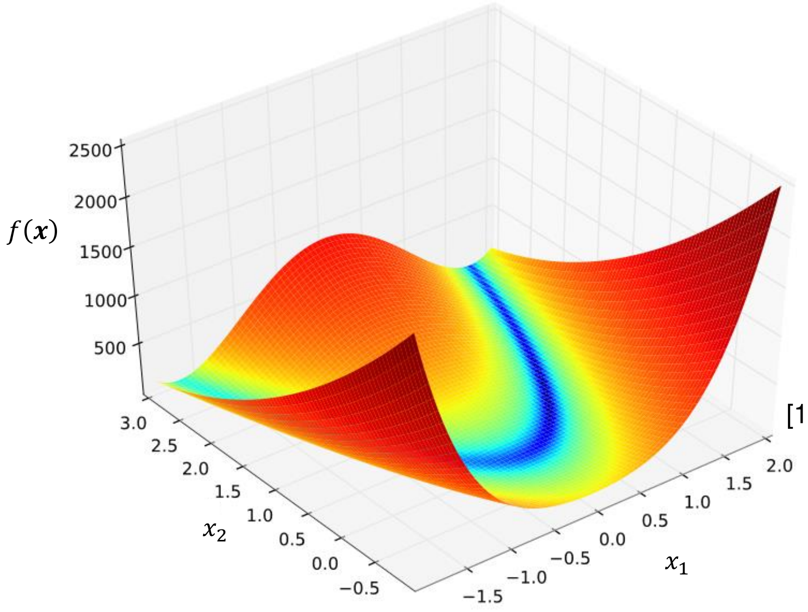


[1] https://commons.wikimedia.org/w/index.php?curid=9941741

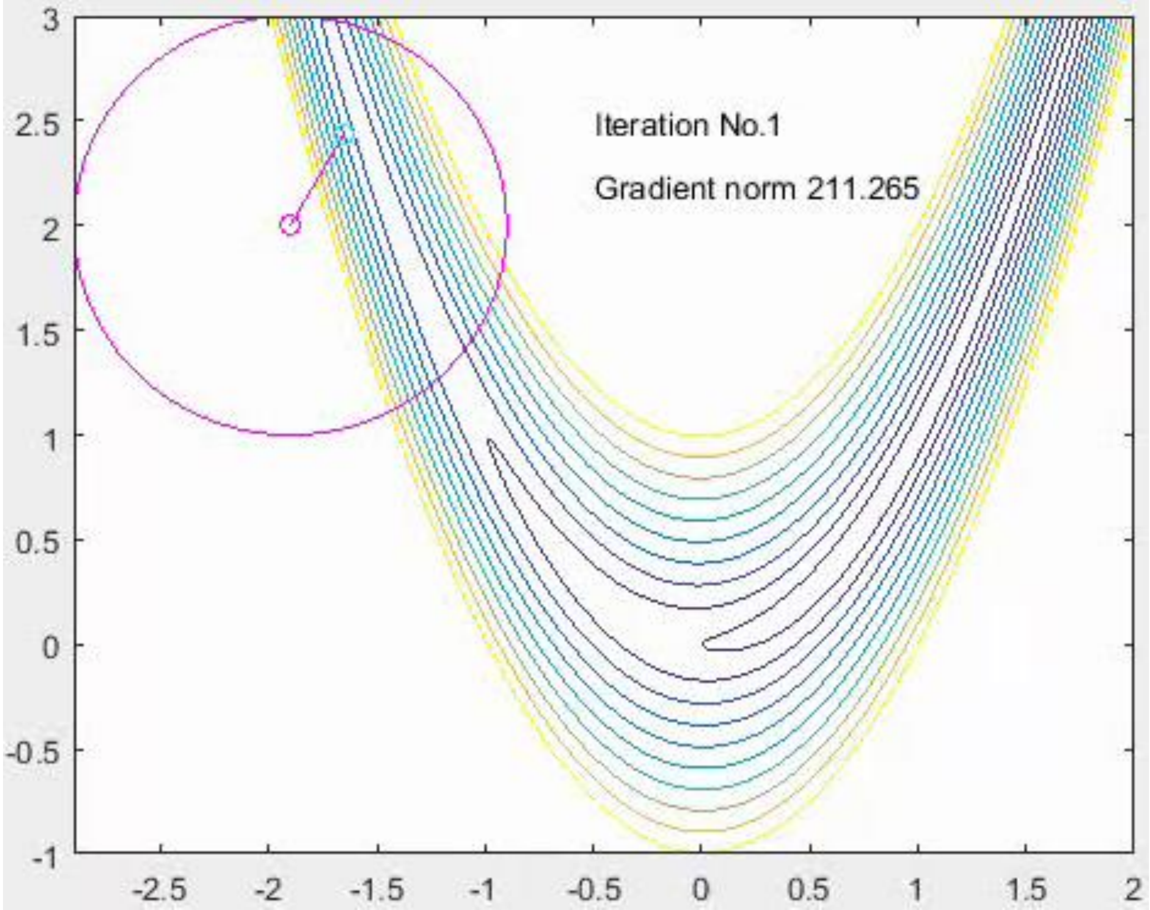
Illustration of Convergence – BFGS (Quasi-Newton Method)

$$\min_{\boldsymbol{x}\in R^2} f(\boldsymbol{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- Matlab trust-region (fminunc)



[1]



[1] https://commons.wikimedia.org/w/index.php?curid=9941741

Aachener Verfahrenstechnik

RWTH AACHEN UNIVERSITY

Check Yourself

- Explain trust region method.
- Which model problem is solved in trust-region methods?
- How is trust-region radius updated?