SI152: Numerical Optimization

Lecture 0: Introduction and Fundamentals

Hao Wang
Email: haw309@gmail.com

ShanghaiTech University

September 16, 2025

Outline

1. Course Information
2. What is this course?
3. What is optimization?
4. Examples
5. Algorithms and Convergence

Outline

- Instructor: Hao Wang
- Office: SIST-2-401
- Email: wanghao1@shanghaitech.edu.cn
- Time: 3:00pm–4:40pm, Tuesday and Thursday, week 1–12
- Location: Rm 1D-107
- Teaching assistant: Ken Chen, Yinhao Zhao, Yiran Liu
- Office hours: TBD
- Evaluation: Final 40% + Homework 30% + Project 30%
- Homework: LaTeX preferred, due in one week

Outline

Recall from your primary school

Recall from your middle school (Jiangsu)

3.3 二元一次不等式组与简单的线性规划问题

我们先考察生产中遇到的一个问题：

某工厂生产甲、乙两种产品，生产 1 t 甲种产品需要 A 种原料4 t、B 种原料 12 t，产生的利润为 2 万元；生产 1 t 乙种产品需要 A 种原料 1 t、B 种原料 9 t，产生的利润为 1 万元. 现有库存 A 种原料 10 t、B 种原料 60 t，如何安排生产才能使利润最大？

为理解题意，可将已知数据整理成下表：

	A 种原料(t)	B 种原料(t)	利 润(万元)
甲种产品(1 t)	4	12	2
乙种产品(1 t)	1	9	1
现有库存(t)	10	60	

设计划生产甲、乙两种产品的吨数分别为 x，y，利润为 P（万元）. 根据题意，A，B 两种原料分别不得超过 10 t 和 60 t，又产量不可能是负数，于是可得二元一次不等式组

$$\begin{cases} 4x+y\leqslant 10, \\ 12x+9y\leqslant 60, \\ x\geqslant 0, \\ y\geqslant 0, \end{cases}$$

即

$$\begin{cases} 4x+y\leqslant 10, \\ 4x+3y\leqslant 20, \\ x\geqslant 0, \\ y\geqslant 0. \end{cases}$$

因此，上述问题转化为如下的一个数学问题：在约束条件

$$\begin{cases} 4x+y\leqslant 10, \\ 4x+3y\leqslant 20, \\ x\geqslant 0, \\ y\geqslant 0 \end{cases}$$

下，求出 x，y，使利润

这是一个含有两个变量 x 和 y 的函数，称为目标函数.

$$P=2x+y$$

达到最大.

● 如何解决这个问题？

3.3.1 二元一次不等式表示的平面区域

我们分两步求解上面的问题：

第一步 研究问题中的约束条件，确定数对 (x,y) 的范围；

第二步 在第一步得到的数对 (x,y) 的范围中，找出使 P 达到最大的数对 (x,y).

先讨论第一步.

如图 3－3－1(1)，直线 l：$4x+y=10$ 将平面分成上、下两个半平面区域，直线 l 上的点的坐标满足方程 $4x+y=10$，即 $y=10-4x$，直线 l 上方的平面区域中的点的坐标满足不等式 $y>10-4x$，直线 l 下方的平面区域中的点的坐标满足不等式 $y<10-4x$.

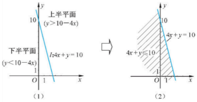


图 3－3－1

因此，$4x+y\leqslant 10$ 在平面上表示的是直线 l 及直线 l 下方的平面区域，即图 3－3－1(2) 中的阴影部分（包括边界直线 l）.

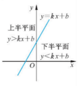


图 3－3－2

一般地，直线 $y=kx+b$ 把平面分成两个区域（图 3－3－2）：

$y>kx+b$ 表示直线上方的平面区域；

$y<kx+b$ 表示直线下方的平面区域.

思 考

对于二元一次不等式 $Ax+By+C>0$ $(A^2+B^2\neq 0)$，如何确定它所表示的平面区域？

例 1 画出下列不等式所表示的平面区域：

(1) $y>-2x+1$； (2) $x-y+2>0$.

解 (1)，(2) 两个不等式所表示的平面区域如图 3－3－3(1)，(2) 所示.

22.1.4 二次函数 $y=ax^2+bx+c$ 的图象和性质

先研究一个具体的二次函数 $y=\frac{1}{2}x^2-6x+21$ 的图象和性质.

思考

我们已经知道二次函数 $y=a(x-h)^2+k$ 的图象和性质，能否利用这些知识来讨论二次函数 $y=\frac{1}{2}x^2-6x+21$ 的图象和性质？

配方可得：

$$
\begin{aligned}
y &= \frac{1}{2}x^2-6x+21 \\
&= \frac{1}{2}(x-6)^2+3.
\end{aligned}
$$

根据前面的知识，我们可以先画出二次函数 $y=\frac{1}{2}x^2$ 的图象，然后把这个图象向右平移 6 个单位长度，再向上平移 3 个单位长度，得到二次函数 $y=\frac{1}{2}x^2-6x+21$ 的图象.

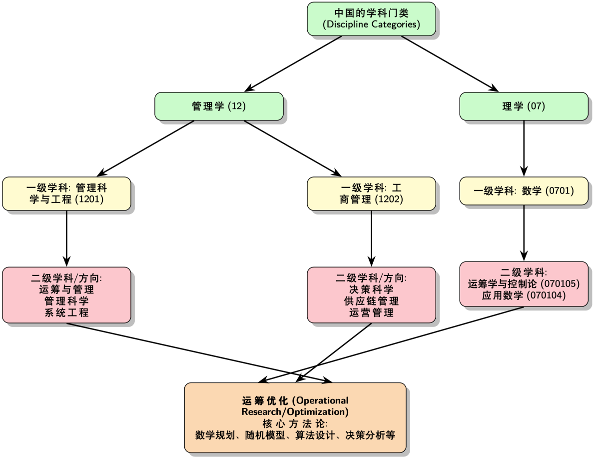
还有其他平移方法吗？

第二十二章　二次函数　37

人民教育出版社

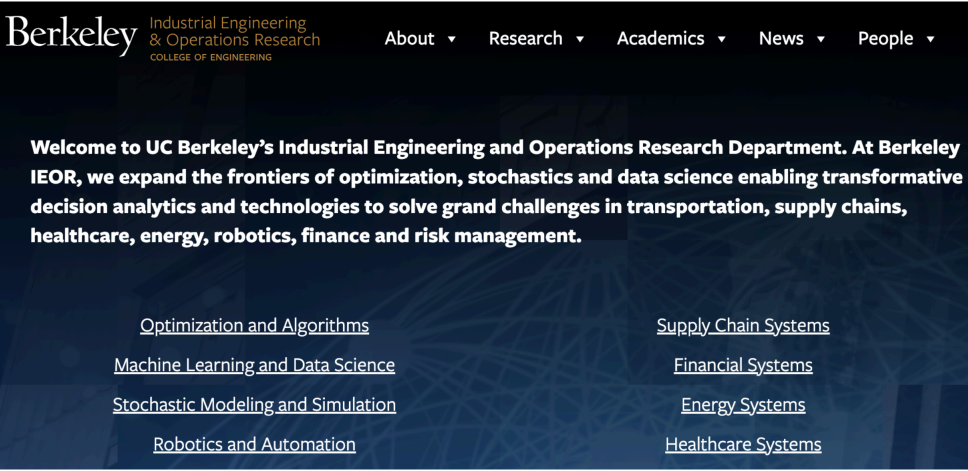
Subject classification

Optimization = Mathematical Programming



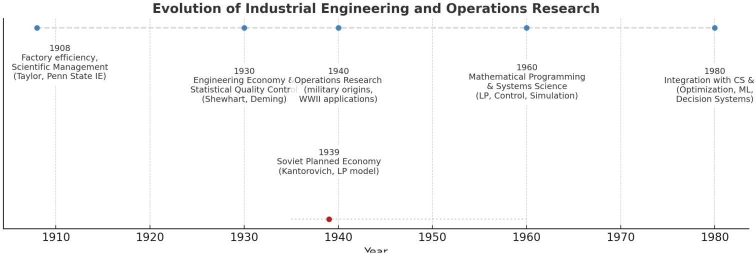
Subject classification (international): Department of Industrial Engineering and Operations Research (IEOR)

IEOR (Industrial Engineering & Operations Research): a discipline focused on the design, optimization, and management of complex systems and processes.



- Origins (Early 20th Century): In 1908, Pennsylvania State University offered the first courses in Industrial Engineering and established a department in 1909. Cornell University awarded the first Ph.D. in the field in 1933.
- Wartime Development (1940s): During World War II, Operations Research (OR) emerged as a discipline focused on tactical efficiency and resource allocation.
- Postwar Expansion (Late 1940s–1950s): After 1949, the RAND Corporation helped expand OR from military to civilian applications.
- Academic Growth (Mid-20th Century): In the 1950s–1960s, linear programming found its first major industrial uses, notably in oil refining at Shell, Exxon, and BP. University departments in Operations Research, Management Science, Industrial Engineering, and Systems Science grew rapidly across Europe and North America.
- Soviet Application (1928–1991): In parallel, the Soviet Union applied mathematical planning methods, most notably Kantorovichs linear programming model, within its centrally planned economy.

Prestigious Industrial Engineering (IE) departments



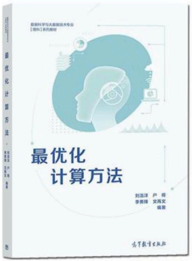
- USA: Georgia Tech (ISyE), UC-Berkeley (IEOR), MIT (ORC), UMich (IOE), Stanford (MS&E), Northwestern (IEMS), Cornell (ORIE), Purdue (IE), UW–Madison (ISE), UT Austin (ORIE)
- Europe: Technical University of Munich, Cambridge, KU Leuven, ETH, TU Delft, Oxford, Manchester...
- Canada: Toronto, McGill

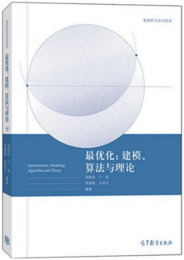
- Formulating Optimization Models: Translate complex real-world challenges from various domains (e.g., scientific, managerial, engineering) into precise mathematical optimization problems.

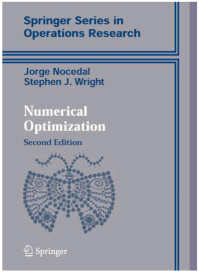
- Implementing Solutions with Modern Solvers: Utilize state-of-the-art optimization software and tools to efficiently solve formulated problems and interpret the results.

- Designing and Analyzing Algorithms: Develop and evaluate fundamental optimization algorithms to gain deeper insights into solution methodologies and address problems where existing solvers are inadequate.

References: nonlinear programming







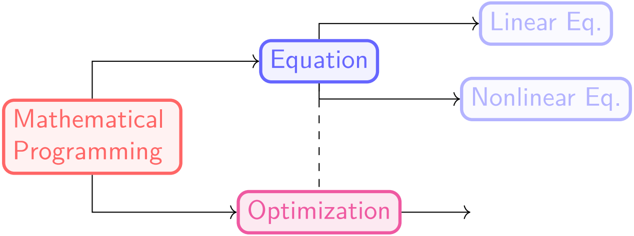
[1] Numerical Optimization, Jorge Nocedal and Stephen Wright, 2nd Edition (2006)

[2] 最优化计算方法，高等教育出版社，文再文 等编著。

其他优秀教材：Course Notes for EE227C (Spring 2018): Convex Optimization and Approximation. https://ee227c.github.io/

Outline

Mathematical Programming, or Optimization deals with solving:



General form of an optimization problem

$$
\begin{aligned}
\underset{x \in \mathbb{R}^n}{\text{minimization}} \quad & f(x) \\
\text{subject to} \quad & c_i(x) = 0, \quad i \in \mathcal{E} \\
& c_i(x) \leq 0, \quad i \in \mathcal{I} \\
& x \in \Omega
\end{aligned}
$$

Three elements:

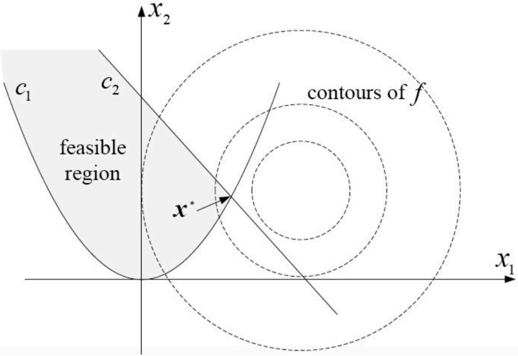
- Objective: a performance measure of a system to minimize (min), maximize (max), sup, inf.
- Variables: features that the users can manipulate
- Constraints: conditions must be satisfied by variables; subject to (s.t.) constraints.

$$\begin{array}{ll} \text{minimize} & (x_1-2)^2+(x_2-1)^2 \\ \text{subject to} & x_1^2-x_2\le 0 \\ & x_1+x_2\le 2. \end{array}$$

Solution by Graphic

- Draw the feasible set
- Draw the contours
- Determine the minimizer

feasible region/feasible set
feasible solution/point
contour
optimal solution
minimizer
solution
minima



Feasible region: closed set.

$$
\begin{aligned}
\underset{x\in\mathbb{R}^n}{\text{minimization}} \quad & f(x) \\
\text{subject to} \quad & c_i(x) = 0, \quad i \in \mathcal{E} \\
& c_i(x) \le 0, \quad i \in \mathcal{I} \\
& x \in \Omega
\end{aligned}
$$

- Linear Optimization v.s. Nonlinear Optimization
- Constrained Optimization v.s. Unconstrained Optimization
- Continuous Optimization v.s. Integer Optimization

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimization}} \quad & f(x) \\ \text{subject to} \quad & c_i(x) = 0, \quad i \in \mathcal{E} \\ & c_i(x) \le 0, \quad i \in \mathcal{I} \\ & x \in \Omega \end{aligned}$$

- Stochastic: time-consuming or even impossible to evaluate the function or the gradient
- Random approximation: $g(x; \xi) \approx \nabla f(x)$, ξ represents a random sampling
- Very popular in big data scenarios: $f(x) = \sum_{i=1}^{N} f_i(x) := f(x; a_i)$

Single-objective Optimization: there are multiple objective functions to minimize

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

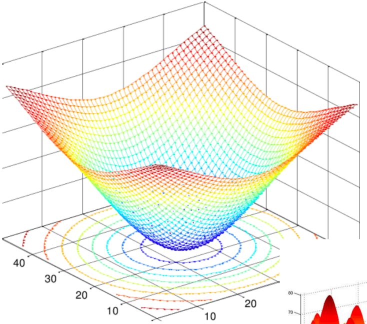
Multi-objective Optimization:

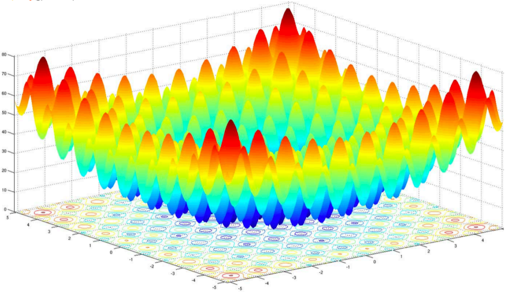
$$\min_{x \in X} \{f_1(x), f_2(x), \ldots, f_N(x)\}$$

$$\begin{aligned} \min_{x\in X, y\in Y} \quad & F(x,y) \\ \text{subject to:} \quad & G_i(x,y) \leq 0, i = 1, ..., I \\ & y \in \arg\min_{z\in Y}\{f(x,z) : g_j(x,y), j = 1, ..., J\} \end{aligned}$$

where:

- $F(x, y(x))$ is the objective function of the upper level problem, which depends on the solution $y(x)$ of the lower level problem.
- $g(x, y)$ is the objective function of the lower level problem.


40
30
20
10
10
20



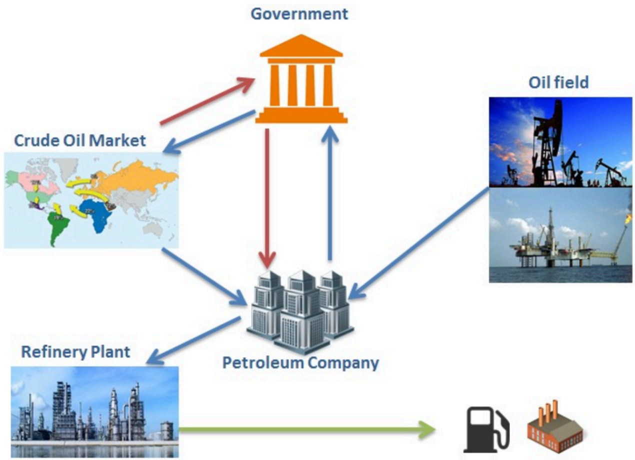
System of equations:

- Linear equations
- Nonlinear equations

Optimization problems:

- Linear Programming
- Nonlinear Programming
- Integer Programming
- Stochastic Programming
- ......

- World War II and Operations Research (1940s): optimizing military logistics, resource allocation, and scheduling. Linear programming (LP) was pioneered by George Dantzig in 1947 to improve the allocation of resources such as fuel, labor, and materials to maximize efficiency and reduce costs.
- Post-War Industry Applications: LP started to find its way into manufacturing, transportation, and resource management. Companies used LP to optimize production schedules, minimize waste, and improve supply chain logistics.
- Refinery Operations and Process Optimization (1950s-1960s): optimization started to be applied to refining processes. Optimization models helped refineries manage complex decisions about blending crude oils and intermediates to produce various fuel products while adhering to quality constraints.

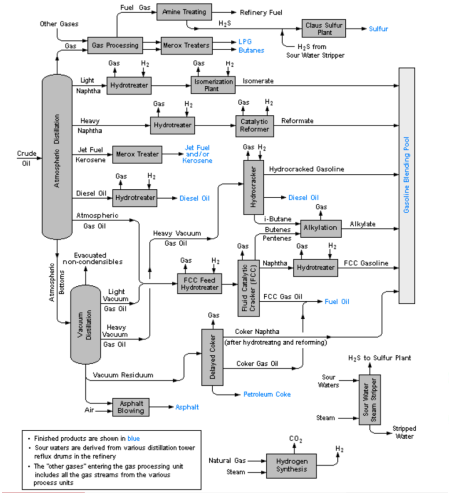


Government
Oil field
Crude Oil Market
Refinery Plant
Petroleum Company

Crude oil:
Type 1
Type 2
Type 3
....

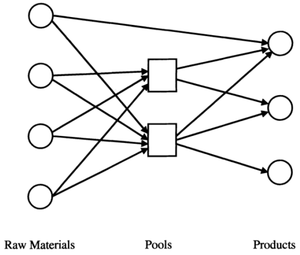
Products:
Motor fuel (87, 89, 91)
Diesel oil
Liquefied petroleum gases (LPG)
Jet aircraft fuel
Kerosene
Heating fuel oils
Lubricating oils
Asphalt
Petroleum coke
....

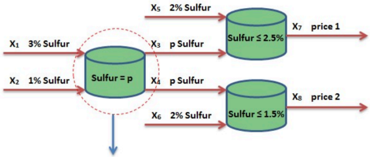
Refinery Flowchart



- Finished products are shown in blue
- Sour waters are derived from various distillation tower reflux drums in the refinery
- The "other gases" entering the gas processing unit includes all the gas streams from the various process units

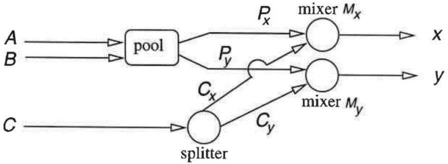
Pooling model





$p X_3 + p X_4 = 3\% X_1 + 1\% X_2$ **Bilinearity: nonlinear! nonconvex!**

Example: Haverly Pooling Problem



A, B, C : quantities of crude oil A, B, C, respectively
x, y : quantities of products x and y
p : sulfur content of outstream of the pool
P_x, P_y : outstreams of the pool
C_x, C_y : outstreams of the splitter

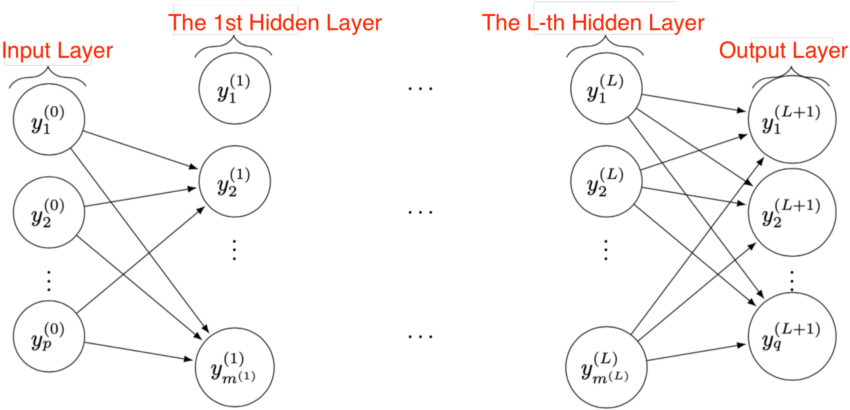
$$\max_{x,y,A,B,p,C_x,C_y,P_x,P_y} \quad 9x + 15y - 6A - 8B - 10(C_x + C_y)$$

s.t.

$$\begin{aligned}
P_x + P_y - A - B &= 0 \\
x - P_x - C_x &= 0 \\
y - P_y - C_y &= 0 \\
pP_x + 2C_x - 2.5x &\le 0 \\
pP_y + 2C_y - 1.5y &\le 0 \\
pP_x + pP_y - 3A - B &= 0 \\
0 \le x \le 200, 0 \le y \le 200, 0 \le p &\le 100 \\
0 \le A, B, C_x, C_y, P_x, P_y &\le 500
\end{aligned}$$

sulfur content of A=3
sulfur content of B=1
sulfur content of C=2
sulfur content of x ≤ 2.5
sulfur content of y ≤ 1.5
Demands of x and y are ≤ 200
Supplies of A, B, C_x, C_y, P_x, P_y ≤ 500

Example: DNN



$$\min_x \quad \sum_{i=1}^{m} \|h(a_i; x) - b_i\|_2^2 + \lambda r(x)$$

Outline

Iterative algorithm: given x^0, generate sequence $\{x_k\}$ by

$$x^{k+1} \leftarrow \mathscr{A}(x_k)$$

- $\mathscr{A}$ is constructed by using function values, objectives, Hessian, etc.
- In theory, terminate finitely or generate an infinite sequence $\{x_k\}$ to approximate the real solution.
- In practice, terminate when a tolerance is reached.
- Generally, the sequence converges to an "optimal solution".

Recall the following definition.

Convergent sequence

A sequence $\{x_k\}$ is said to converge to x^* if

$$\lim_{k\to\infty} \|x_k - x^*\|_2 = 0.$$

The following theorem is also extremely useful when analyzing algorithms.

Bolzano-Weierstrauss Theorem

Every bounded sequence in $\mathbb{R}^n$ has a convergent subsequence.

Globally convergent algorithm

If, from any initial iterate x^0 (and under presumed conditions), the iterate sequence $\{x_k\}$ generated by an algorithm for solving a problem converges to a solution of the problem, then the algorithm is said to be globally convergent (under the presumed conditions).

- Note that, in the context of optimization, this definition does not presume require convergence to a globally optimal solution. (Even optimization experts are often confused by this!)
- This definition is not saying that a given iterate sequence is "globally" convergent. After all, if such an algorithm produces a sequence $\{x_k\}$, the sequence itself is simply convergent.
- Rather, this definition refers to a property of an algorithm and the fact that it produces a convergent sequence from any starting point.
- We may also say that an algorithm is globally convergent if, e.g., a stationarity measure vanishes, even if the iterate sequence itself does not converge; e.g., if $\nabla f(x_k) \to 0$.

Locally convergent algorithm

If, from any initial iterate x^0 in a neighborhood of a solution x^* (and under presumed conditions), the iterate sequence $\{x_k\}$ generated by an algorithm converges to x^*, then the algorithm is said to be locally convergent to x^* (under the presumed conditions).

- A locally convergent algorithm is not necessarily globally convergent.
- A globally convergent algorithm is locally convergent.
- However, an important characterization of local convergence is the corresponding rate of convergence. This distinguishes "how quickly" the iterates converge; e.g., 0.99999999999_k converges to 0, but I don't what to have to wait for it to get very close to 0...

Suppose $\|x_k - x^*\|_2 \to 0$; i.e., the sequence $\{x_k\}$ converges to x^*.

Q-linear convergence

If there exists a constant $c \in [0,1)$ and $\hat{k} \geq 0$ such that

$$\|x^{k+1} - x^*\|_2 \leq c\|x_k - x^*\|_2 \text{ for all } k \geq \hat{k},$$

then $\{x_k\}$ converges Q-linearly to x^*.

Q-superlinear convergence

If there exists a sequence $\{c_k\} \to 1$ such that

$$\|x^{k+1} - x^*\|_2 \leq c_k\|x_k - x^*\|_2,$$

then $\{x_k\}$ converges Q-sublinearly to x^*.

Suppose $\|x_k - x^*\|_2 \to 0$; i.e., the sequence $\{x_k\}$ converges to x^*.

Q-superlinear convergence

If there exists a sequence $\{c_k\} \to 0$ such that

$$\|x^{k+1} - x^*\|_2 \leq c_k \|x_k - x^*\|_2,$$

then $\{x_k\}$ converges Q-superlinearly to x^*.

Q-quadratic convergence

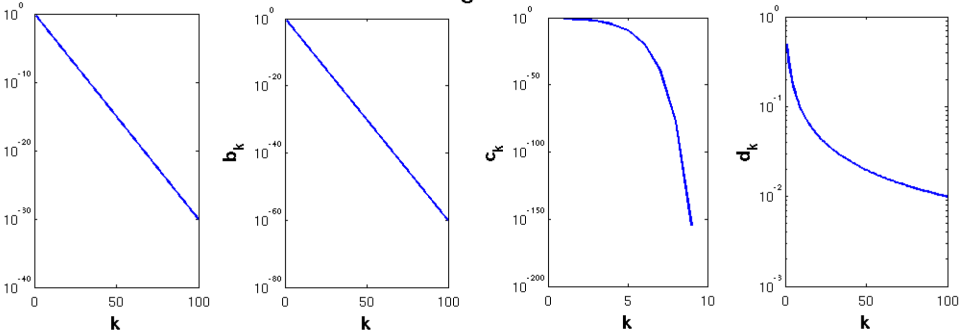
If there exists a constant $c \geq 0$ and $\hat{k} \geq 0$ such that

$$\|x^{k+1} - x^*\|_2 \leq c\|x_k - x^*\|_2^2 \text{ for all } k \geq \hat{k},$$

then $\{x_k\}$ converges Q-superlinearly to x^*.

There is also "R-" convergence, but we'll skip that; so we'll drop the "Q-".

Convergence Plot



The constants c (and $\{c_k\}$) can make a difference, but generally, linear convergence is "slow" and superlinear/quadratic convergence is eventually very "fast".

- $\{1 + 2^{-k}\} = \{2, \frac{3}{2}, \frac{5}{4}, \frac{9}{8}, ...\}$ converges linearly to $x^* = 1$.
  Proof:

- $\{1 + 2^{-2^k}\} = \{\frac{3}{2}, \frac{5}{4}, \frac{17}{16}, \frac{257}{256}, ...\}$ converges quadratically to $x^* = 1$.
  Proof:

Table: Examples of Different Convergence Rates

Linear	Sublinear	Superlinear	Quadratic
1.000000e-01	1.000000e+00	3.162278e-02	1.000000e-02
1.000000e-02	5.000000e-01	1.000000e-03	1.000000e-04
1.000000e-03	3.333333e-01	1.778280e-05	1.000000e-08
1.000000e-04	2.500000e-01	1.000000e-07	1.000000e-16
1.000000e-05	2.000000e-01	3.162278e-10	1.000000e-32
1.000000e-06	1.666667e-01	5.623413e-13	1.000000e-64
1.000000e-07	1.428571e-01	5.623413e-16	1.000000e-128
1.000000e-08	1.250000e-01	3.162278e-19	1.000000e-256
1.000000e-09	1.111111e-01	1.000000e-22	0.000000e+00
1.000000e-10	1.000000e-01	1.778280e-26	0.000000e+00

Typically, this measures the decay of some optimality criterion, such as

$$f(x_k) - f^* \leq \frac{C}{k}.$$

Examples:

- Gradient Descent on convex smooth functions: $f(x_k) - f^* = O(1/k)$.
- Nesterov's Accelerated Gradient: $f(x_k) - f^* = O(1/k^2)$.
- Subgradient method on convex Lipschitz functions: $f(x_k) - f^* = O(1/\sqrt{k})$.

In this context:

- $O(1/k)$: sublinear convergence rate,
- $O(1/k^2)$: accelerated sublinear rate,
- $O(\rho^k), \ \rho < 1$: linear (geometric) rate.

Alternative description using ϵ-optimality.

- Programming language: Matlab, Python
- Optimization toolbox: Excel, Matlab, Python, Cplex, Lingo, Lindo
- State-of-the-art solver: NEOs optimization solvers
  https://neos-server.org/neos/solvers/index.html
- Modeling language: AMPL, GAMS, AIMMS, Lingo, Lindo

NEOs solver

https://neos-server.org/neos/solvers/index.html

NEOS Contact Help

Listed below are the available solvers organized by Problem Type. An addit

If you need help in selecting a solver, consult the Optimization Tree of the N
problem.

Each solver has sample problems and background information on the solve
you encounter problems, consult the NEOS Server FAQ, or contact us by c

Problem Type | Solver

- Job Queue Tools
- Application
- Bound Constrained Optimization
- Combinatorial Optimization and Integer Programming
- Complementarity Problems
- Extended Mathematical Programming
- Global Optimization
- Linear Network Programming
- Linear Programming
- Mathematical Programs with Equilibrium Constraints
- Mixed Integer Linear Programming
- Mixed Integer Nonlinearly Constrained Optimization
- Mixed-Integer Optimal Control Problems

Problem Type | Solver

Job Queue Tools

- View Job Queue
- View Job Results / Kill a Job

Application

- CONVERT [GAMS Input]
- Domino [jpeg Input]
- ECM [csv Input][single_text Input][zip Input]
- Fishwerks [csv Input]

Bound Constrained Optimization

- L-BFGS-B [AMPL Input]

Combinatorial Optimization and Integer Programming

- BiqMac [SPARSE Input]
- concorde [TSP Input]

Complementarity Problems

- Knitro [AMPL Input]
- MILES [GAMS Input]
- NLPEC [GAMS Input]
- PATH [AMPL Input][GAMS Input]

Extended Mathematical Programming

- DE [GAMS Input]
- JAMS [GAMS Input]

Global Optimization

- ANTIGONE [GAMS Input]
- ASA [AMPL Input]
- BARON [AMPL Input][GAMS Input]
- Couenne [AMPL Input][GAMS Input]
- icos [AMPL Input]
- LGO [AMPL Input]
- LINDOGlobal [GAMS Input]
- PGAPack [AMPL Input]
- PSwarm [AMPL Input]
- RAPOSa [AMPL Input]
- scip [AMPL Input][CPLEX Input][GAMS Input][MPS Input][OSIL Input][Python Input][ZIMPL Input]