"商务视角下的数据分析"课程所覆盖的专题

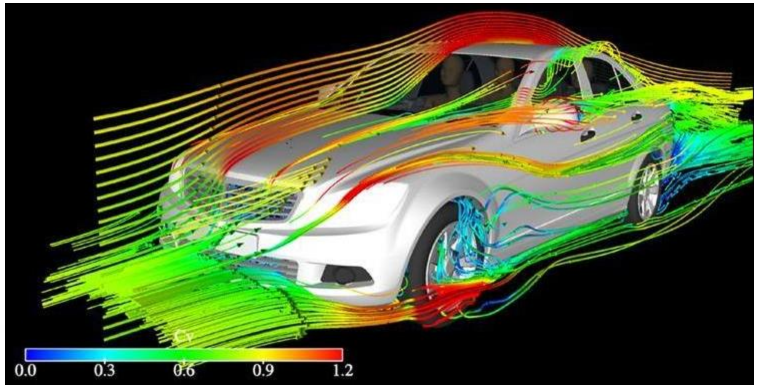
1. **简介**
2. **商务思维 (Business thinking)**
   - 所谓的"商务 (BUSINESS)" – 其实就是学会做出获得更多利润的决策 (making decisions to earn more profit)
   - 管理技巧 (Management skills) – 如何落实那些决策
   - 试试创业？ – 可以！但是要慎重！！
3. **数据分析的方法概览 (Data Analytics methods)**
   - 其实，数据分析有着悠久的历史 (HISTORY view about Data Analytics)
   - 理解数据分析方法的 – 一点优化的技巧 (OPTIMIZATION)
   - 来自统计学的数据分析方法 (STATISTICS) – 基于抽样的推断 (一个有趣的视角来梳理而已，不重复)
   - 来自机器学习的数据分析方法 (BASIC + ADVANCED) – 基于数据的知识发现 (KDD)
4. **实用技巧 (Practical skills)**
   - 大商务，需要大数据
   - 大商务的两个挑战: "秒杀" 和 "精准广告/推荐"
5. **课程总结**

一点优化的技巧 (OPTIMIZATION)

- **还是喜欢从历史入手 –**
  - Optimization? 最优化?
  - A brief history
  - **Calculus** ([partial] derivative) + **Linear Algebra** – modern tools for optimization
    - Calculus of variations [变分法]
    - Operational Research [运筹学]
- **优化问题概览 及其解决方案**
  - LP, NLP (QP,SOCP,SDP, CP, PP)
  - Solutions: Descent, Newton, …

优化 (Optimization)无处不在

- □ **We always intend to maximize or minimize something**
  - from engineering design to financial markets
    - ➢Design the shape of a car with minimum aerodynamic drag [空气阻力]



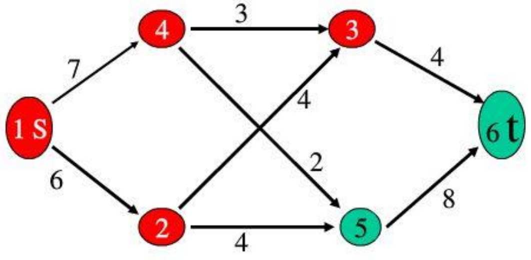
  - from our daily activity to planning our holidays

青藏线
G109
北
青海湖
西宁
格尔木
都兰县
茶卡镇
茶卡盐湖
344
296
公里数(Km)
塔尔寺
昆仑山口
可可西里自然保护区
270
五道梁
玉珠峰
6178
风火山口
147
沱沱河（唐古拉镇）
唐古拉山口
279
藏北草原
头二九山
安多县
137
申格里贡山
那曲县
纳木措
念青唐古拉峰
7162
166
当雄县

  - and computer sciences to industrial applications
    - ➢the maximal network flow
      Shortest path/Critical path
  - …

实例：

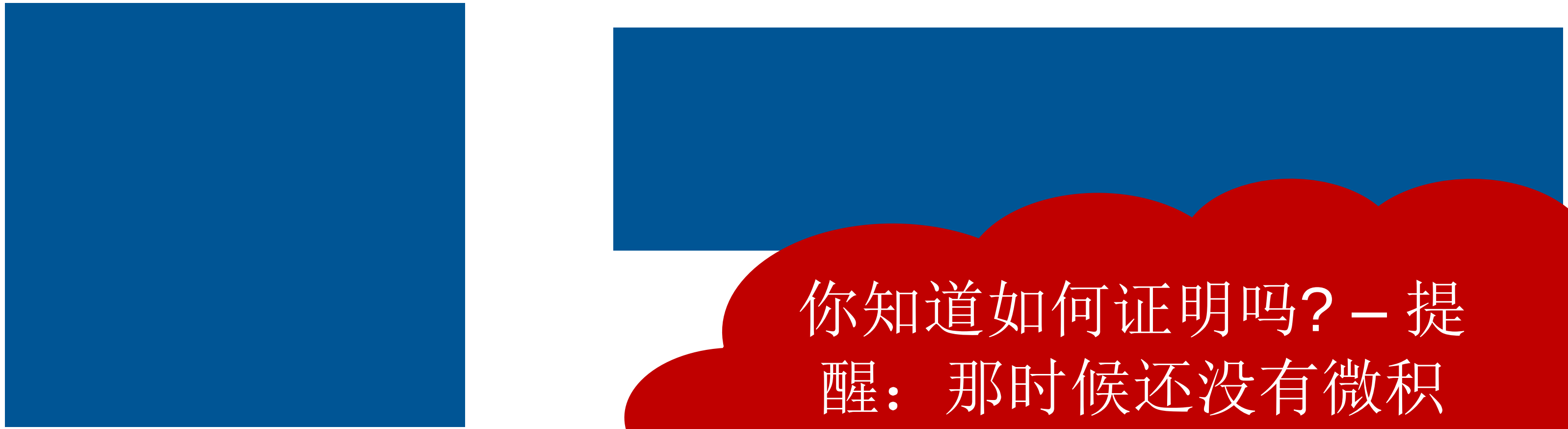
有一自来水管道输送系统，起点是S，目标是T，途中经过的管道都有一个最大的容量。



- 问题：问从S到T的最大水流量是多少？

其实，优化问题很早就有了

- **300 BC**
  - Euclid proved that
    - a square has the greatest area among the rectangles with given total length of the edges
    - 边长固定，在长方形中，正方形（正方形是长方形的特例）面积最大



你知道如何证明吗? – 提醒：那时候还没有微积分 (calculus) 和 优化论 (Optimization)哟!

200 BC, Zenodorus Dido's problem
Greatest area under a curve

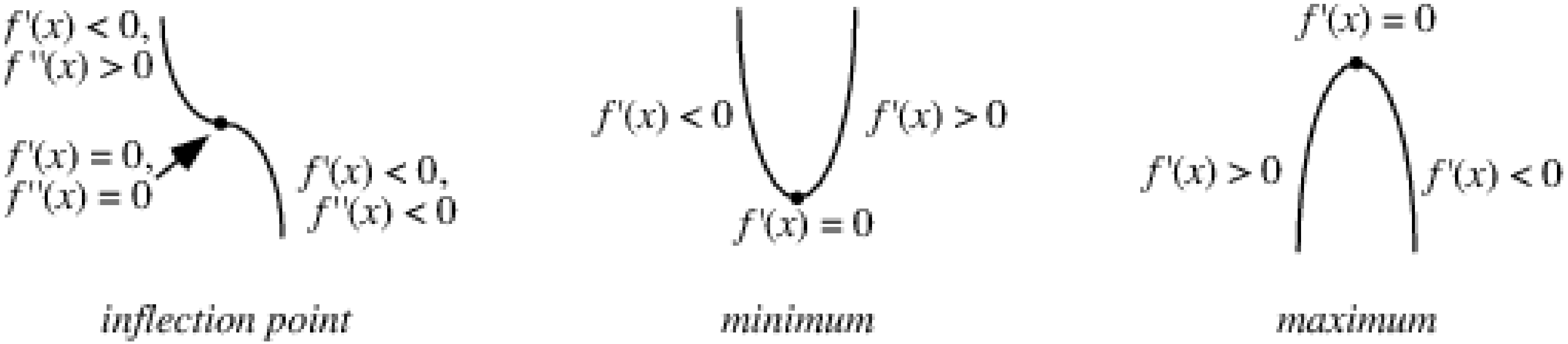
Dido Purchases Land for the Foundation of Carthage. Engraving by Matthäus Merian the Elder, in *Historische Chronica*, Frankfurt a.M., 1630. Dido's people cut the hide of an ox into thin strips and try to enclose a maximal domain.

17th – 18th century
CoV: Calculus of Variations [变分法]

- Before the invention of calculus of variations only some odd [零散的] optimization problems are being investigated.

- **With calculus, the Stationary poin**
satisfies $f'(x) = 0$ with second deri
  - (Local) Minimum: $f''(x) > 0$
  - (Local) Maximum: $f''(x) < 0$
  - Point of inflexion $f''(x) = 0$

These concepts are further extended to N-Dim vectors and matrices



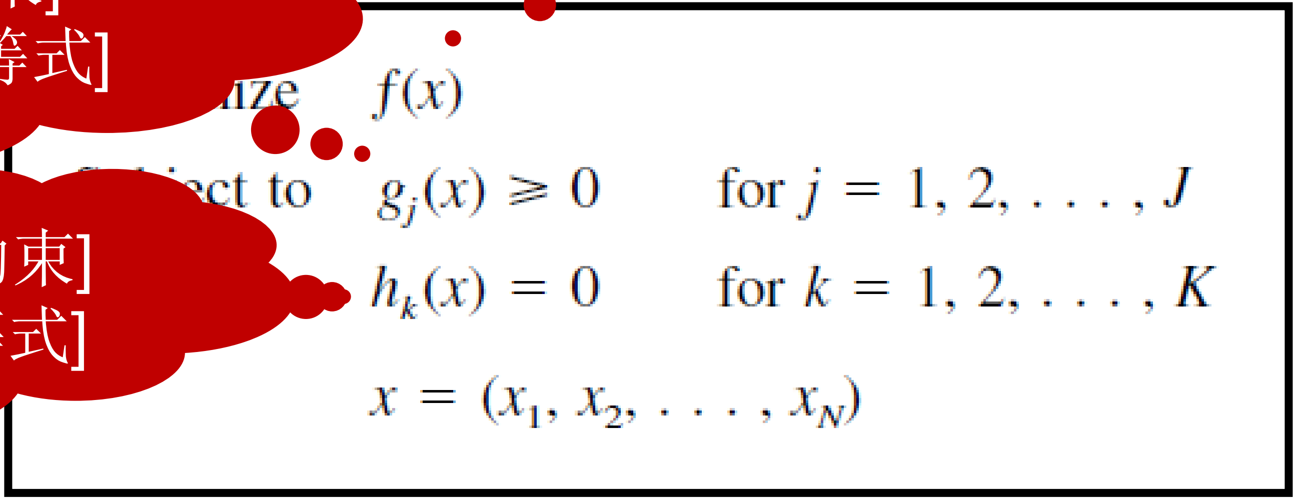
https://en.wikipedia.org/wiki/Stationary_point

General form of Optimization problems

Objective Function
[目标函数]

Constraint [约束] –
Inequality [不等式]

Constraint [约束]
– Equality [等式]



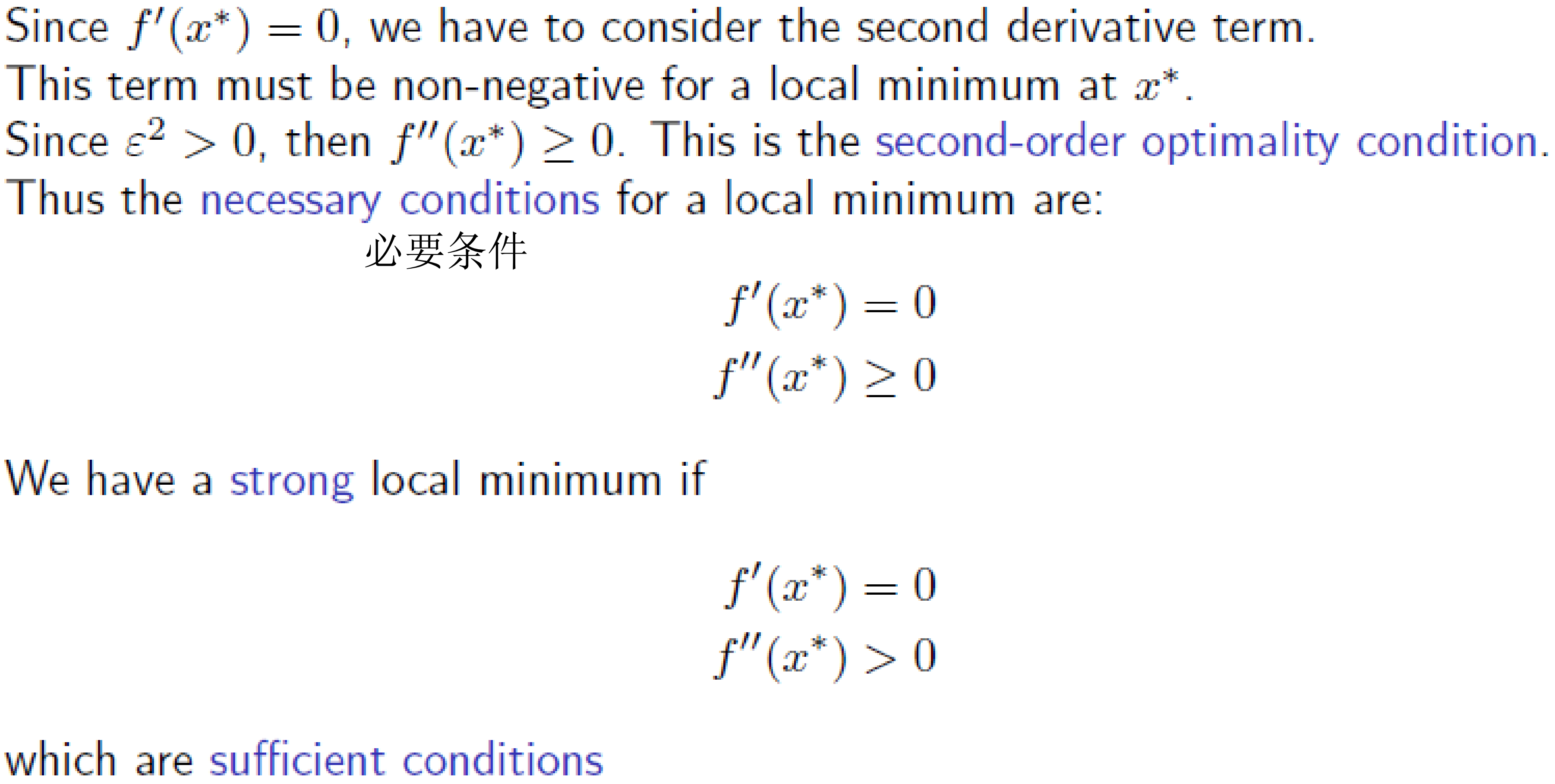
ize $f(x)$

ect to $g_j(x) \geqslant 0 \qquad \text{for } j = 1, 2, \ldots, J$

$h_k(x) = 0 \qquad \text{for } k = 1, 2, \ldots, K$

$x = (x_1, x_2, \ldots, x_N)$

Optimality Conditions 2



Since $f'(x^*) = 0$, we have to consider the second derivative term.
This term must be non-negative for a local minimum at x^*.
Since $\varepsilon^2 > 0$, then $f''(x^*) \geq 0$. This is the second-order optimality condition.
Thus the necessary conditions for a local minimum are:

必要条件

$$f'(x^*) = 0$$
$$f''(x^*) \geq 0$$

We have a strong local minimum if

$$f'(x^*) = 0$$
$$f''(x^*) > 0$$

which are sufficient conditions

充分条件

Example A: unconstrained OP

You all may remember “极值定律：Extreme value theorem”

$$f(x) = 5x^6 - 36x^5 + \frac{165}{2}x^4 - 60x^3 + 36$$

$$\frac{df}{dx} = 30x^5 - 180x^4 + 330x^3 - 180x^2 = 30x^2(x-1)(x-2)(x-3)$$

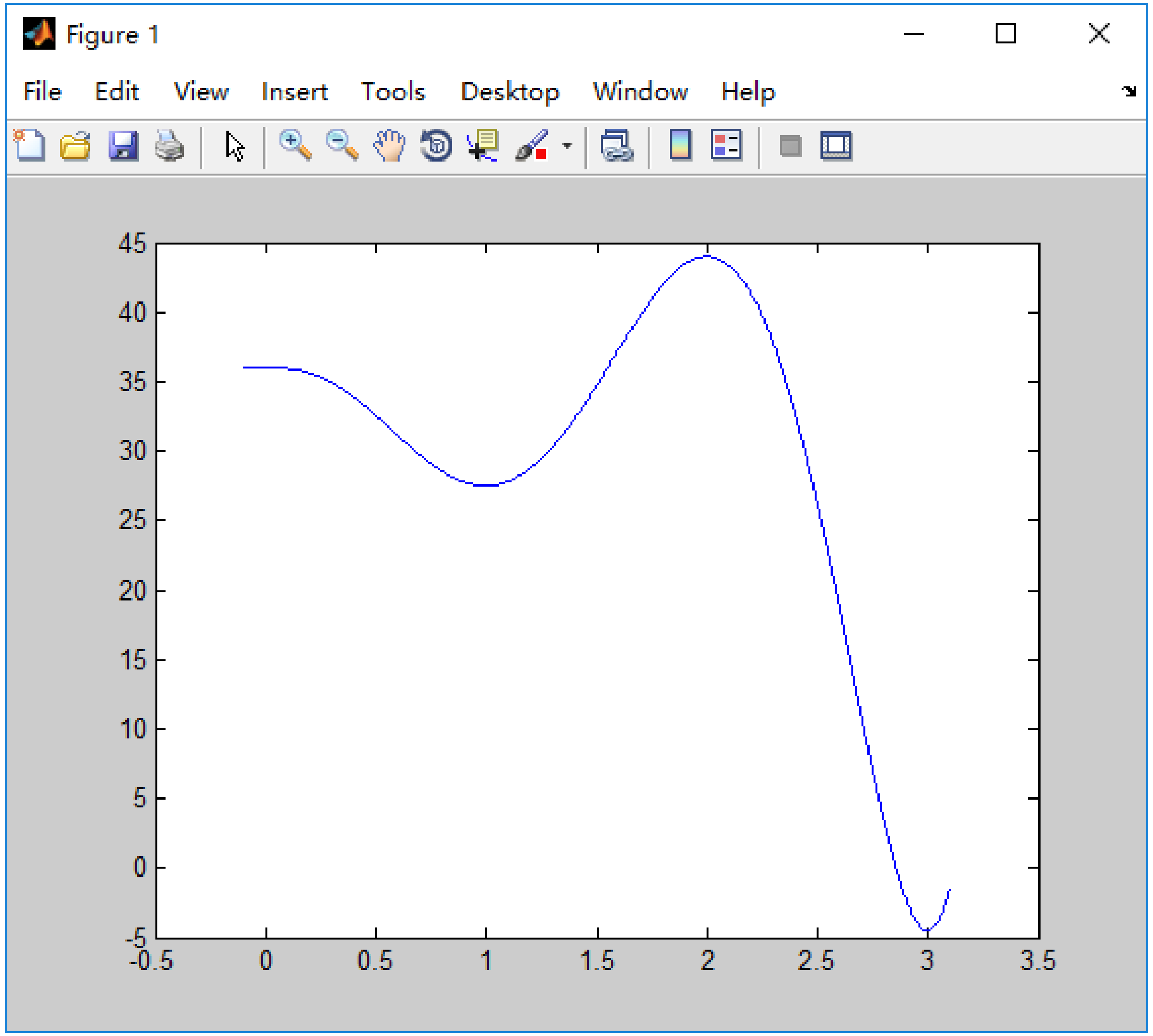
Stationary points $x = 0, 1, 2, 3$

$$\frac{d^2f}{dx^2} = 150x^4 - 720x^3 + 990x^2 - 360x$$

x	$f(x)$	d^2f/dx^2	
0	36	0	
1	27.5	60	-Local minimum
2	44	-120	-Local maximum
3	5.5	540	-Local minimum

At $x = 0$ $\frac{d^3f}{dx^3} = 600x^3 - 2160x^2 + 1980x - 360 = -360$ - Inflection point [拐点]

- **2017年8月18日23:19:13**
- **想画出来看看**
- **Matlab**
  - >> x=[-0.1:0.0000001:3.1];
  - >> fx=5*x.^6 – 36*x.^5 + 165/2*x.^4 - 60*x.^3 + 36;
  - >> plot(x,fx)
- **是对的**
- **不过，一开始尺度不对，画不出来 ☺**



Example **B-1**

- $\boldsymbol{f(x) = 2x - x^2}$
  - First derivative?
  - Second derivative?
  - Stationary point – Max, Min, Inflection point [拐点]?

brachistochrone [brəˈkistəˌkrəun]
n. 最捷落径, 最速降线, 捷线； 反射波垂直时距表

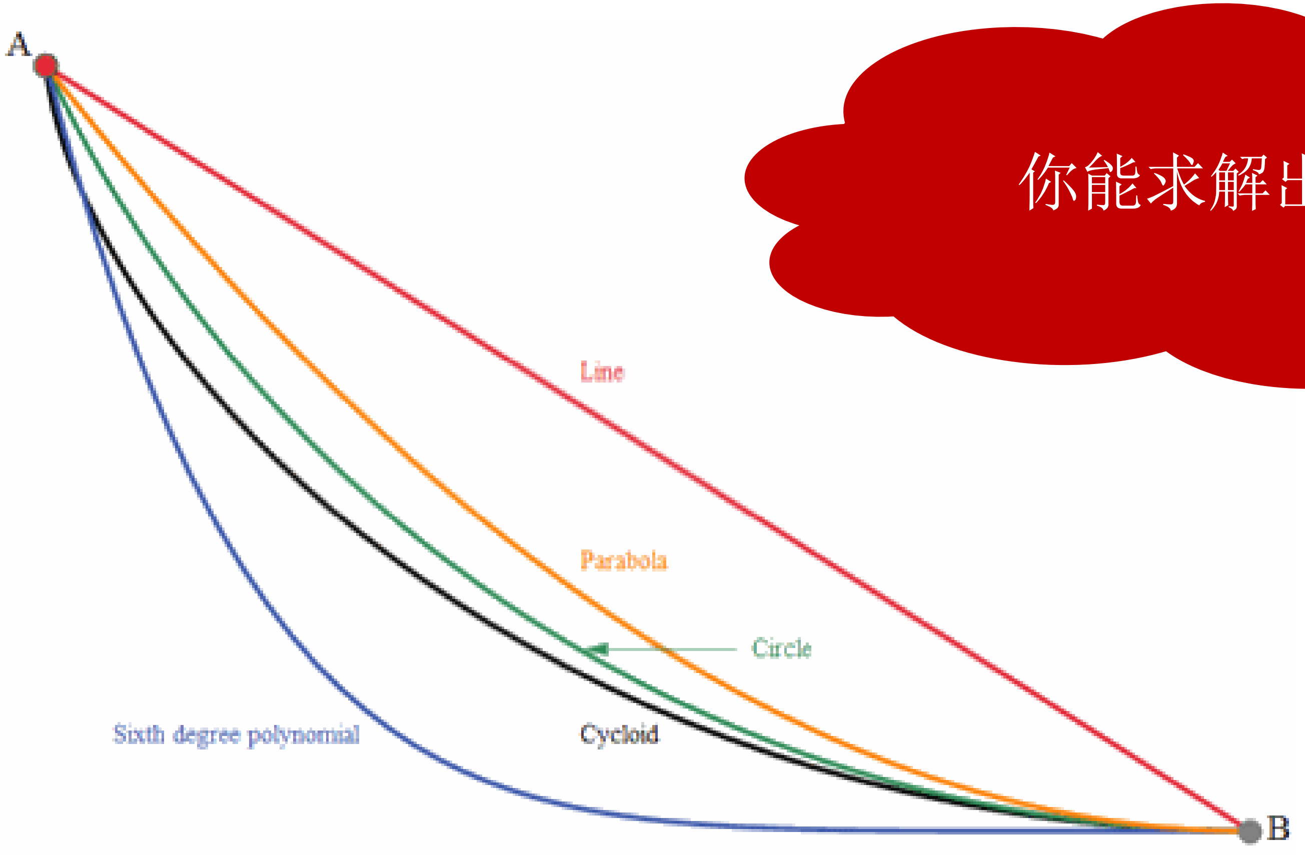
- **Skills for stationary point could be extended to multivariable functions with the help of LA (Linear Algebra)**
  - **Matrix Calculus [矩阵分析]**
- **CoV: Calculus of variations [变分法] – to find optimal function**
  - Issac Newton (1660s) and G. W. von Leibniz (1670s) create mathematical analysis that forms the basis of calculus of variations (**CoV**).

[brəˈkɪstəˌkrəʊn]

  - **Brachistochrone Problem** [最速降線問題]
    - 1696 Johann and Jacob Bernoulli studied Brachistochrone's problem, calculus of variations is born
    - Find the shape of the curve down which a bead sliding from rest and accelerated by gravity will slip (without friction) from one point to another in the least time.

http://mathworld.wolfram.com/BrachistochroneProblem.html https://zh.wikipedia.org/wiki/最速降線問題

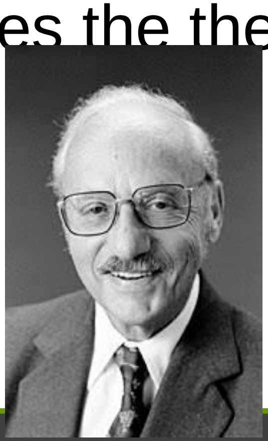
- **Many CoV problems are from physics – modeling [建模] capability is quite powerful!**



你能求解出来吗?

Early 19th century – Operational Research [运筹学]

- **After the world war II optimization develops simultaneously with Operations Research (OR).**
  - J. Von Neumann is an important person behind the development of operations research.
- **The field of algorithmic research expands as electronic calculation (Computers) develops.**
  - 1947 **George B. Dantzig**, who works for US air-forces, presents the **Simplex** method [**单纯形**] for solving LP-problems, **John von Neumann** establishes the theory of **duality** [**对偶**] for LP-problems

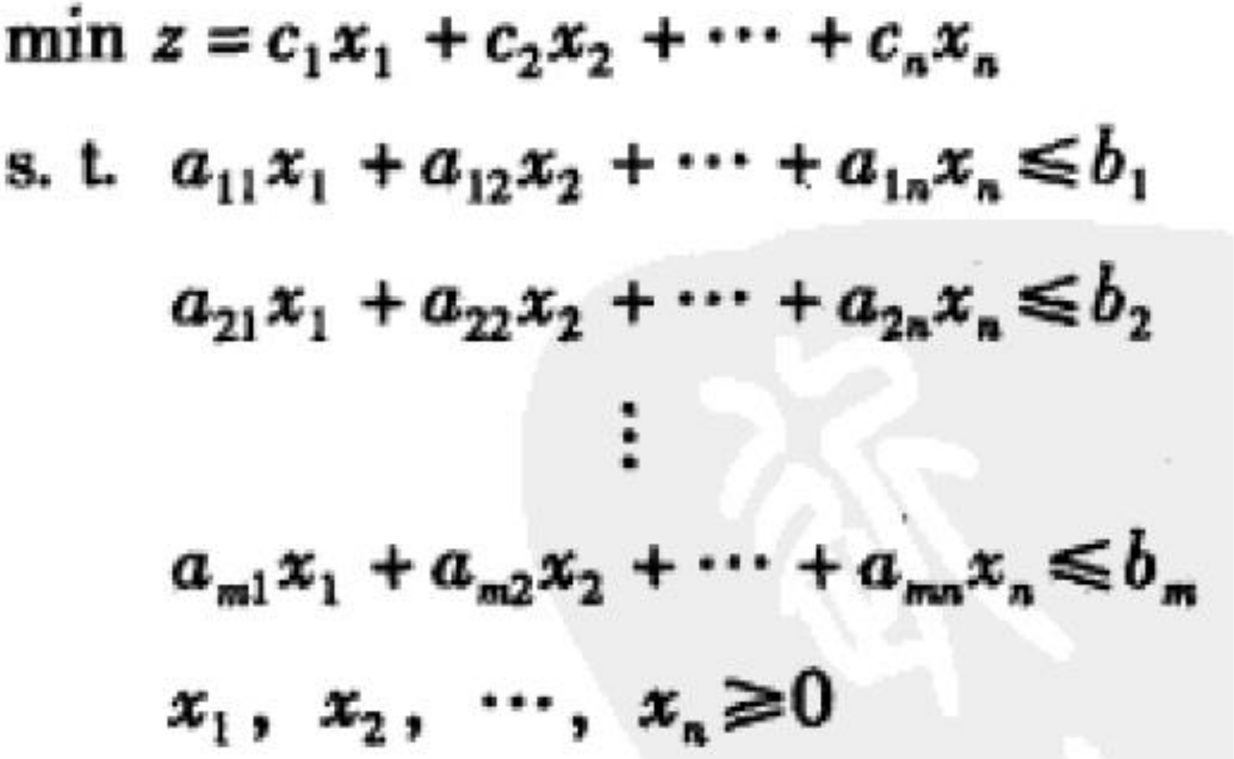


NLP? – At least one of the objective and constrained functions is not linear

The previous extreme-value theorem based method could not be used for LP. Why?

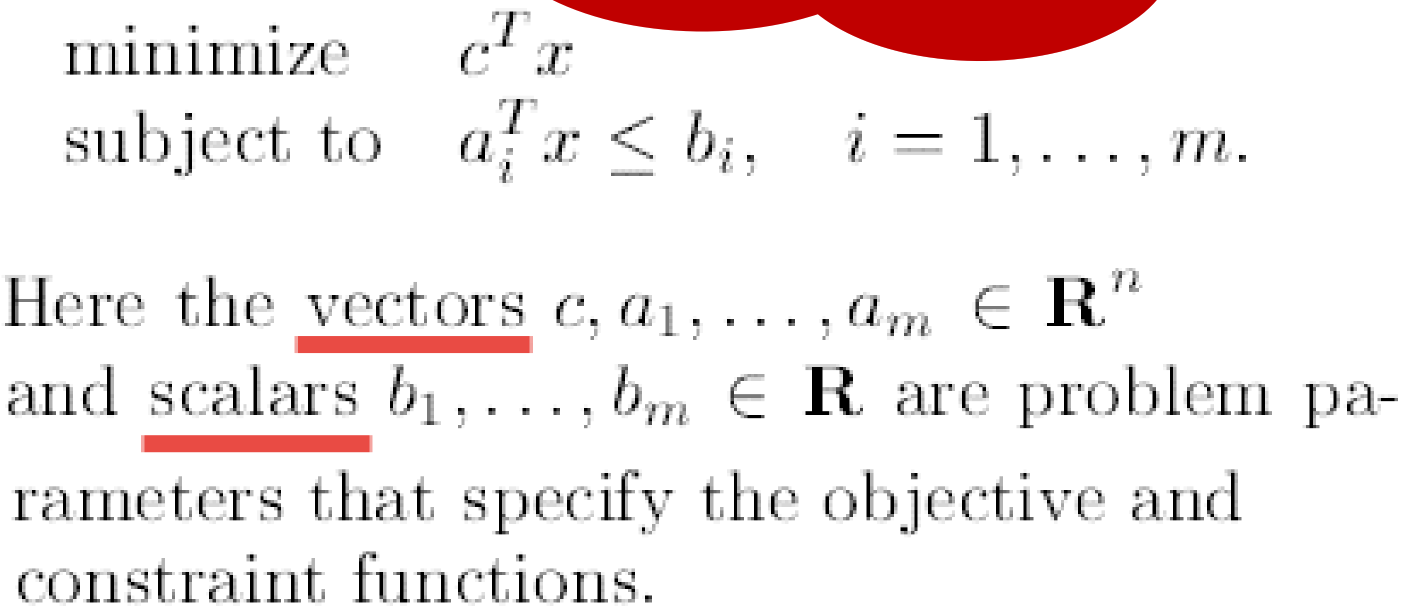
ing [线性

bjective [目标] and all co s are linear



$$
\begin{aligned}
\min\ z &= c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{s. t.}\ & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leqslant b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leqslant b_2 \\
& \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leqslant b_m \\
& x_1,\ x_2,\ \cdots,\ x_n \geqslant 0
\end{aligned}
$$

Here the math is based on Vectors, later simplified into Matrix

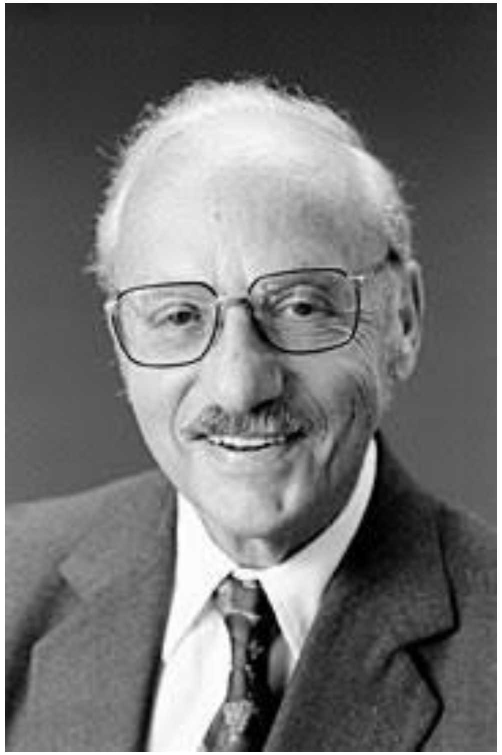


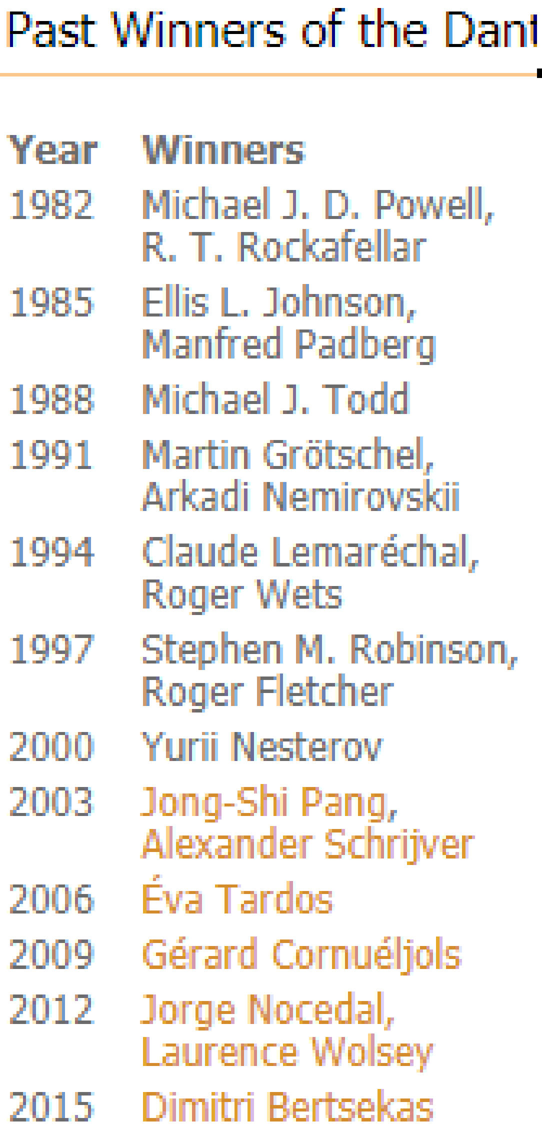
$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \ldots, m. \end{array}$$

Here the vectors $c, a_1, \ldots, a_m \in \mathbf{R}^n$ and scalars $b_1, \ldots, b_m \in \mathbf{R}$ are problem parameters that specify the objective and constraint functions.

Anecdote

- **Top Prizes w.r.t Optimization**
  - The **George B. Dantzig Prize** (pure optimization: Mathematical Optimization Society)
    - The Dantzig Prize was founded by a group of George B. Dantzig's <u>former students </u>(R. W. Cottle, E. L. Johnson, R. M. van Slyke, and R. J.-B. Wets) and was first awarded in 1982.





Past Winners of the Dant

Year	Winners
1982	Michael J. D. Powell, R. T. Rockafellar
1985	Ellis L. Johnson, Manfred Padberg
1988	Michael J. Todd
1991	Martin Grötschel, Arkadi Nemirovskii
1994	Claude Lemaréchal, Roger Wets
1997	Stephen M. Robinson, Roger Fletcher
2000	Yurii Nesterov
2003	Jong-Shi Pang, Alexander Schrijver
2006	Éva Tardos
2009	Gérard Cornuéljols
2012	Jorge Nocedal, Laurence Wolsey
2015	Dimitri Bertsekas

http://www.mathopt.org/?nav=dantzig

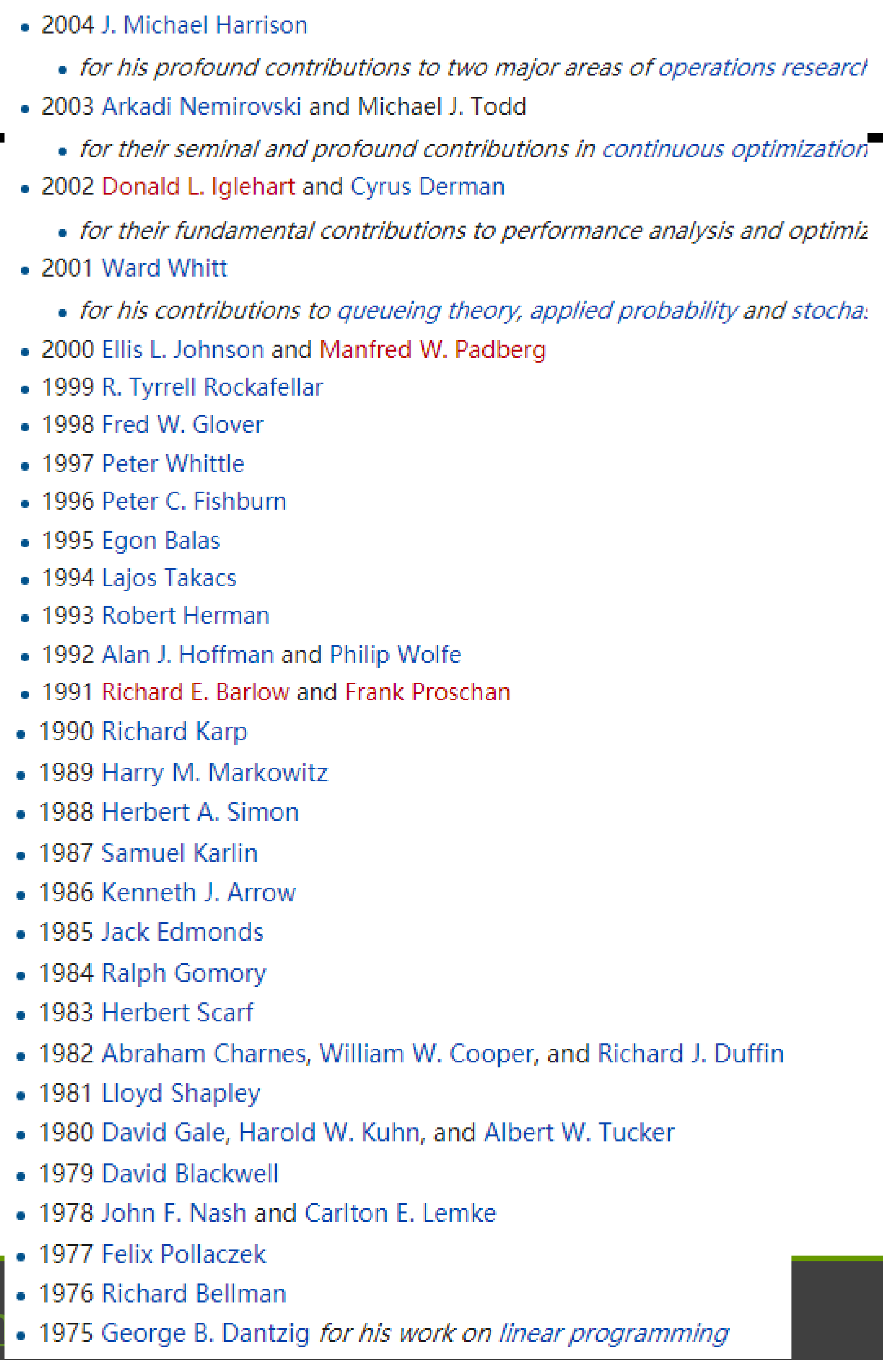
https://www.siam.org/prizes/sponsored/dantzig.php

Anecdote

□ Top Prizes w.r.t Optimization

John von Neumann Theory Prize (Operational Research)

- The **John von Neumann Theory Prize** of the Institute for Operations Research and the Management Sciences (INFORMS) is awarded annually to an individual (or sometimes a group) who has made fundamental and sustained contributions to theory in operations research and the management sciences. It is regarded the "**Nobel Prize**" of the field.
- George B. Dantzig is the 1st winner of this prize (1975)
  - ✓ 1975 George B. Dantzig *for his work on linear programming*



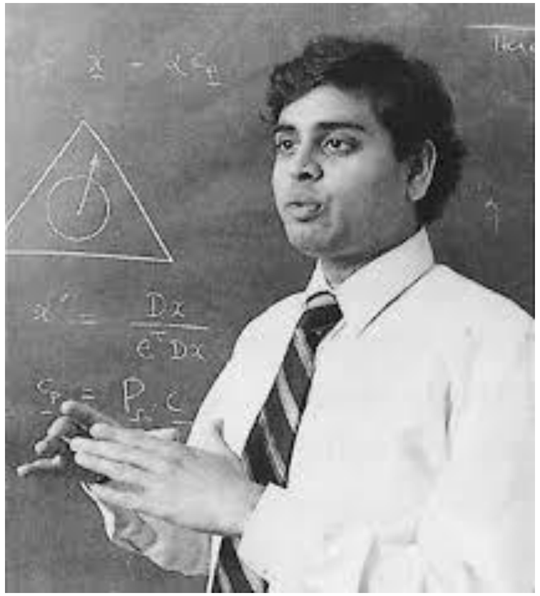
- 2004 J. Michael Harrison
  - *for his profound contributions to two major areas of operations researc*
- 2003 Arkadi Nemirovski and Michael J. Todd
  - *for their seminal and profound contributions in continuous optimization*
- 2002 Donald L. Iglehart and Cyrus Derman
  - *for their fundamental contributions to performance analysis and optimiz*
- 2001 Ward Whitt
  - *for his contributions to queueing theory, applied probability and stocha*
- 2000 Ellis L. Johnson and Manfred W. Padberg
- 1999 R. Tyrrell Rockafellar
- 1998 Fred W. Glover
- 1997 Peter Whittle
- 1996 Peter C. Fishburn
- 1995 Egon Balas
- 1994 Lajos Takacs
- 1993 Robert Herman
- 1992 Alan J. Hoffman and Philip Wolfe
- 1991 Richard E. Barlow and Frank Proschan
- 1990 Richard Karp
- 1989 Harry M. Markowitz
- 1988 Herbert A. Simon
- 1987 Samuel Karlin
- 1986 Kenneth J. Arrow
- 1985 Jack Edmonds
- 1984 Ralph Gomory
- 1983 Herbert Scarf
- 1982 Abraham Charnes, William W. Cooper, and Richard J. Duffin
- 1981 Lloyd Shapley
- 1980 David Gale, Harold W. Kuhn, and Albert W. Tucker
- 1979 David Blackwell
- 1978 John F. Nash and Carlton E. Lemke
- 1977 Felix Pollaczek
- 1976 Richard Bellman
- 1975 George B. Dantzig *for his work on linear programming*

https://en.wikipedia.org/wiki/John_von_Neumann_Th

- **Optimal control theory begins to develop as a separate discipline from CoV.**
  - Space race gives additional boost for research in optimal control theory

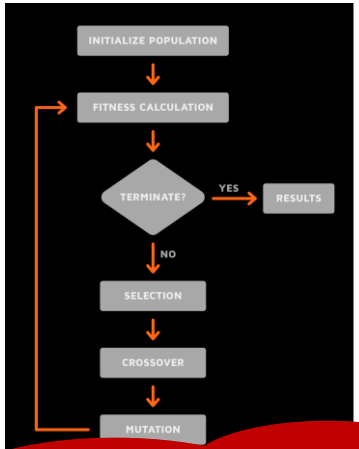
- **1957 Richard E. Bellman presents the optimality principle [优化原理]**
  - We'll meet this in **MDP** – Markov Decision Process [马尔科夫决策]
  - But you may have known it by the shortest path or critical path in network flow

- **1984 Narendra Karmarkar's polynomial time algorithm for LP-problems begins a boom of interior point methods [内点法].**
  - The first polynomial time algorithm for LP, the ellipsoid method [椭球法], was already presented by **Leonid Khachiyan** in 1979



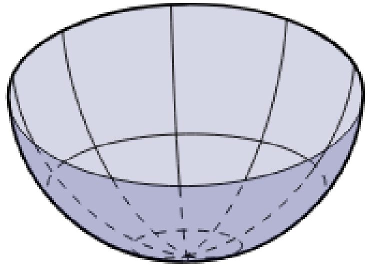
https://en.wikipedia.org/wiki/Leonid_Khachiyan

- 1980s as <u>computers</u> become more efficient, **<u>heuristic algorithms</u> [启发式算法]** for (global) optimization and large scale problems begin to gain popularity
  1. Genetic Algorithm [遗传算法]
  2. Simulated Annealing Algorithm [模拟退火算法]
  3. Ant Algorithm [蚁群算法]



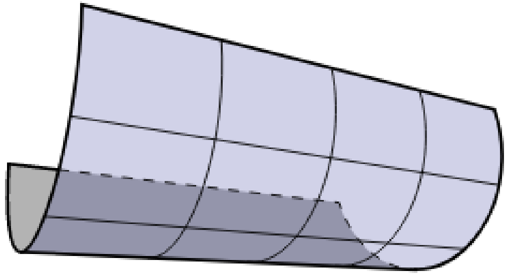
- **1990s the use of interior point methods exp optimization [SDP: 半正定规划]**
  1. A is (positive) **<u>semidefinite</u>** matrix, and write $A \succcurlyeq 0$, if all **eigenvalues** of A are **nonnegative**.
  2. A is (positive) **<u>definite</u>**, and write $A \succ 0$, if all **eigenvalues** of A are positive.

Here the math is Matrix (Vector of Vectors)



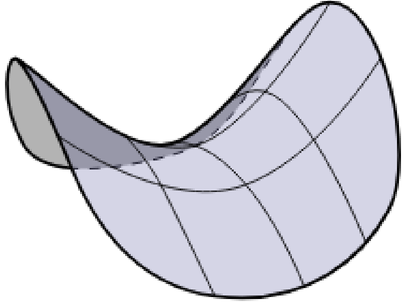
$x^2 + y^2$

(definite)



x^2

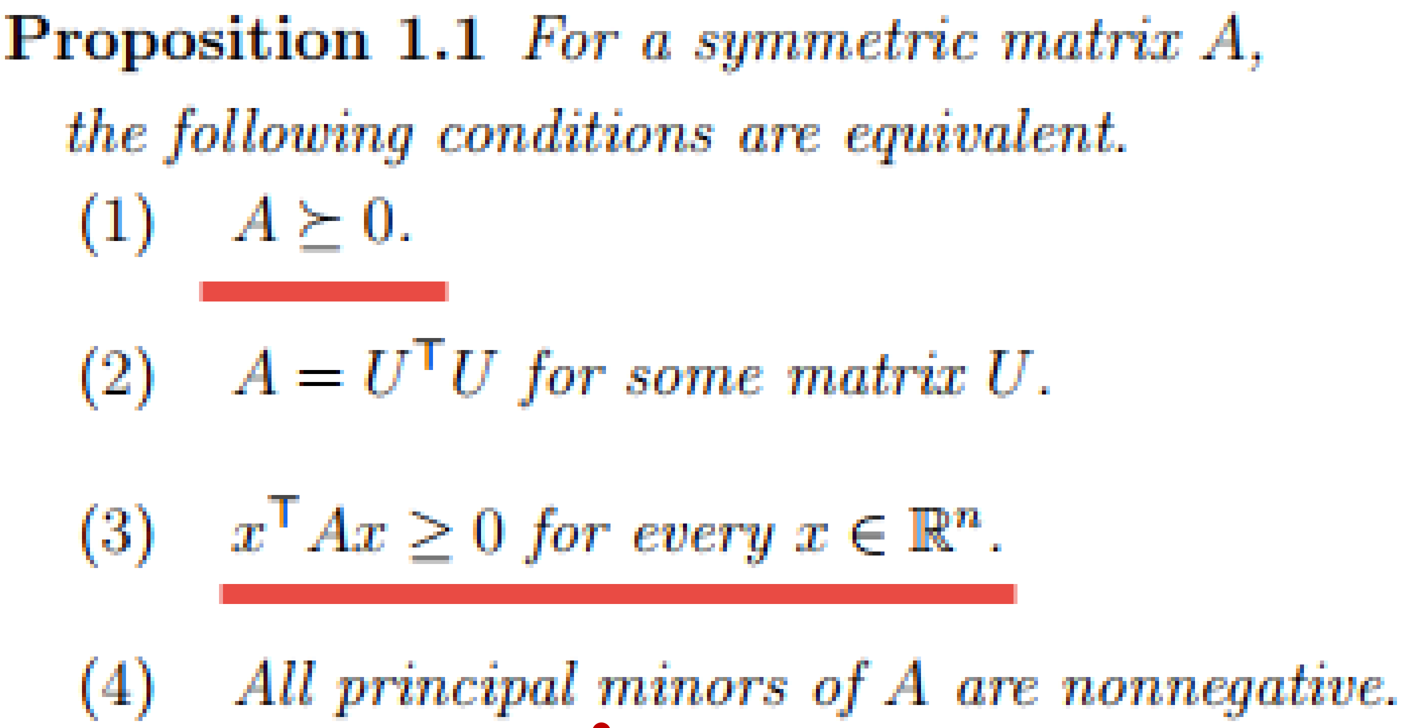
(semidefinite)



$x^2 - y^2$

(indefinite)

- **For 3 D (Extended to vectors), <u>Gradient</u> → First derivative, <u>Hessian matrix</u> → Second derivative**
  - $F(x, y) = x^2 + y^2$
  - Gradient: $\nabla F(x, y) = \begin{bmatrix} \frac{\partial F}{\partial x} \\ \frac{\partial F}{\partial y} \end{bmatrix}$
  - Hessian: $H_F = \begin{bmatrix} \frac{\partial^2 F}{\partial x^2} & \frac{\partial^2 F}{\partial x \partial y} \\ \frac{\partial^2 F}{\partial y \partial x} & \frac{\partial^2 F}{\partial y^2} \end{bmatrix}$
  - Necessary cond. for optimizer: $\nabla F(x, y) = 0$→ $\frac{\partial F}{\partial x} = 2x = 0$→$x = y = 0$
  - Hessian: $\boldsymbol{H_F} = \begin{bmatrix} \mathbf{2} & \mathbf{0} \\ \mathbf{0} & \mathbf{2} \end{bmatrix} \succcurlyeq \mathbf{0}$ is **definite**, which implies (0,0) is the global optimizer – **minimum**



Proposition 1.1 *For a symmetric matrix A, the following conditions are equivalent.*

(1) $A \succeq 0$.

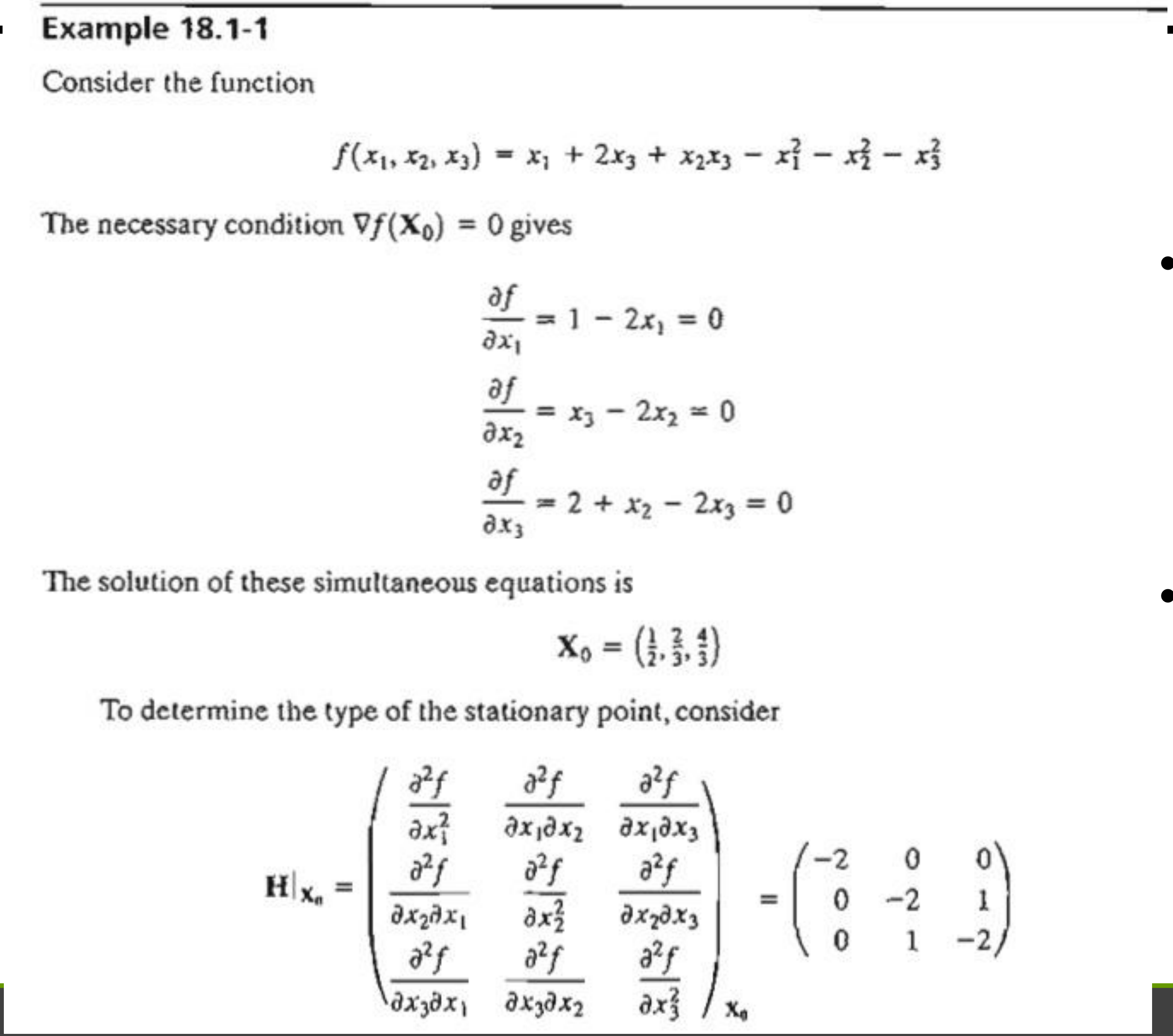
(2) $A = U^\top U$ *for some matrix* U.

(3) $x^\top A x \geq 0$ *for every* $x \in \mathbb{R}^n$.

(4) *All principal minors of A are nonnegative.*

Do you remember Principle minor [主子式]? ☺

http://www.readorrefer.in/article/Necessary-and-Sufficient-Conditions--Unconstrained-Problems_11261/



Example 18.1-1

Consider the function

$$f(x_1, x_2, x_3) = x_1 + 2x_3 + x_2x_3 - x_1^2 - x_2^2 - x_3^2$$

The necessary condition $\nabla f(\mathbf{X}_0) = 0$ gives

$$\frac{\partial f}{\partial x_1} = 1 - 2x_1 = 0$$

$$\frac{\partial f}{\partial x_2} = x_3 - 2x_2 = 0$$

$$\frac{\partial f}{\partial x_3} = 2 + x_2 - 2x_3 = 0$$

The solution of these simultaneous equations is

$$\mathbf{X}_0 = \left(\tfrac{1}{2}, \tfrac{2}{3}, \tfrac{4}{3}\right)$$

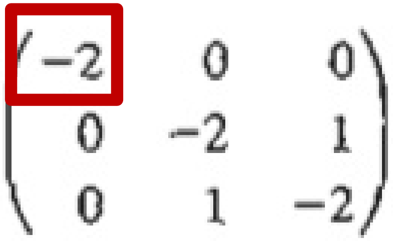
To determine the type of the stationary point, consider

$$\mathbf{H}|_{\mathbf{X}_0} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f}{\partial x_3 \partial x_1} & \frac{\partial^2 f}{\partial x_3 \partial x_2} & \frac{\partial^2 f}{\partial x_3^2} \end{pmatrix}_{\mathbf{X}_0} = \begin{pmatrix} -2 & 0 & 0 \\ 0 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix}$$

- The principal minor [主子式] determinants [行列式] of $H|_{x_0}$ have the values -2,4, and -6, respectively.

- Thus, $H|_{x_0}$ is negative-definite and x_0= (1/2, 2/3, 4/3) represents a **maximum point**.

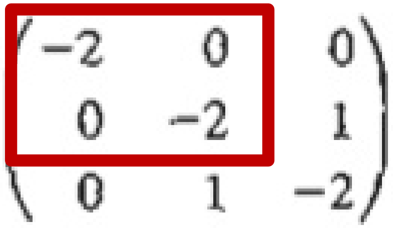
- 1st order leading principal minor



$$\begin{pmatrix} \boxed{-2} & 0 & 0 \\ 0 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix}$$

Determinant [行列式] is -2

- 2nd order leading principal minor

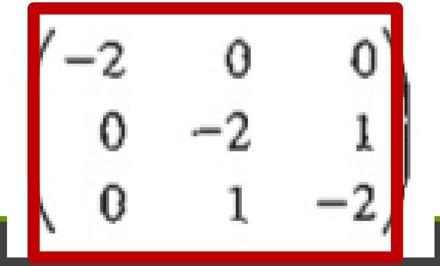


$$\begin{pmatrix} \boxed{\begin{matrix} -2 & 0 \\ 0 & -2 \end{matrix}} & \begin{matrix} 0 \\ 1 \end{matrix} \\ \begin{matrix} 0 & 1 \end{matrix} & -2 \end{pmatrix}$$

Determinant is

(-2)*(-2)=4

- 3rd order leading principal minor

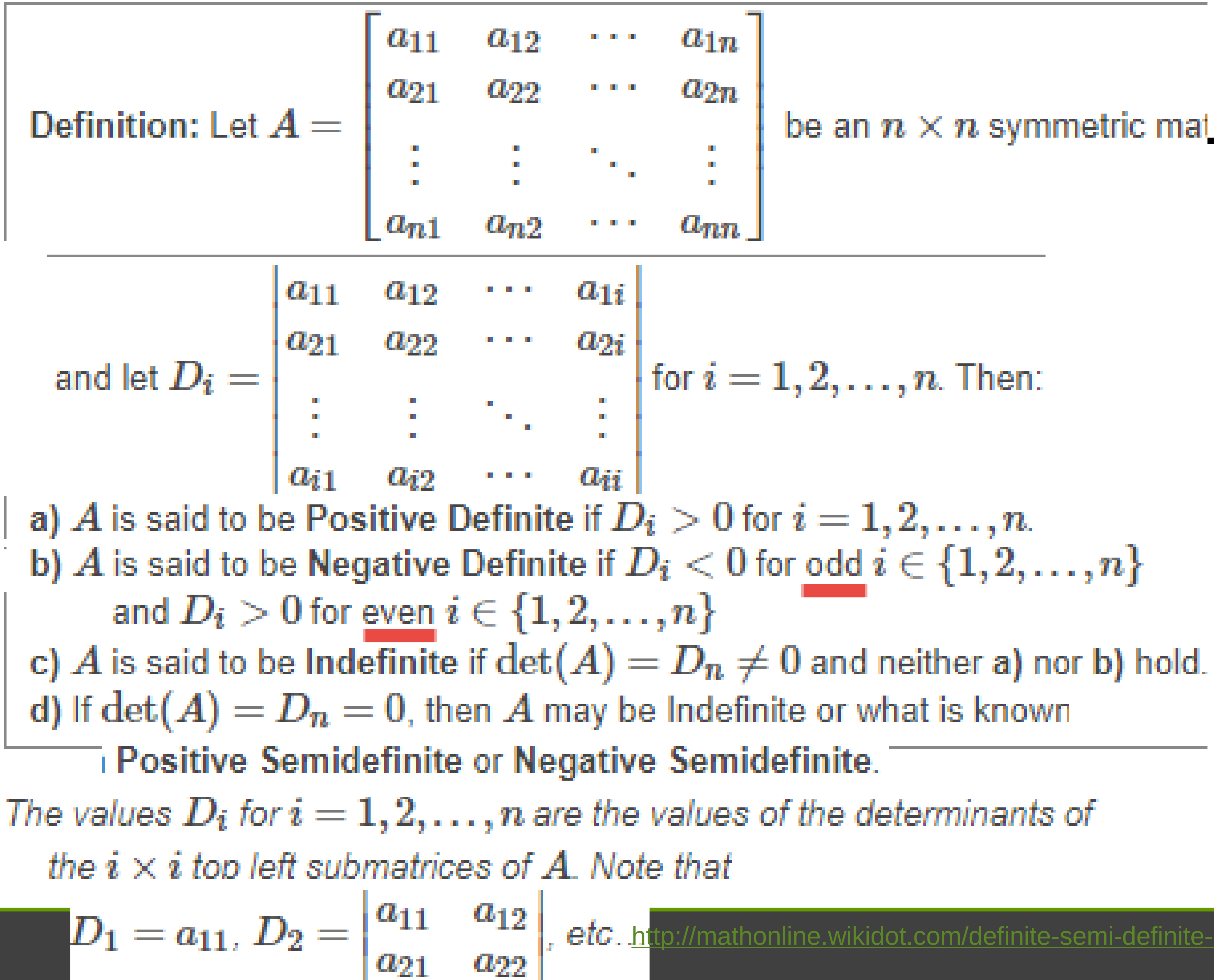


$$\boxed{\begin{pmatrix} -2 & 0 & 0 \\ 0 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix}}$$

Determinant is

$(-2)*(-1)^{1+1}[(-2)*(-2)-1*1]$

= (-2)*1*[4-1] = -6



Definition: Let $A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$ be an $n \times n$ symmetric mat

and let $D_i = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1i} \\ a_{21} & a_{22} & \cdots & a_{2i} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ii} \end{vmatrix}$ for $i = 1, 2, \ldots, n$. Then:

a) A is said to be **Positive Definite** if $D_i > 0$ for $i = 1, 2, \ldots, n$.

b) A is said to be **Negative Definite** if $D_i < 0$ for odd $i \in \{1, 2, \ldots, n\}$ and $D_i > 0$ for even $i \in \{1, 2, \ldots, n\}$

c) A is said to be **Indefinite** if $\det(A) = D_n \neq 0$ and neither **a)** nor **b)** hold.

d) If $\det(A) = D_n = 0$, then A may be Indefinite or what is known **Positive Semidefinite** or **Negative Semidefinite**.

The values D_i for $i = 1, 2, \ldots, n$ are the values of the determinants of the $i \times i$ top left submatrices of A. Note that

$D_1 = a_{11}$, $D_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$, *etc.*

http://mathonline.wikidot.com/definite-semi-definite-and-indefinite-matrices

20th century - present

3 powerful math tools
– Lagrange Multiplier, Duality, KKT

- **Extended to NLP (Nonlinear Programming) with previous ways or new ideas**
  - **Derivatives are general**
    - 1-D: 1st and 2nd derivative
    - n-D: Gradient [梯度] and Hessian matrix
  - **Lagrange transform/multiplier** [拉格朗日乘子]
    - Convert constrained optimization problems into unconstrained
  - **Duality** [对偶] proposed by von Neumann
    - Min max → Max min
  - **KKT**： Karush–Kuhn–Tucker [卡羅需 - 庫恩 - 塔克條件]
    - Necessary condition for optimization problems
  - **Numerical computation** – 数值计算
    - (Gradient) Descent [(梯度)下降]，Newton [牛顿法]，Quasi Newton [拟牛顿法]…

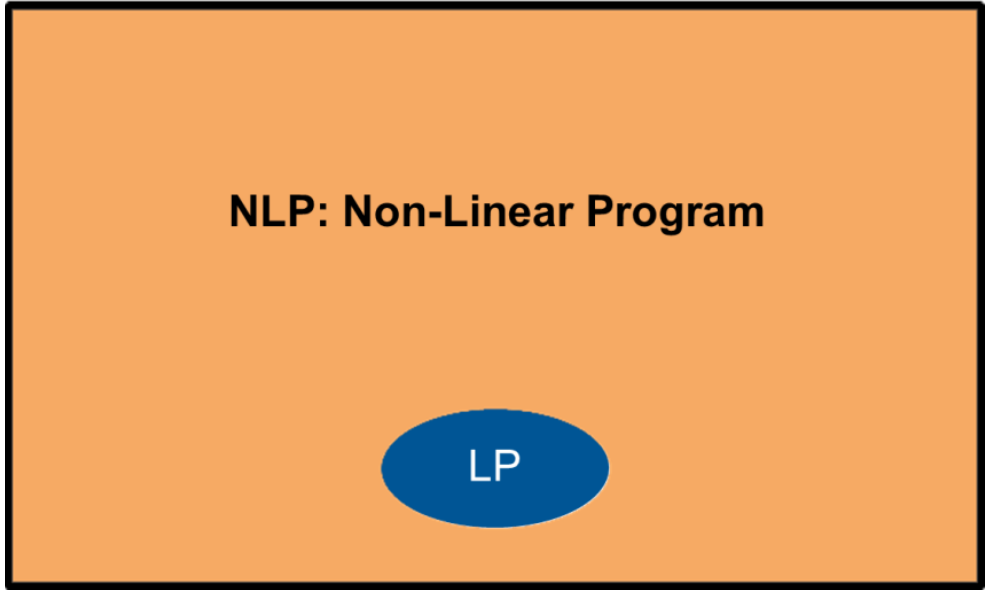
一点优化的技巧 (OPTIMIZATION)

- □ **还是喜欢从历史入手 –**
  - ● Optimization? 最优化?
  - ● A brief history
  - ● **Calculus** ([partial] derivative) + **Linear Algebra** – modern tools for optimization
    - ➢ Calculus of variations [变分法]
    - ➢ Operational Research [运筹学]
- □ **优化问题概览 及其解决方案**
  - ● LP, NLP (QP,SOCP,SDP, CP, PP)
  - ● Solutions: Descent, Newton, …

Now we have many optimization (programming)

NLP: Natural Language Processing

- **Generally, 2 categories**
  - LP and NLP: Non-Linear Programming



NLP: Non-Linear Program

LP

$$
\begin{aligned}
\min\ z &= c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{s. t.}\quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leqslant b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leqslant b_2 \\
& \qquad\vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leqslant b_m \\
& x_1,\ x_2,\ \cdots,\ x_n \geqslant 0
\end{aligned}
$$

  - ➢ NLP: One of objective function or constrained functions is non linear
    - ✓ By linear, the order of the variables is 1.

- □ LP

$$
\begin{aligned}
\min z &= c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{s. t.}\quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leqslant b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leqslant b_2 \\
& \qquad \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leqslant b_m \\
& x_1,\ x_2,\ \cdots,\ x_n \geqslant 0
\end{aligned}
$$

- □ NLP

➢ $\max(xy)$

➢ $s.t$

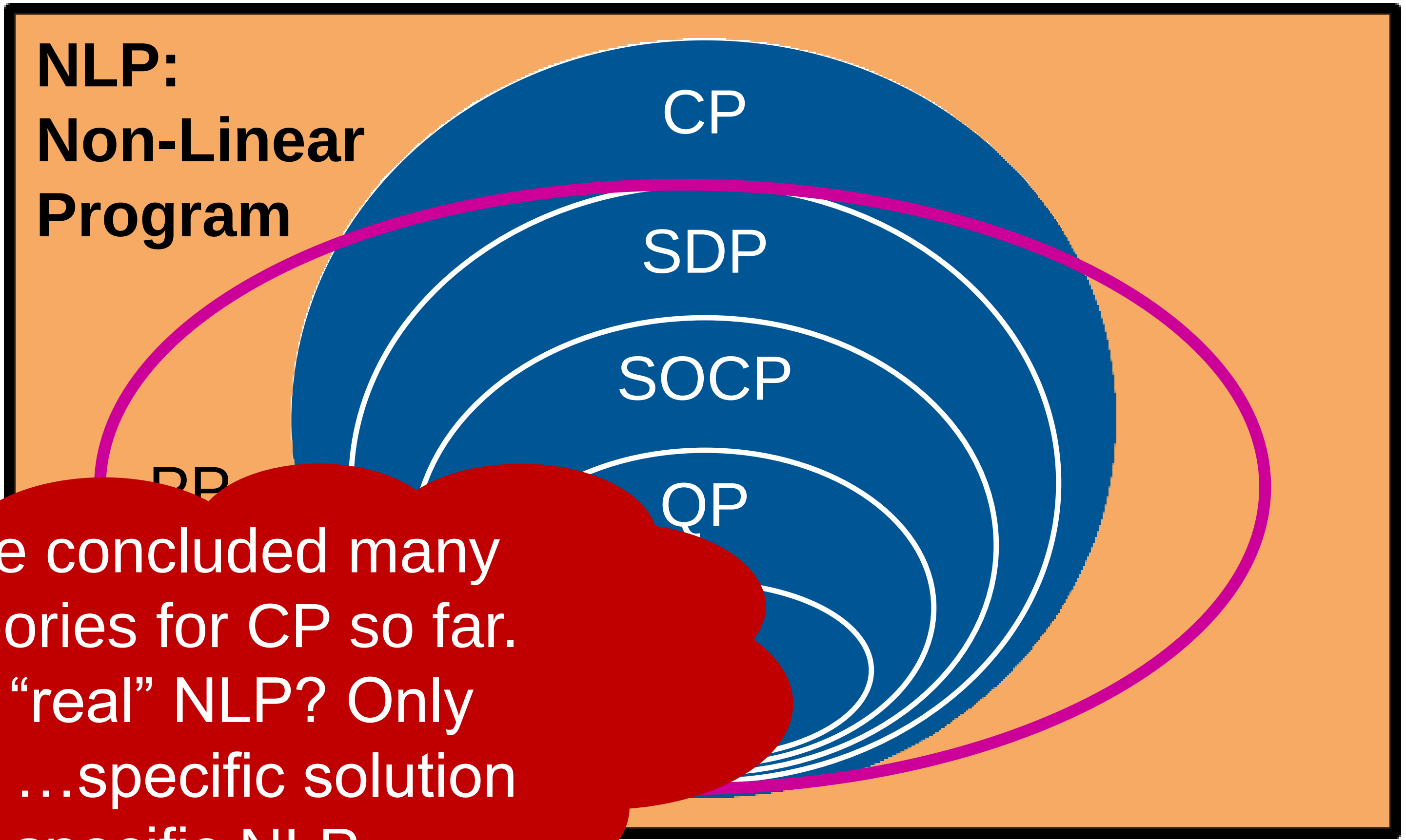
- ✓ $x + y = C$
- ✓ $x > 0, y > 0$

$$\min \int_{x_0}^{x_1} \left(\frac{1 + (y')^2}{y} \right)^{1/2} dx$$

$$F - y' F_{y'} = \text{constant}$$

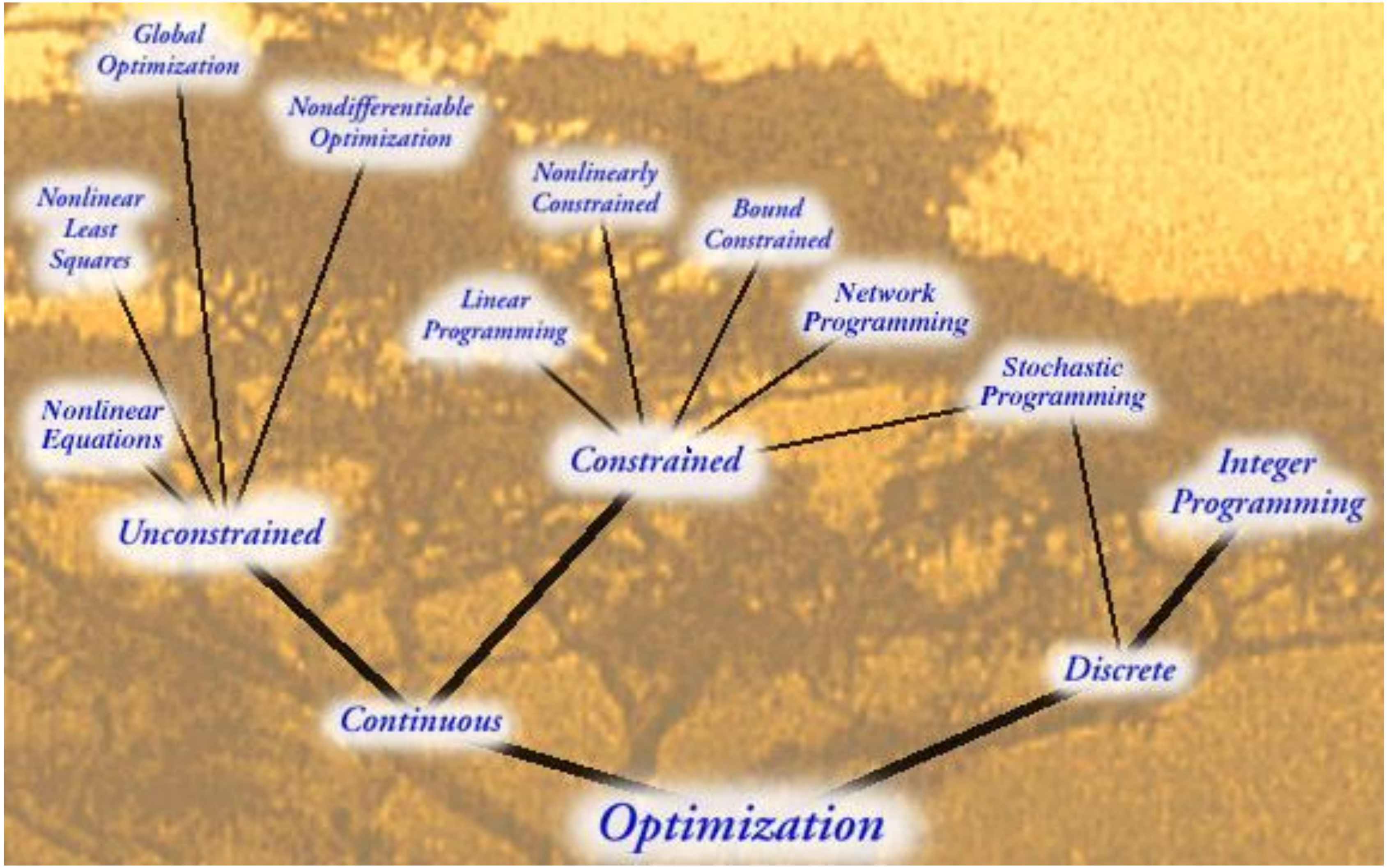
□ More precise about Optimization

- According to LU Wu-Sheng@UofVictoria and Stephen Boyd@Stanford



LP: Linear Programming
QP: Quadratic Programming 二次规划
SOCP: Second Order Cone Programming 二阶锥规划
SDP: Semi-definite Programming 半正定规划
CP: Convex 凸规划
PP: Polynomial Programming

We have concluded many rules/theories for CP so far. But for “real” NLP? Only heuristic …specific solution for specific NLP


Global Optimization
Nondifferentiable Optimization
Nonlinear Least Squares
Nonlinearly Constrained
Bound Constrained
Linear Programming
Network Programming
Stochastic Programming
Nonlinear Equations
Constrained
Integer Programming
Unconstrained
Discrete
Continuous
Optimization

Convex optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le b_i, \quad i = 1, \dots, m \end{array}$$

- objective and constraint functions are convex:

$$f_i(\alpha x + \beta y) \le \alpha f_i(x) + \beta f_i(y)$$

if $\alpha + \beta = 1$, $\alpha \ge 0$, $\beta \ge 0$

- includes least-squares problems and linear programs as special cases

Convex and concave functions in two variables

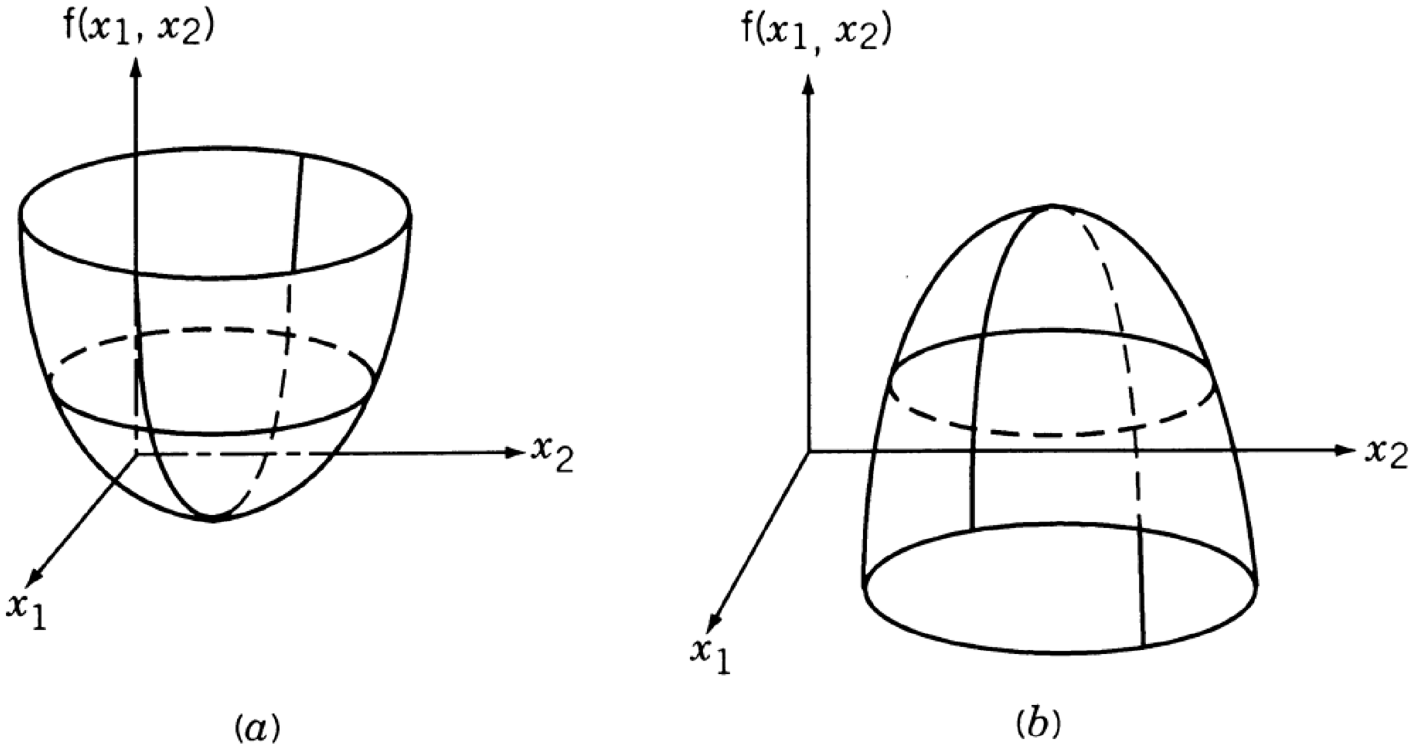


Figure A.2 Functions of two variables: (*a*) convex function in two variables; (*b*) concave function in two variables.

The general form

$$\begin{array}{lll} \text{Minimize} & f(x) & \\ \text{Subject to} & g_j(x) \geqslant 0 & \text{for } j = 1, 2, \ldots, J \\ & h_k(x) = 0 & \text{for } k = 1, 2, \ldots, K \\ & x = (x_1, x_2, \ldots, x_N) & \end{array}$$

- **4 specific types according to difficulty**
  - No constraints,

    $$\begin{array}{ll} \text{Minimize} & f(x) \\ & x = (x_1, x_2, \ldots, x_N) \end{array}$$

  - Only equality constraints,

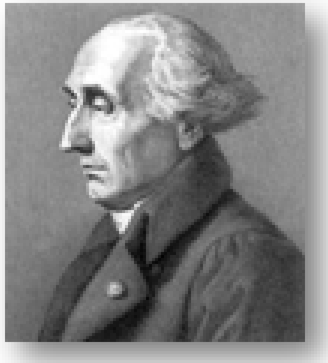
    $$\begin{array}{lll} \text{Minimize} & f(x) & \\ \text{Subject to} & h_k(x) = 0 & \text{for } k = 1, 2, \ldots, K \\ & x = (x_1, x_2, \ldots, x_N) & \end{array}$$

  - Only inequality constraints,

    $$\begin{array}{lll} \text{Minimize} & f(x) & \\ \text{Subject to} & g_j(x) \geqslant 0 & \text{for } j = 1, 2, \ldots, J \\ & x = (x_1, x_2, \ldots, x_N) & \end{array}$$

  - Hybrid: equality and inequality constraints.

Lagrange Multiplier



The Lagrange method writes the constrained optimization problem in the following form

Objective

Choice variables

$$\max_{x,y} f(x,y)$$

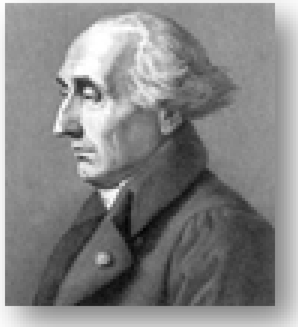
Constraint(s)

$$subject\ to\quad g(x,y) \geq 0$$

The problem is then rewritten as follows

$$1 = f(x,y) + \lambda g(x,y)$$

Multiplier (assumed greater or equal to zero)



So, we have our Lagrangian function….

$$1 = f(x,y) + \lambda g(x,y)$$

We need the derivatives with respect o both 'x' and 'y' to be zero

$$\boxed{\begin{aligned} 1_x &= f_x(x,y) + \lambda g_x(x,y) = 0 \\ 1_y &= f_y(x,y) + \lambda g_y(x,y) = 0 \end{aligned}}$$

And then we have the "multiplier conditions"

$$\boxed{\lambda \geq 0 \qquad g(x,y) \geq 0 \qquad \lambda g(x,y) = 0}$$

□ Example B:

$$\text{Maximize} \quad f(x) = x_1 + x_2$$

$$\text{Subject to} \quad x_1^2 + x_2^2 = 1$$

$$L(x; v) = x_1 + x_2 - v(x_1^2 + x_2^2 - 1)$$

$$\frac{\partial L}{\partial x_1} = 1 - 2vx_1 = 0$$

$$\frac{\partial L}{\partial x_2} = 1 - 2vx_2 = 0$$

$$h_1(x) = x_1^2 + x_2^2 - 1 = 0$$

$$L(x;\, v) = x_1 + x_2 - v(x_1^2 + x_2^2 - 1)$$

$$(x^{(1)};\, v_1) = \left(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}; -\sqrt{\frac{1}{2}}\right)$$

$$(x^{(2)};\, v_2) = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}; \sqrt{\frac{1}{2}}\right)$$

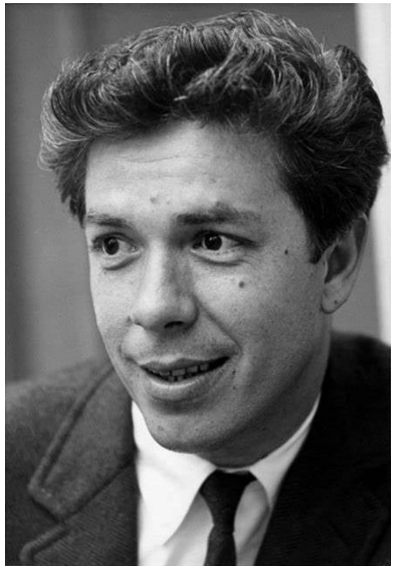
$$H_L(x^{(1)};\, v_1) = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix}$$

positive definite

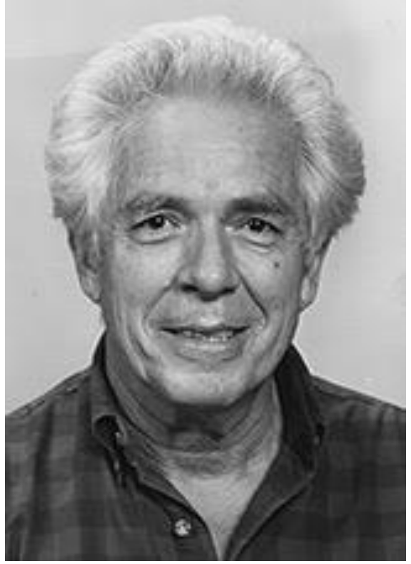
$$H_L(x^{(2)};\, v_2) = \begin{bmatrix} -\sqrt{2} & 0 \\ 0 & -\sqrt{2} \end{bmatrix}$$

negative definite

$$x_1^{o} \doteq x_2^{o} = 1/\sqrt{2}$$



William Karush



Harold W. Kuhn

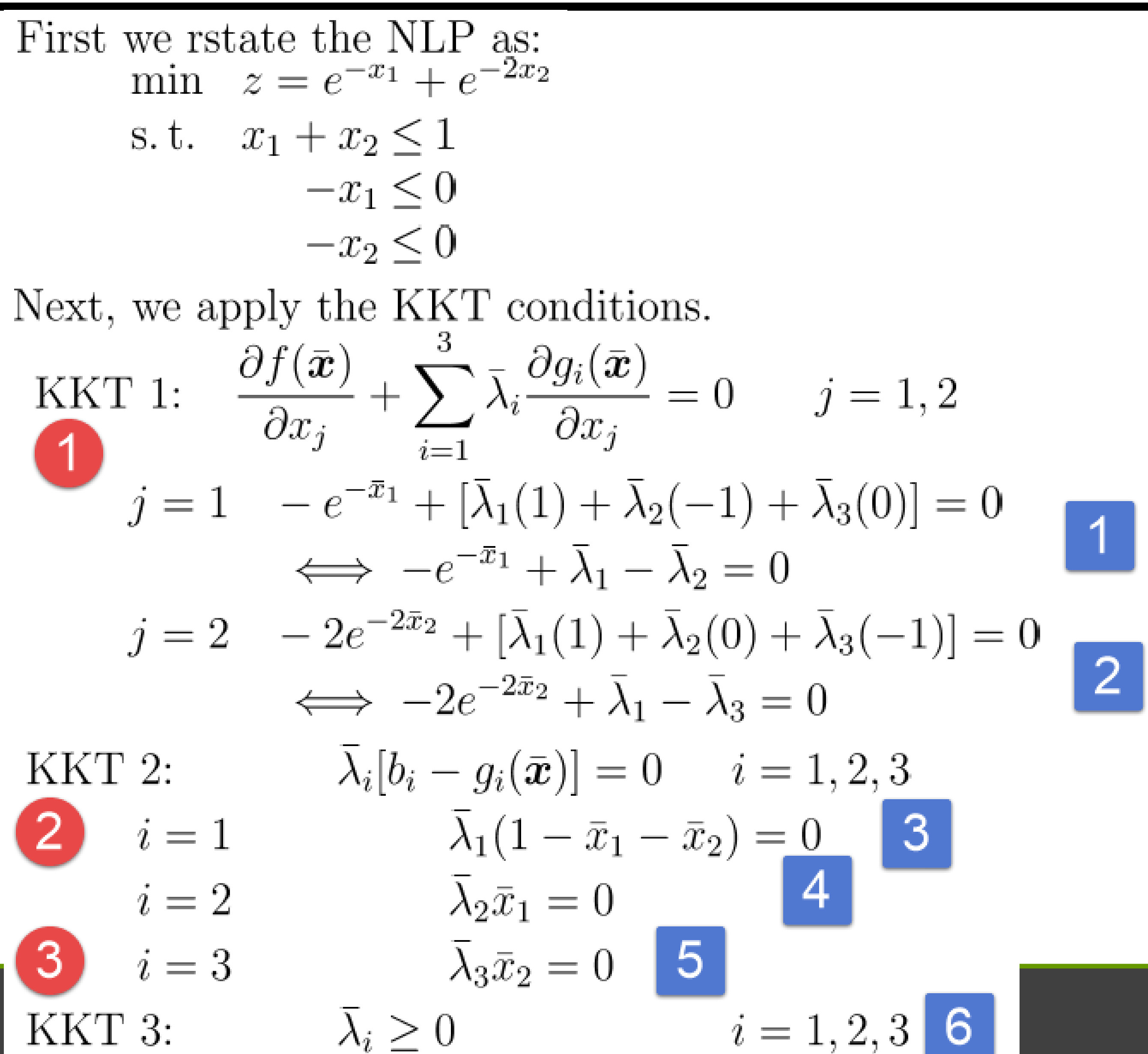
Albert William Tucker

https://en.wikipedia.org/wiki/Harold_W._Kuhn

https://en.wikipedia.org/wiki/Albert_W._Tucker

https://en.wikipedia.org/wiki/William_Karush

$$\begin{aligned} \min \quad & z = e^{-x_1} + e^{-2x_2} \\ \text{s.t.} \quad & x_1 + x_2 \le 1 \\ & x_1, x_2 \ge 0 \end{aligned}$$



First we rstate the NLP as:

$$\begin{aligned} \min \quad & z = e^{-x_1} + e^{-2x_2} \\ \text{s.t.} \quad & x_1 + x_2 \le 1 \\ & -x_1 \le 0 \\ & -x_2 \le 0 \end{aligned}$$

Next, we apply the KKT conditions.

KKT 1: $$\frac{\partial f(\bar{\boldsymbol{x}})}{\partial x_j} + \sum_{i=1}^{3} \bar{\lambda}_i \frac{\partial g_i(\bar{\boldsymbol{x}})}{\partial x_j} = 0 \qquad j = 1,2$$

1

$$j = 1 \qquad -e^{-\bar{x}_1} + [\bar{\lambda}_1(1) + \bar{\lambda}_2(-1) + \bar{\lambda}_3(0)] = 0$$

$$\iff \quad -e^{-\bar{x}_1} + \bar{\lambda}_1 - \bar{\lambda}_2 = 0$$ 1

$$j = 2 \qquad -2e^{-2\bar{x}_2} + [\bar{\lambda}_1(1) + \bar{\lambda}_2(0) + \bar{\lambda}_3(-1)] = 0$$

$$\iff \quad -2e^{-2\bar{x}_2} + \bar{\lambda}_1 - \bar{\lambda}_3 = 0$$ 2

KKT 2: $$\bar{\lambda}_i[b_i - g_i(\bar{\boldsymbol{x}})] = 0 \qquad i = 1,2,3$$

2 $$i = 1 \qquad \bar{\lambda}_1(1 - \bar{x}_1 - \bar{x}_2) = 0$$ 3

$$i = 2 \qquad \bar{\lambda}_2 \bar{x}_1 = 0$$ 4

3 $$i = 3 \qquad \bar{\lambda}_3 \bar{x}_2 = 0$$ 5

KKT 3: $$\bar{\lambda}_i \ge 0 \qquad i = 1,2,3$$ 6

- Thus we must solve equations (1) - (6) for x_1, x_2 and λ_1, λ_2, λ_3 with the condition that x_1 and x_2 must also be feasible.

$$\begin{aligned} \text{s. t.} \quad x_1 + x_2 &\leq 1 \\ x_1, x_2 &\geq 0 \end{aligned}$$

  - These equations are nonlinear, and there is no general method to solve nonlinear equations analytically

- **For our system, note that since we must have x_i>=0, then either x_i>0 or x_i=0. Therefore, we have 4 situations**

Case 1. $\bar{x}_1 = 0,\ \bar{x}_2 = 0.$

From (3): $\bar{\lambda}_1 = 0.$

Now from (1): $-e^0 + 0 - \bar{\lambda}_2 = 0$

$$\iff \ -1 - \bar{\lambda}_2 = 0 \ \iff \ \bar{\lambda}_2 = -1,$$

which is not valid.

Case 2. $\bar{x}_1 = 0$, $\bar{x}_2 > 0$.

From (5): $\bar{\lambda}_3 = 0$.

Now from (2): $-2e^{-2\bar{x}_2} + \bar{\lambda}_1 - 0 = 0$

$\iff \bar{\lambda}_1 = 2e^{-2\bar{x}_2} > 0$. Since $\bar{\lambda}_1 > 0$,

equation (3) implies that $1 - \bar{x}_1 - \bar{x}_2 = 0$

$\iff 1 - 0 - \bar{x}_2 = 0 \iff \bar{x}_2 = 1$.

And so this gives $\bar{\lambda}_1 = 2e^{-2}$.

And now from (1) we have $-e^0 + 2e^{-2} - \bar{\lambda}_2 = 0$

$\iff \bar{\lambda}_2 = 2e^{-2} - 1 \approx -0.729 < 0$

which is not valid.

Case 3. $\bar{x}_1 > 0$, $\bar{x}_2 = 0$.

From (4): $\bar{\lambda}_2 = 0$.

Now from (1): $-e^{-\bar{x}_1} + \bar{\lambda}_1 - 0 = 0 \iff \bar{\lambda}_1 = e^{-\bar{x}_1} > 0$. Since $\bar{\lambda}_1 > 0$, equation (3) implies that $1 - \bar{x}_1 - \bar{x}_2 = 0 \iff 1 - \bar{x}_1 - 0 = 0 \iff \bar{x}_1 = 1$. This yields $\bar{\lambda}_1 = e^{-1}$.

And now from (2) we have $-2e^0 + e^{-1} - \bar{\lambda}_3 = 0 \iff \bar{\lambda}_3 = e^{-1} - 2 \approx -1.632 < 0$ which is not valid.

Case 4. $\bar{x}_1 > 0$, $\bar{x}_2 > 0$.

(Since the first three cases yield invalid solutions, this case must give us the correct solution.)

From (4): $\bar{\lambda}_2 = 0$.

From (5): $\bar{\lambda}_3 = 0$.

Equations (1) and (2) now yield $\bar{\lambda}_1 = e^{-\bar{x}_1}$ and $\bar{\lambda}_1 = 2e^{-2\bar{x}_2}$, respectively. Since $\bar{\lambda}_1 = e^{-\bar{x}_1} > 0$, equation (3) implies $1 - \bar{x}_1 - \bar{x}_2 = 0 \iff \bar{x}_1 = 1 - \bar{x}_2$. Using this result and equating the two expressions for $\bar{\lambda}_1$ yields

$$\begin{aligned}
& e^{-\bar{x}_1} = 2e^{-2\bar{x}_2} \\
\iff \quad & e^{-\bar{x}_1} = e^{\ln 2 - 2\bar{x}_2} \\
\iff \quad & -\bar{x}_1 = \ln 2 - 2\bar{x}_2 \\
\iff \quad & -(1 - \bar{x}_2) = \ln 2 - 2\bar{x}_2 \\
\iff \quad & \bar{x}_2 = \frac{1}{3}(1 + \ln 2)
\end{aligned}$$

from which we get $\bar{x}_1 = \frac{1}{3}(2 - \ln 2)$ and $\bar{\lambda}_1 = 2^{1/3}e^{-2/3}$.

Thus, the solution to the system of equations (1)–(6) is $\bar{x}_1 = \frac{1}{3}(2 - \ln 2)$, $\bar{x}_2 = \frac{1}{3}(1 + \ln 2)$, $\bar{\lambda}_1 = 2^{1/3}e^{-2/3}$, $\bar{\lambda}_2 = 0$, and $\bar{\lambda}_3 = 0$; furthermore, $\bar{x}_1$ and $\bar{x}_2$ are the optimal solution values to the original NLP. To finish we find the optimal z-value:

$$\begin{aligned}
z_{\max} &= e^{-\bar{x}_1} + e^{-2\bar{x}_2} \\
&= 3(2e)^{-2/3}
\end{aligned}$$

Regularization [正则化] skill

- **By adding some regularization part into the objective function, we can confine the shape of the target parameters**
  - Widespread used in **ML** (Machine Learning), **CV** (Computer Vision), etc.

岭回归

Ridge Regression的优化目标为:

$$\beta^* = argmin_\beta \frac{1}{n}\|y - X\beta\|_2^2 + \boxed{\lambda\|\beta\|_2^2}$$

L2 norm [L2 范式] $\|X\|_2 = \sqrt{\sum_{i=1}^{M} x_i^2}$

套索算法

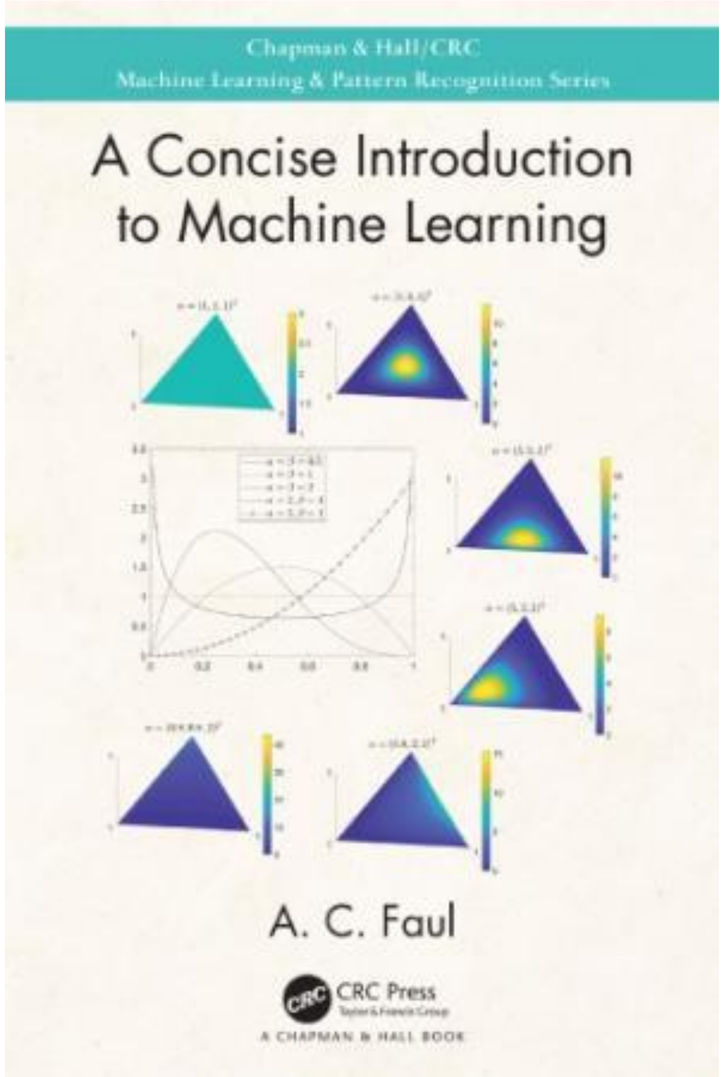
Lasso的优化目标为:

L1 norm [L1范式] – where p=1, $\|X\|_1 = \sum_{i=1}^{M} |x_i|$

L0 norm [L0范式] – Special: least non-zeros

$$\beta^* = argmin_\beta \frac{1}{n}\|y - X\beta\|_2^2 + \boxed{\lambda\|\beta\|_1}$$

Lasso算法（Least Absolute Shrinkage and Selection Operator，又译最小绝对值收敛和选择算子、套索算法）



Chapman & Hall/CRC
Machine Learning & Pattern Recognition Series

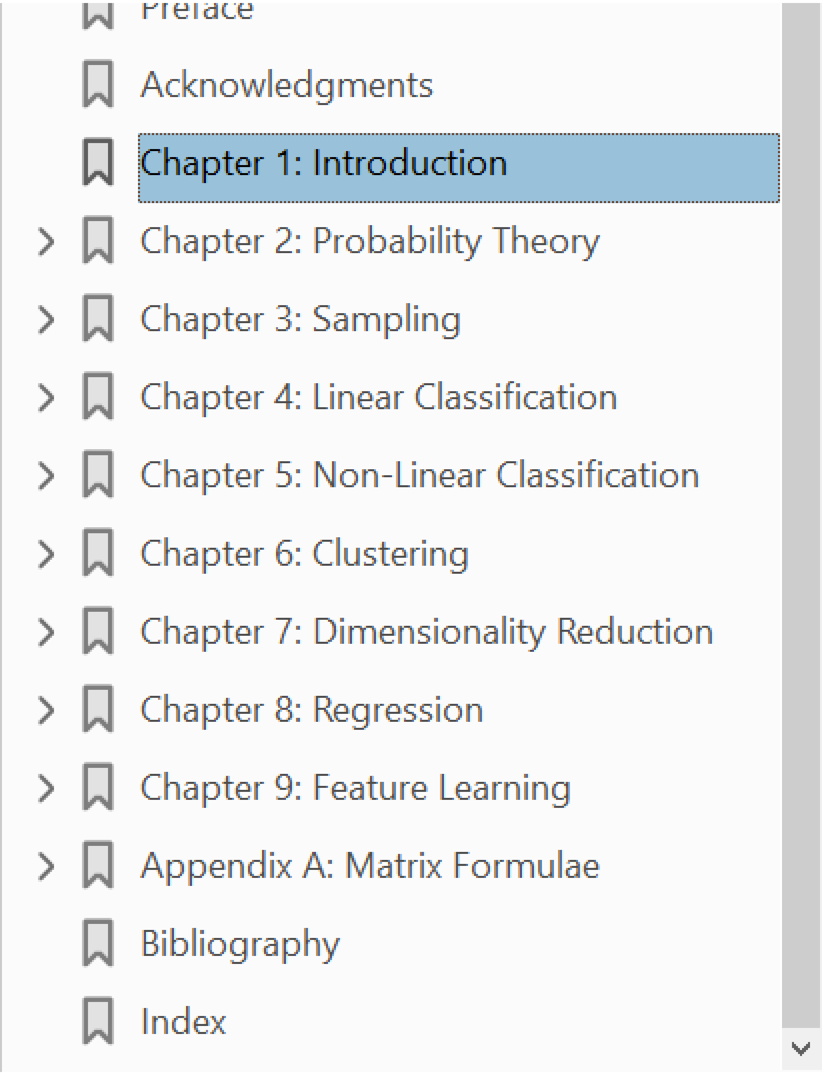
A Concise Introduction to Machine Learning

A. C. Faul

CRC Press
Taylor & Francis Group

A CHAPMAN & HALL BOOK

- A Concise Introduction to Machine Learning
- *Anita C. Faul*
- **2020**



一点优化的技巧 (OPTIMIZATION)

□ **还是喜欢从历史入手 –**

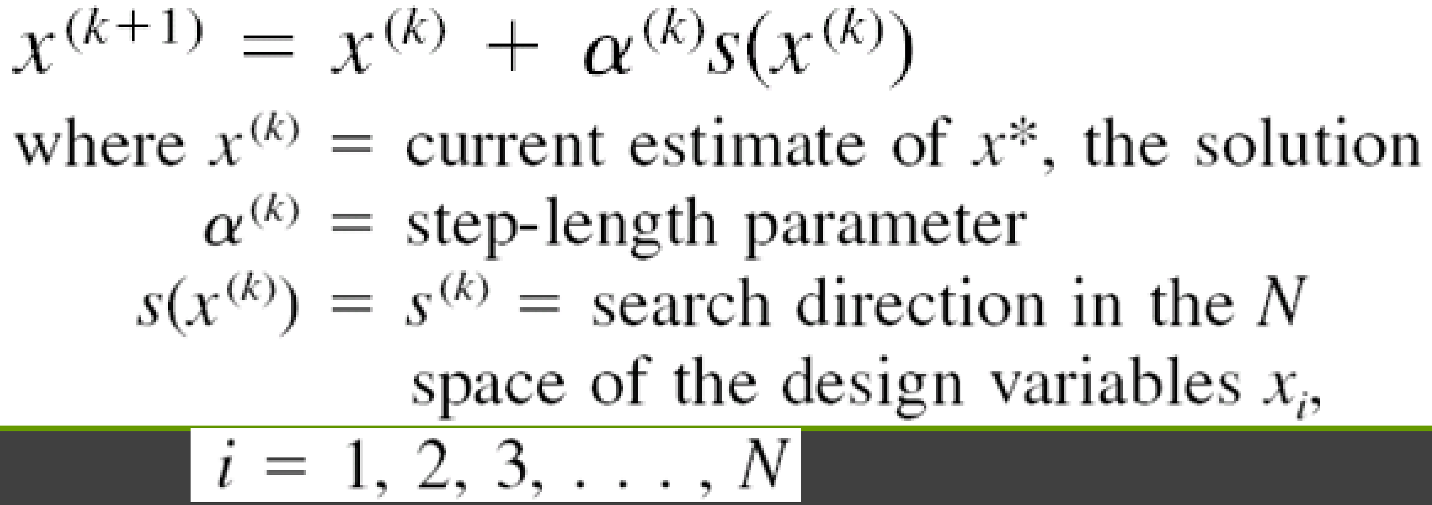
- Optimization? 最优化?
- A brief history
- **Calculus** ([partial] derivative) + **Linear Algebra** – modern tools for optimization
  - ➢ Calculus of variations [变分法]
  - ➢ Operational Research [运筹学]

□ **优化问题概览 及其解决方案**

- LP, NLP (QP,SOCP,SDP, CP, PP)
- Solutions: Descent, Newton, …

Numeric Methods

- **Generally we focus on the numeric methods for unconstrained Ops**
  - Only -Equality constrained OP could be converted to unconstrained by using Lagrange Multiplier directly
  - Part of Inequality (Only or Hybrid) constrained OP could be converted to unconstrained – KKT 1
- **All of the methods considered here employ a similar iteration procedure:** **Gradient Descent Method [梯度下降法]**



$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} s(x^{(k)})$$

where $x^{(k)}$ = current estimate of x^*, the solution

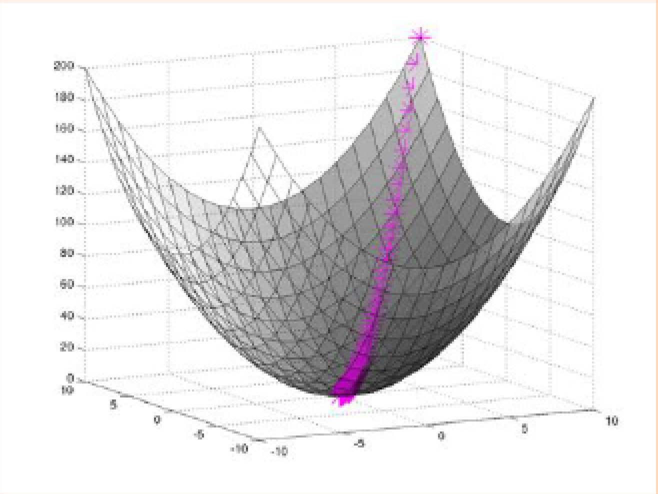
$\alpha^{(k)}$ = step-length parameter

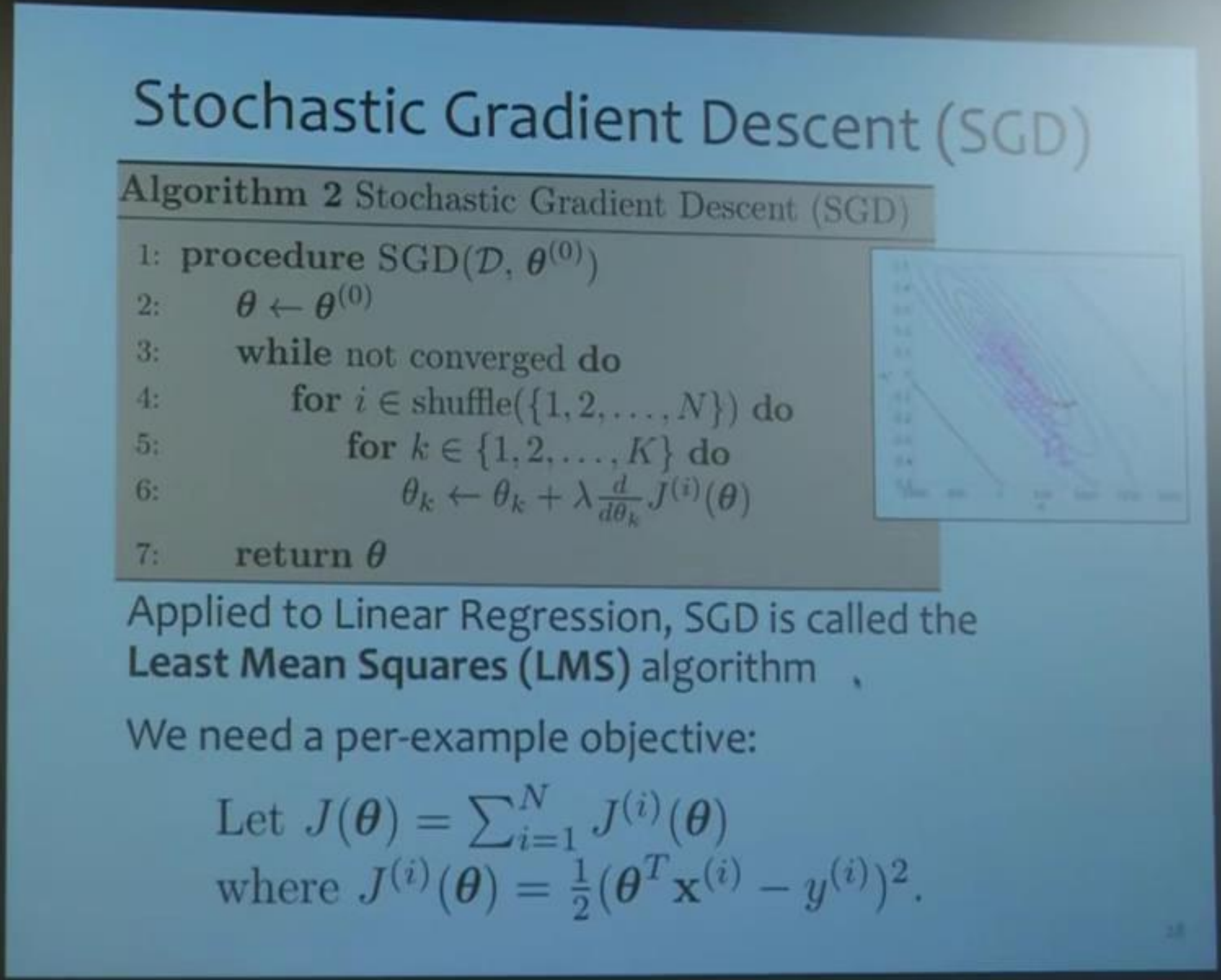
$s(x^{(k)}) = s^{(k)}$ = search direction in the N space of the design variables x_i, $i = 1, 2, 3, \ldots, N$

Gradient Descent

Algorithm 1 Gradient Descent

```
procedure GD($\mathcal{D}$, $\boldsymbol{\theta}^{(0)}$)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(0)}$
    while not converged do
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$
    return $\boldsymbol{\theta}$
```





Stochastic Gradient Descent (SGD)

Algorithm 2 Stochastic Gradient Descent (SGD)

```
procedure SGD($\mathcal{D}$, $\boldsymbol{\theta}^{(0)}$)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(0)}$
    while not converged do
        for $i \in$ shuffle($\{1, 2, \dots, N\}$) do
            for $k \in \{1, 2, \dots, K\}$ do
                $\theta_k \leftarrow \theta_k + \lambda \frac{d}{d\theta_k} J^{(i)}(\boldsymbol{\theta})$
    return $\boldsymbol{\theta}$
```

Applied to Linear Regression, SGD is called the **Least Mean Squares (LMS)** algorithm

We need a per-example objective:

$$\text{Let } J(\boldsymbol{\theta}) = \textstyle\sum_{i=1}^{N} J^{(i)}(\boldsymbol{\theta})$$
$$\text{where } J^{(i)}(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta}^T \mathbf{x}^{(i)} - y^{(i)})^2.$$

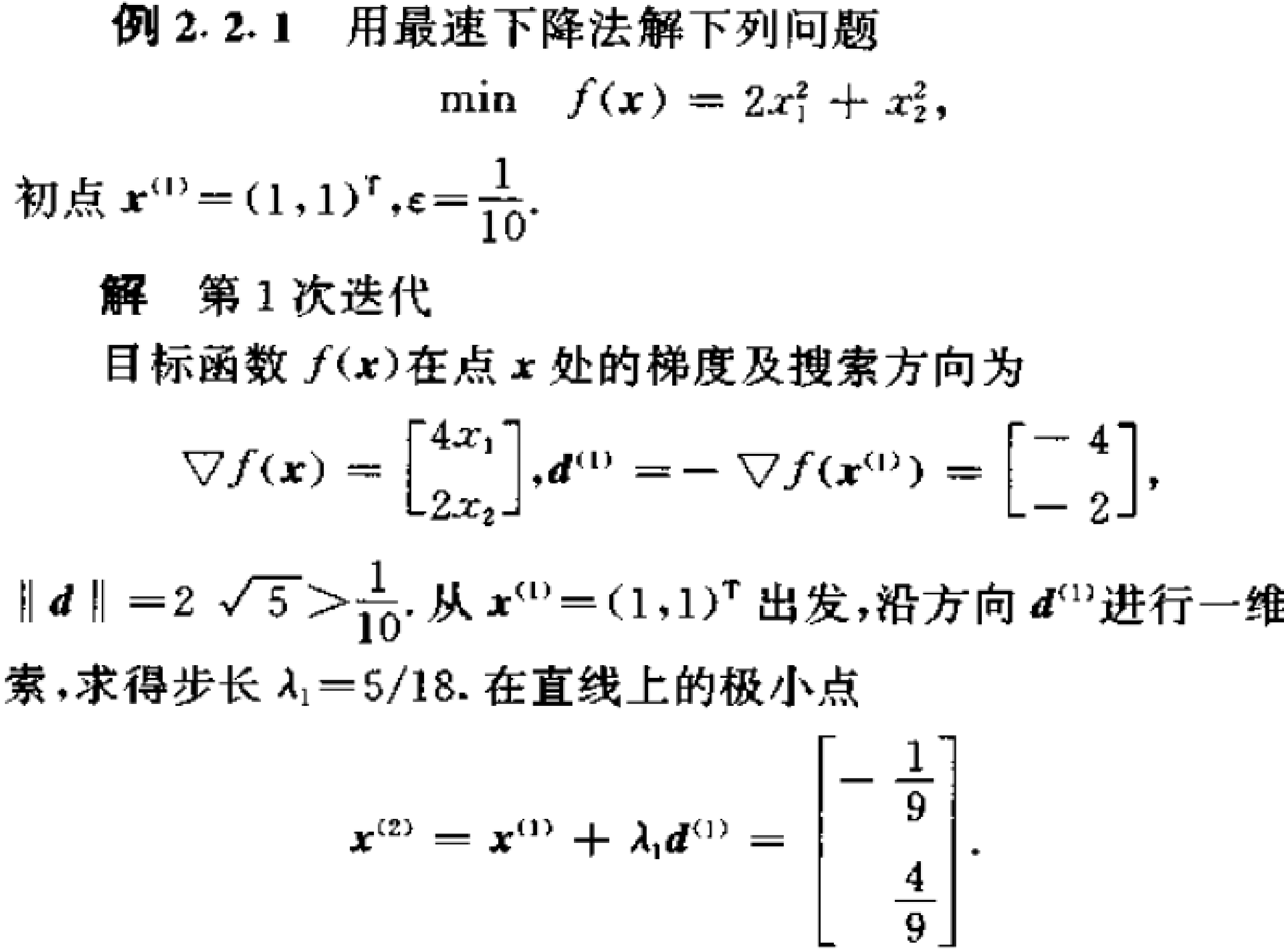
Steepest Gradient descent [最速下降法]

$$\left.\begin{aligned} \boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)}, \\ \boldsymbol{d}^{(k)} &= -\nabla f(\boldsymbol{x}^{(k)}), \\ \lambda_k:\ f(\boldsymbol{x}^{(k)}+\lambda_k \boldsymbol{d}^{(k)}) &= \min_{\lambda\geqslant 0} f(\boldsymbol{x}^{(k)}+\lambda \boldsymbol{d}^{(k)}). \end{aligned}\right\} \quad (2.26)$$

- Pseudo code:

```
① Configure: ε>0, k=1, $x^{(1)}$ ← random [(0,0,…,0)]
② $\boldsymbol{d}^{(k)} = -\nabla f(\boldsymbol{x}^{(k)})$,
③ Stop if $\left\|d^{(k)}\right\| < \varepsilon$; else compute optimal $\lambda_k$ which is
   determined by following optimization problem
       $f(\boldsymbol{x}^{(k)}+\lambda_k \boldsymbol{d}^{(k)}) = \min_{\lambda\geqslant 0} f(\boldsymbol{x}^{(k)}+\lambda \boldsymbol{d}^{(k)})$.
④ Set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda_k \boldsymbol{d}^{(k)}$ and k→k+1, goto ②
```

□ Example of Steepest Descent method



例 2.2.1 用最速下降法解下列问题

$$\min \quad f(\boldsymbol{x})=2x_1^2+x_2^2,$$

初点 $\boldsymbol{x}^{(1)}=(1,1)^{\mathrm{T}},\epsilon=\frac{1}{10}$.

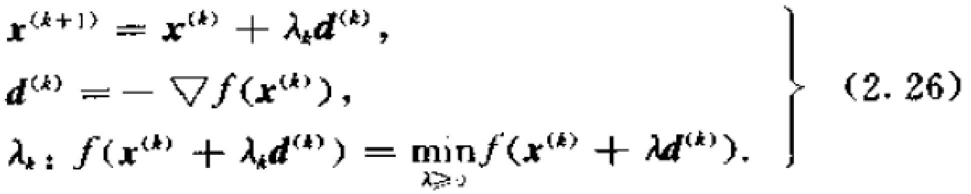
解 第1次迭代

目标函数 $f(\boldsymbol{x})$在点 $\boldsymbol{x}$ 处的梯度及搜索方向为

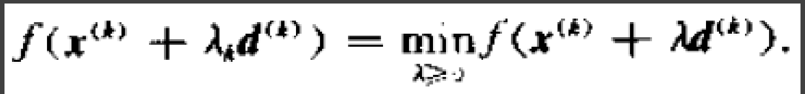
$$\nabla f(\boldsymbol{x})=\begin{bmatrix}4x_1\\2x_2\end{bmatrix},\boldsymbol{d}^{(1)}=-\nabla f(\boldsymbol{x}^{(1)})=\begin{bmatrix}-4\\-2\end{bmatrix},$$

$\|\boldsymbol{d}\|=2\sqrt{5}>\frac{1}{10}$. 从 $\boldsymbol{x}^{(1)}=(1,1)^{\mathrm{T}}$ 出发,沿方向 $\boldsymbol{d}^{(1)}$进行一维搜索,求得步长 $\lambda_1=5/18$. 在直线上的极小点

$$\boldsymbol{x}^{(2)}=\boldsymbol{x}^{(1)}+\lambda_1\boldsymbol{d}^{(1)}=\begin{bmatrix}-\frac{1}{9}\\ \frac{4}{9}\end{bmatrix}.$$



$$\left.\begin{aligned}&\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\lambda_k\boldsymbol{d}^{(k)},\\&\boldsymbol{d}^{(k)}=-\nabla f(\boldsymbol{x}^{(k)}),\\&\lambda_k:\ f(\boldsymbol{x}^{(k)}+\lambda_k\boldsymbol{d}^{(k)})=\min_{\lambda\geqslant 0}f(\boldsymbol{x}^{(k)}+\lambda\boldsymbol{d}^{(k)}).\end{aligned}\right\}\quad(2.26)$$



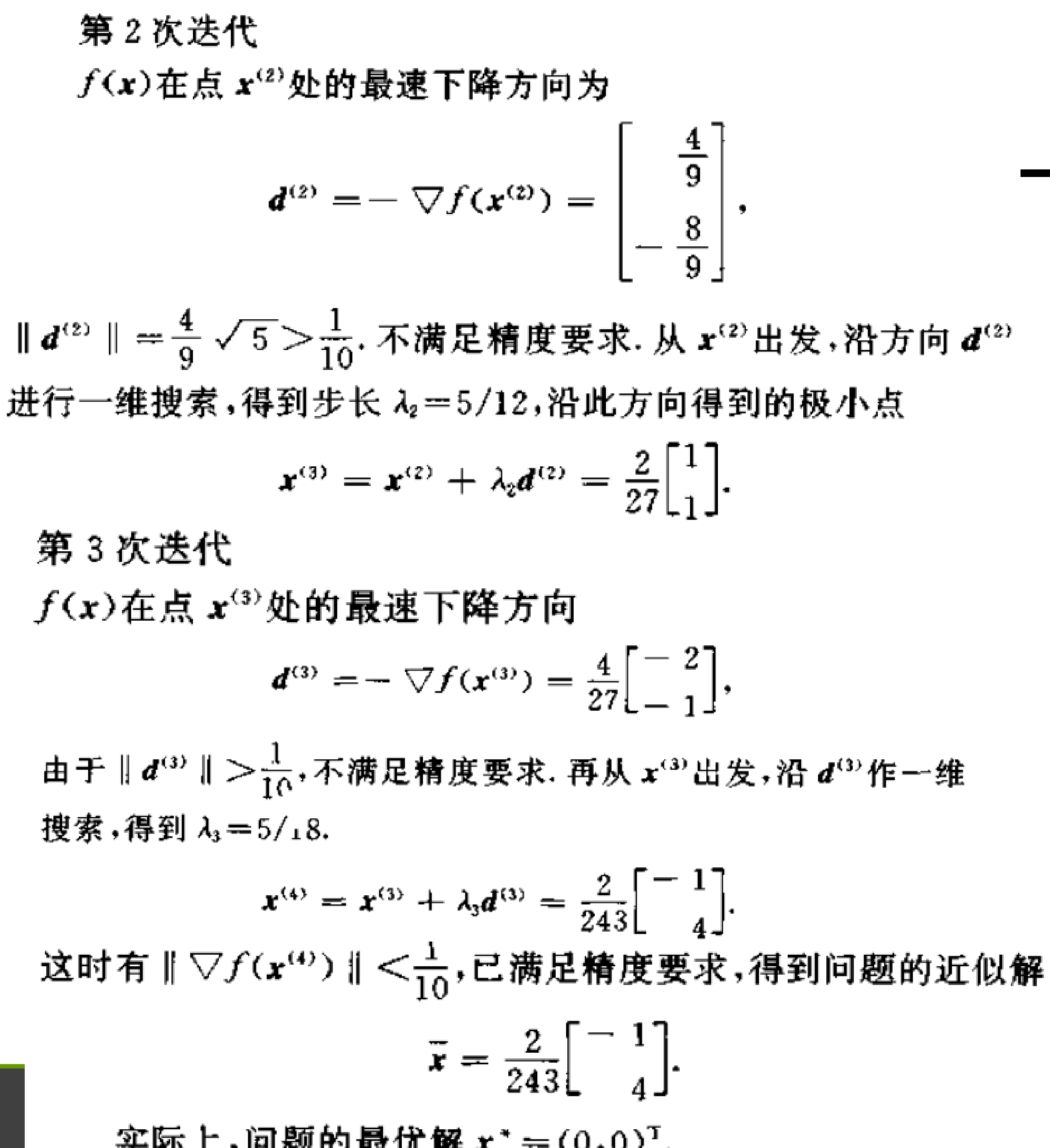
$$f(\boldsymbol{x}^{(k)}+\lambda_k\boldsymbol{d}^{(k)})=\min_{\lambda\geqslant 0}f(\boldsymbol{x}^{(k)}+\lambda\boldsymbol{d}^{(k)}).$$

- **λ$_1$=5/18 是如何求解的？**
  - $x^{(1)} = \begin{bmatrix}1\\1\end{bmatrix},\ \ d^{(1)} = \begin{bmatrix}-4\\-2\end{bmatrix}$
- **Then λ$_1$ is determined by following MIN**
  - $\min\limits_{\lambda\geq 0} f\left(x^{(1)} + \lambda d^{(1)}\right) = \min\limits_{\lambda\geq 0} f\left(\begin{bmatrix}1\\1\end{bmatrix} + \lambda\begin{bmatrix}-4\\-2\end{bmatrix}\right) = \min\limits_{\lambda\geq 0} f\left(\begin{bmatrix}1-4\lambda\\1-2\lambda\end{bmatrix}\right) = \min\limits_{\lambda\geq 0} 2(1-4\lambda)^2 + (1-2\lambda)^2$
- **It is again a MIN optimization problem.**
  - The object function is $f(\lambda) = 2(1-4\lambda)^2 + (1-2\lambda)^2$. We can use derivative computation again $\frac{\partial f(\lambda)}{\partial\lambda} = 0$

$$2*2*(1-4\lambda)*(-4) + 2*(1-2\lambda)*(-2) = 0$$

$$4(1-4\lambda) + (1-2\lambda) = 0$$

$$\lambda = \frac{5}{18}$$



第 2 次迭代

$f(\boldsymbol{x})$在点 $\boldsymbol{x}^{(2)}$处的最速下降方向为

$$\boldsymbol{d}^{(2)} = -\nabla f(\boldsymbol{x}^{(2)}) = \begin{bmatrix} \frac{4}{9} \\ -\frac{8}{9} \end{bmatrix},$$

$\| \boldsymbol{d}^{(2)} \| = \frac{4}{9}\sqrt{5} > \frac{1}{10}$. 不满足精度要求. 从 $\boldsymbol{x}^{(2)}$出发，沿方向 $\boldsymbol{d}^{(2)}$进行一维搜索，得到步长 $\lambda_2 = 5/12$，沿此方向得到的极小点

$$\boldsymbol{x}^{(3)} = \boldsymbol{x}^{(2)} + \lambda_2 \boldsymbol{d}^{(2)} = \frac{2}{27}\begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

第 3 次迭代

$f(\boldsymbol{x})$在点 $\boldsymbol{x}^{(3)}$处的最速下降方向

$$\boldsymbol{d}^{(3)} = -\nabla f(\boldsymbol{x}^{(3)}) = \frac{4}{27}\begin{bmatrix} -2 \\ -1 \end{bmatrix},$$

由于 $\| \boldsymbol{d}^{(3)} \| > \frac{1}{10}$，不满足精度要求. 再从 $\boldsymbol{x}^{(3)}$出发，沿 $\boldsymbol{d}^{(3)}$作一维搜索，得到 $\lambda_3 = 5/18$.

$$\boldsymbol{x}^{(4)} = \boldsymbol{x}^{(3)} + \lambda_3 \boldsymbol{d}^{(3)} = \frac{2}{243}\begin{bmatrix} -1 \\ 4 \end{bmatrix}.$$

这时有 $\| \nabla f(\boldsymbol{x}^{(4)}) \| < \frac{1}{10}$，已满足精度要求，得到问题的近似解

$$\bar{\boldsymbol{x}} = \frac{2}{243}\begin{bmatrix} -1 \\ 4 \end{bmatrix}.$$

实际上，问题的最优解 $\boldsymbol{x}^* = (0,0)^{\mathrm{T}}$.

Newton's Method

Newton's method for finding a zero can be derived from the Taylor's series expansion about the current iteration x_k,

$$f(x_{k+1}) = f(x_k) + (x_{k+1} - x_k)f'(x_k) + \mathcal{O}((x_{k+1} - x_k)^2)$$

Ignoring the terms higher than order two and assuming the function next iteration to be the root (i.e., $f(x_{k+1}) = 0$), we obtain,

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

This iterative procedure converges quadratically, so

$$\lim_{k\to\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^2} = \text{const.}$$

Batch vs stochastic optimization

Batch

$$W_i \leftarrow W_i - \eta \sum_{j=1}^{N} \frac{\partial l(x_j, y_j)}{\partial W_i}$$

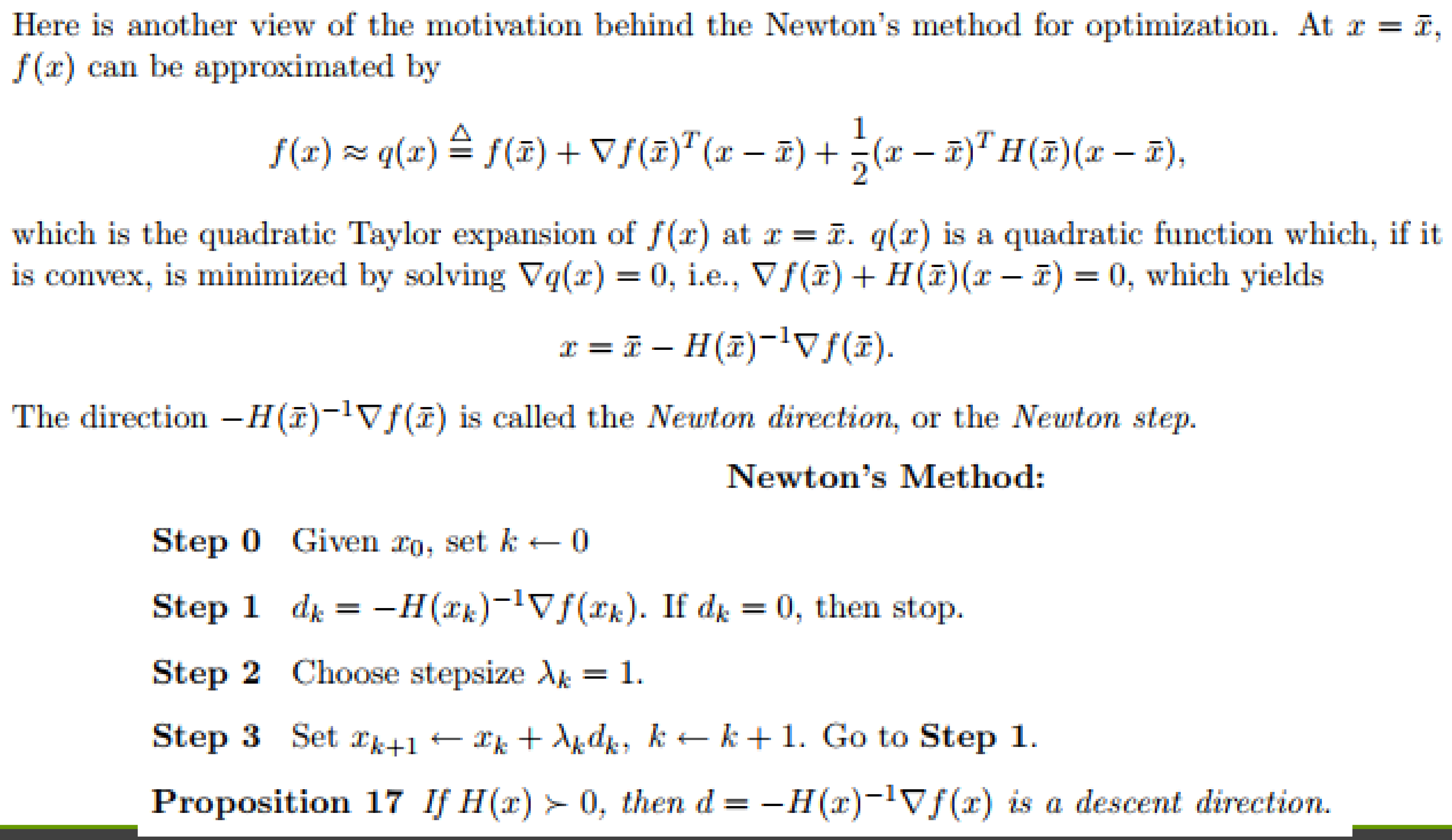
Online/Stochastic

$$W_i \leftarrow W_i - \eta \frac{\partial l(x_j, y_j)}{\partial W_i}$$

Minibatch

$$W_i \leftarrow W_i - \eta \sum_{j=k}^{k+m} \frac{\partial l(x_j, y_j)}{\partial W_i}$$

Newton method (variation)



Here is another view of the motivation behind the Newton's method for optimization. At $x = \bar{x}$, $f(x)$ can be approximated by

$$f(x) \approx q(x) \triangleq f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T H(\bar{x})(x - \bar{x}),$$

which is the quadratic Taylor expansion of $f(x)$ at $x = \bar{x}$. $q(x)$ is a quadratic function which, if it is convex, is minimized by solving $\nabla q(x) = 0$, i.e., $\nabla f(\bar{x}) + H(\bar{x})(x - \bar{x}) = 0$, which yields

$$x = \bar{x} - H(\bar{x})^{-1} \nabla f(\bar{x}).$$

The direction $-H(\bar{x})^{-1} \nabla f(\bar{x})$ is called the *Newton direction*, or the *Newton step*.

Newton's Method:

Step 0 Given x_0, set $k \leftarrow 0$

Step 1 $d_k = -H(x_k)^{-1} \nabla f(x_k)$. If $d_k = 0$, then stop.

Step 2 Choose stepsize $\lambda_k = 1$.

Step 3 Set $x_{k+1} \leftarrow x_k + \lambda_k d_k$, $k \leftarrow k + 1$. Go to **Step 1**.

Proposition 17 *If $H(x) \succ 0$, then $d = -H(x)^{-1} \nabla f(x)$ is a descent direction.*

Example 2: $f(x) = -\ln(1 - x_1 - x_2) - \ln x_1 - \ln x_2$.

$$\nabla f(x) = \begin{bmatrix} \frac{1}{1-x_1-x_2} - \frac{1}{x_1} \\ \frac{1}{1-x_1-x_2} - \frac{1}{x_2} \end{bmatrix},$$

$$H(x) = \begin{bmatrix} \left(\frac{1}{1-x_1-x_2}\right)^2 + \left(\frac{1}{x_1}\right)^2 & \left(\frac{1}{1-x_1-x_2}\right)^2 \\ \left(\frac{1}{1-x_1-x_2}\right)^2 & \left(\frac{1}{1-x_1-x_2}\right)^2 + \left(\frac{1}{x_2}\right)^2 \end{bmatrix}.$$

$x^\star = \left(\frac{1}{3}, \frac{1}{3}\right)$, $f(x^\star) = 3.295836866$.

k	$(x_k)_1$	$(x_k)_2$	$\|x_k - \bar{x}\|$
0	0.85	0.05	0.58925565098879
1	0.717006802721088	0.0965986394557823	0.450831061926011
2	0.512975199133209	0.176479706723556	0.238483249157462
3	0.352478577567272	0.273248784105084	0.0630610294297446
4	0.338449016006352	0.32623807005996	0.00874716926379655
5	0.333337722134802	0.333259330511655	$7.41328482837195e^{-5}$
6	0.333333343617612	0.33333332724128	$1.19532211855443e^{-8}$
7	0.333333333333333	0.333333333333333	$1.57009245868378e^{-16}$

Many other algorithms

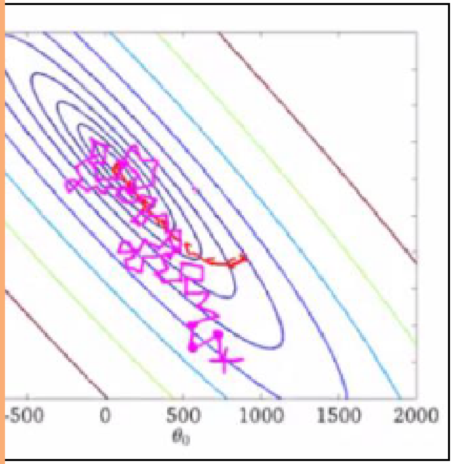
- **Conjugate Gradient Method**
- **Modified Newton's Method**
- **Quasi-Newton Methods (拟牛顿)**
  - …
  - Davidon-Fletcher-Powell (**DFP**) Method
  - Broyden-Fletcher-Goldfarb-Shanno (**BFGS**) Method
    - The DFP update was soon superseded by the BFGS formula, which is generally considered to be the most effective quasi-Newton update.

Broyden, Fletcher, Goldfarb and Shanno at the NATO Optimization Meeting (Cambridge, UK, 1983), a seminal meeting for continuous optimization

Stochastic Gradient Descent (SGD)

Algorithm 2 Stochastic Gradient Descent (SGD)

```
procedure SGD($\mathcal{D}, \boldsymbol{\theta}^{(0)}$)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(0)}$
    while not converged do
        for $i \in$ shuffle($\{1, 2, \ldots, N\}$) do
            $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \nabla_{\boldsymbol{\theta}} J^{(i)}(\boldsymbol{\theta})$
    return $\boldsymbol{\theta}$
```



We need a per-example objective:

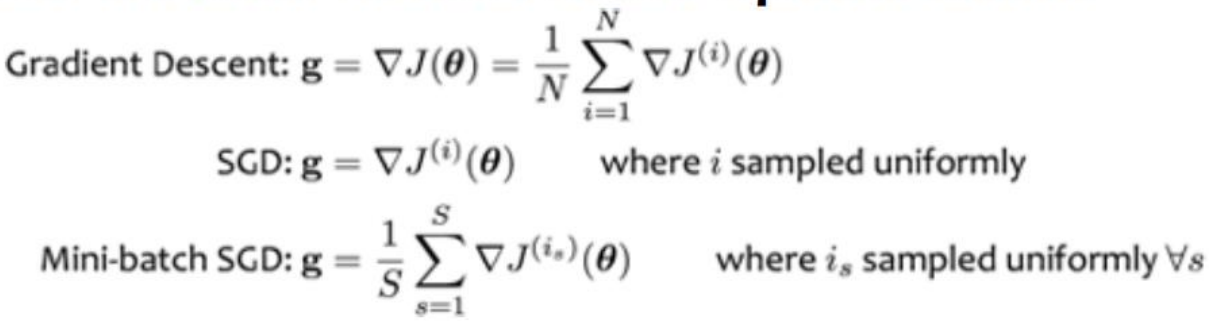
$$\text{Let } J(\boldsymbol{\theta}) = \sum_{i=1}^{N} J^{(i)}(\boldsymbol{\theta})$$

In practice, it is common to implement SGD using sampling **without** replacement (i.e. shuffle({1,2,… N}), even though most of the theory is for sampling **with** replacement (i.e. Uniform({1,2,… N}).

- **Gradient Descent:**
  Compute true gradient exactly from all N examples
- **Stochastic Gradient Descent (SGD):**
  Approximate true gradient by the gradient of one randomly chosen example
- **Mini-Batch SGD:**
  Approximate true gradient by the average gradient of K randomly chosen examples

while not converged: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \lambda \mathbf{g}$

Three variants of first-order optimization:



Gradient Descent: $\mathbf{g} = \nabla J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \nabla J^{(i)}(\boldsymbol{\theta})$

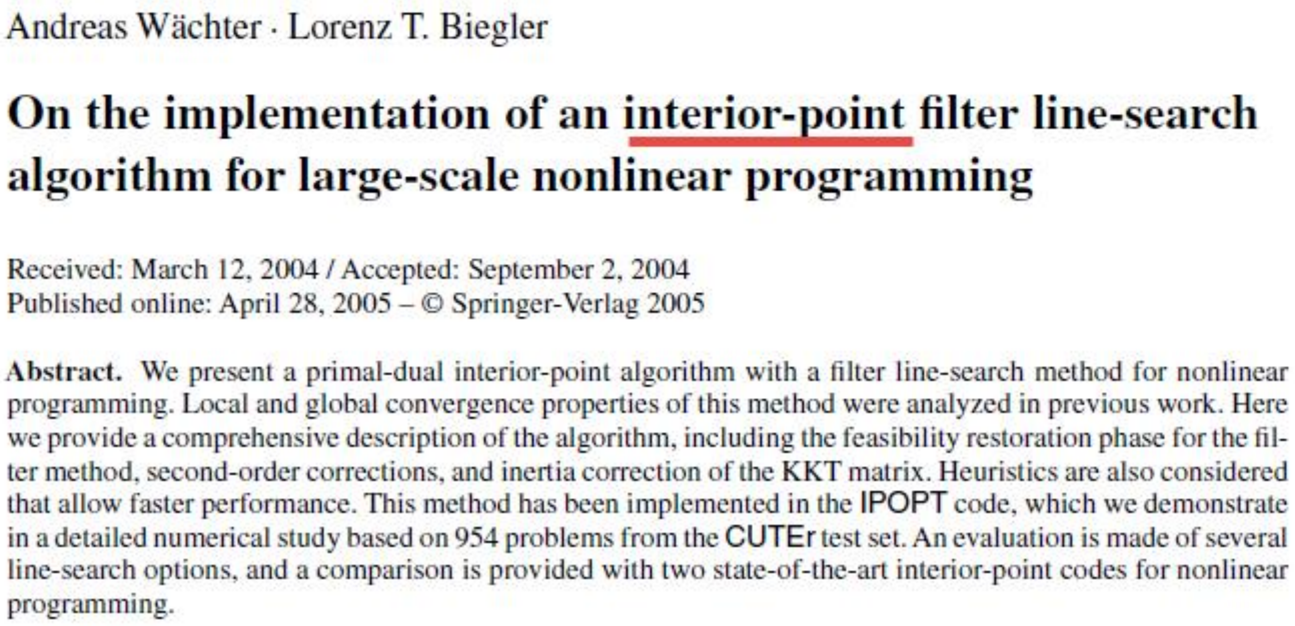
SGD: $\mathbf{g} = \nabla J^{(i)}(\boldsymbol{\theta})$ where i sampled uniformly

Mini-batch SGD: $\mathbf{g} = \frac{1}{S}\sum_{s=1}^{S} \nabla J^{(i_s)}(\boldsymbol{\theta})$ where i_s sampled uniformly $\forall s$

Recently…

- IPOPT

Math. Program., Ser. A 106, 25–57 (2006)



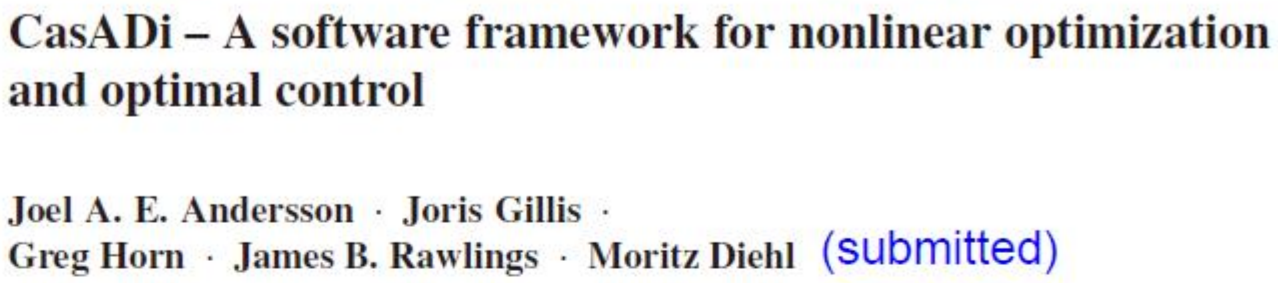
Andreas Wächter · Lorenz T. Biegler

On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming

Received: March 12, 2004 / Accepted: September 2, 2004
Published online: April 28, 2005 – © Springer-Verlag 2005

Abstract. We present a primal-dual interior-point algorithm with a filter line-search method for nonlinear programming. Local and global convergence properties of this method were analyzed in previous work. Here we provide a comprehensive description of the algorithm, including the feasibility restoration phase for the filter method, second-order corrections, and inertia correction of the KKT matrix. Heuristics are also considered that allow faster performance. This method has been implemented in the IPOPT code, which we demonstrate in a detailed numerical study based on 954 problems from the CUTEr test set. An evaluation is made of several line-search options, and a comparison is provided with two state-of-the-art interior-point codes for nonlinear programming.

CasADi

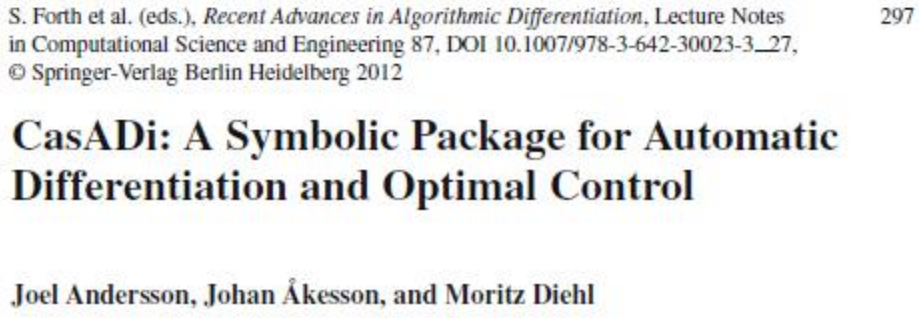


CasADi – A software framework for nonlinear optimization and optimal control

Joel A. E. Andersson · Joris Gillis ·
Greg Horn · James B. Rawlings · Moritz Diehl

(submitted)

existing reference:



S. Forth et al. (eds.), *Recent Advances in Algorithmic Differentiation*, Lecture Notes in Computational Science and Engineering 87, DOI 10.1007/978-3-642-30023-3_27, © Springer-Verlag Berlin Heidelberg 2012 297

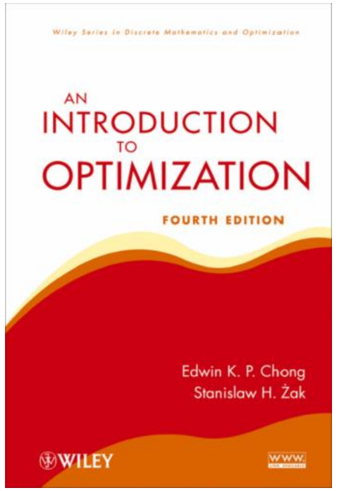
CasADi: A Symbolic Package for Automatic Differentiation and Optimal Control

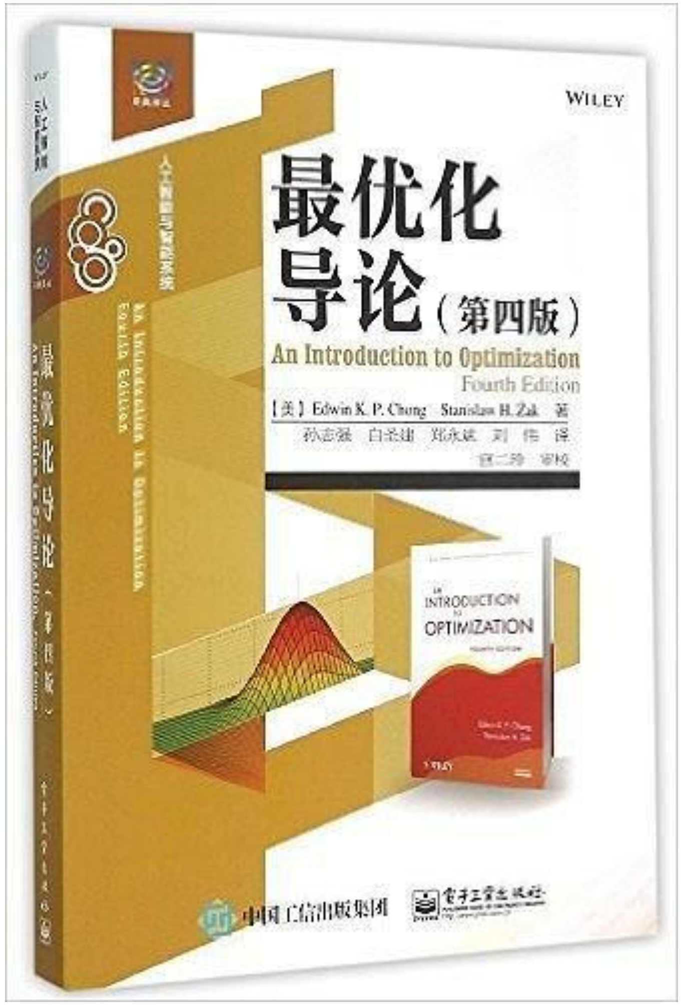
Joel Andersson, Johan Åkesson, and Moritz Diehl

- Inequality constraints can be addressed by Interior Point (IP) methods, e.g. in IPOPT code

- Derivatives of problem functions can be automatically provided e.g. by CasADi optimization environment

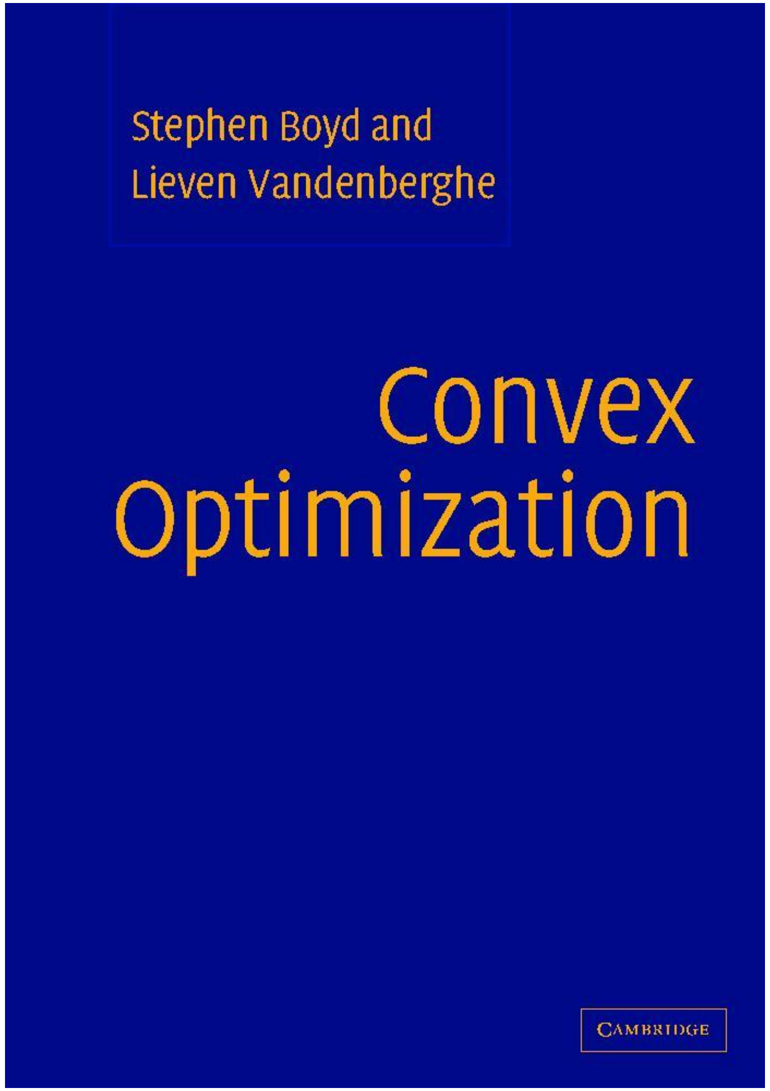
You can try ☺ But,
I should admit I
did not ☺


Wiley Series in Discrete Mathematics and Optimization
AN INTRODUCTION TO OPTIMIZATION
FOURTH EDITION
Edwin K. P. Chong
Stanislaw H. Żak
WILEY


WILEY
最优化导论（第四版）
An Introduction to Optimization
Fourth Edition
【美】Edwin K. P. Chong　Stanislaw H. Żak　著
孙志强　白圣建　郑永斌　刘伟　译
人工智能与智能系统
中国工信出版集团
电子工业出版社

- □ 原作名: An Introduction to Optimization,Foulth Edition
- □ 中文名：最优化导论
- □ 作者: Edwin K. P. Chong / Stanislaw H. Zak
- □ 出版社: 电子工业出版社
- □ 译者: 孙志强 / 白圣建 / 郑永斌 / 刘伟
- □ 出版年: 2015-10
- □ 定价: 89.00
- □ 丛书: 经典译丛·人工智能与智能系统
- □ ISBN: 9787121267154

https://stanford.edu/~boyd/cvxbook/



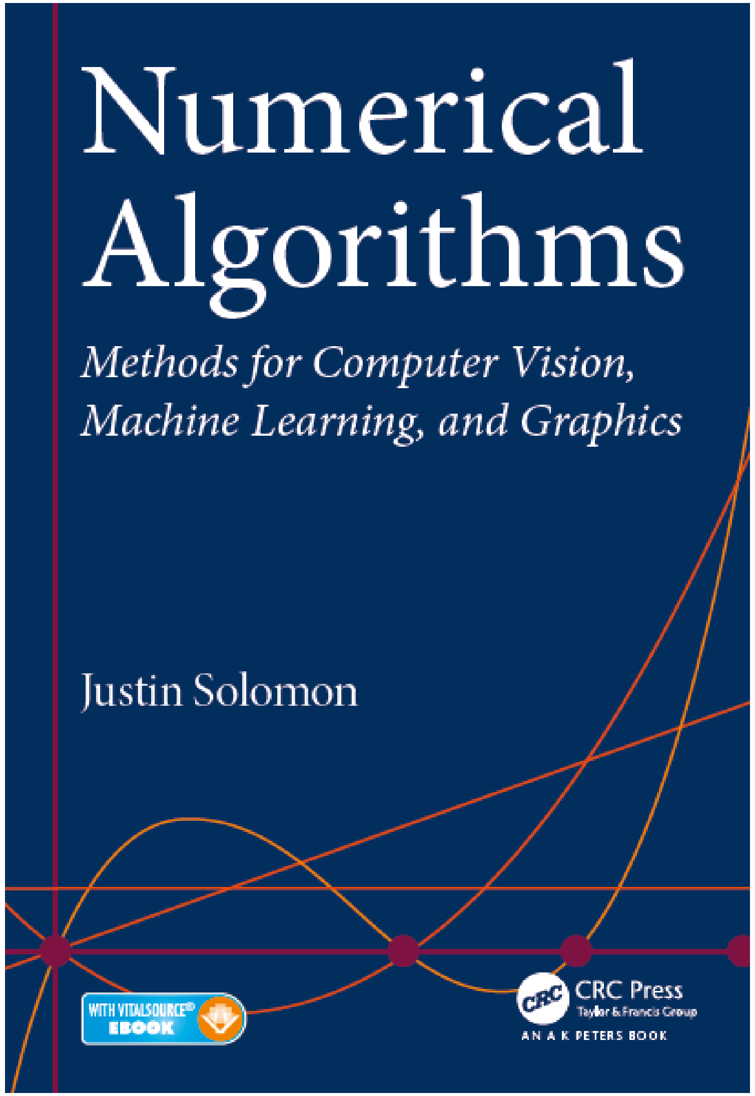
Stephen Boyd and
Lieven Vandenberghe

Convex
Optimization

CAMBRIDGE

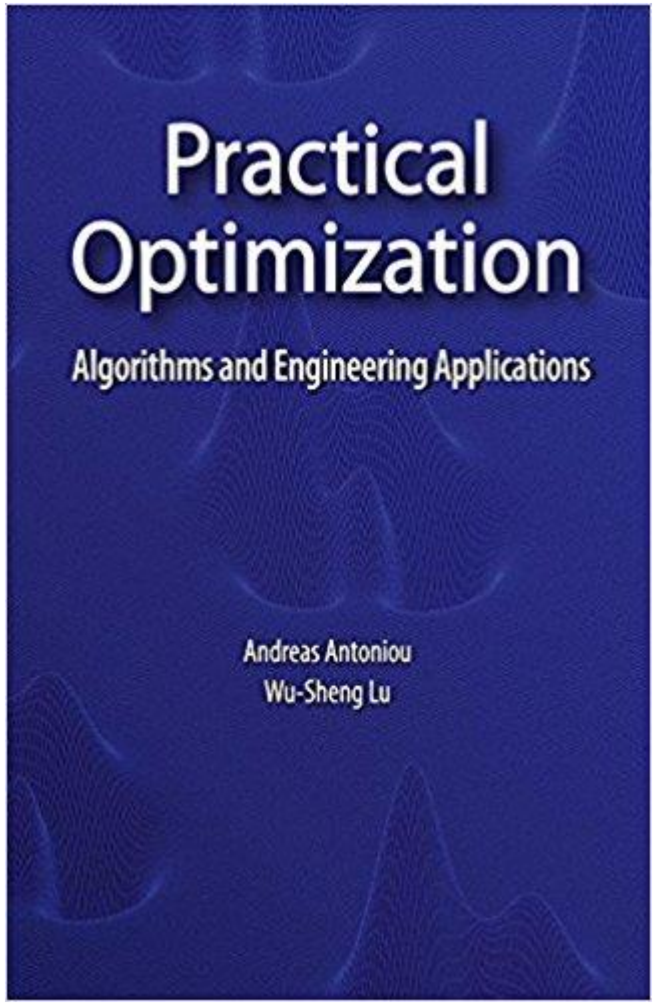
- *Convex Optimization*
  Stephen Boyd and Lieven Vandenberghe

  Cambridge University Press



- □ **Numerical Algorithms**
- □ **Methods for Computer Vision, Machine Learning, and Graphics**
- □ **Justin Solomon**

- □ **D:\My7\MyProjects\Books\01 商务智能-大企业生存之道\课程设计的资料\05 C 优化论**
  - Numerical-Algorithms-Methods-for-Computer-Vision-Machine-Learning-and-Graphics.pdf



Practical Optimization

Algorithms and Engineering Applications

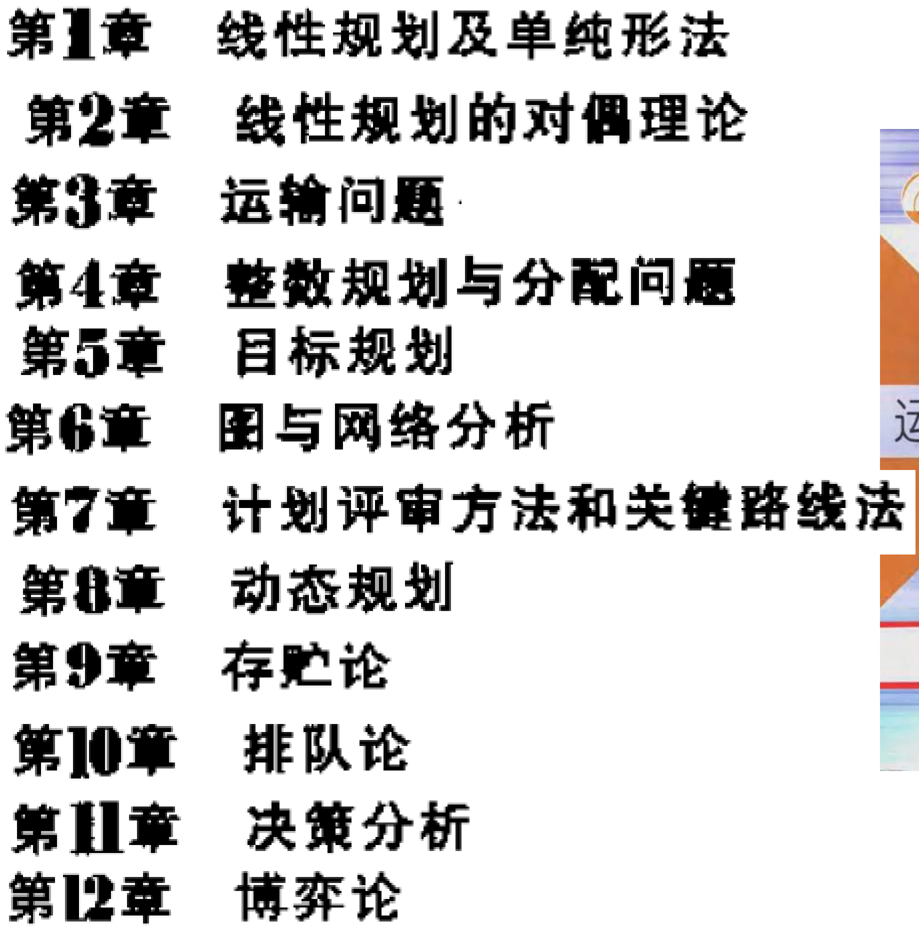
Andreas Antoniou

Wu-Sheng Lu

- □ **Practical Optimization: Algorithms and Engineering Applications 2007th Edition**
- □ **by Andreas Antoniou, Wu-Sheng Lu**

http://www.ece.uvic.ca/~andreas/Books.html

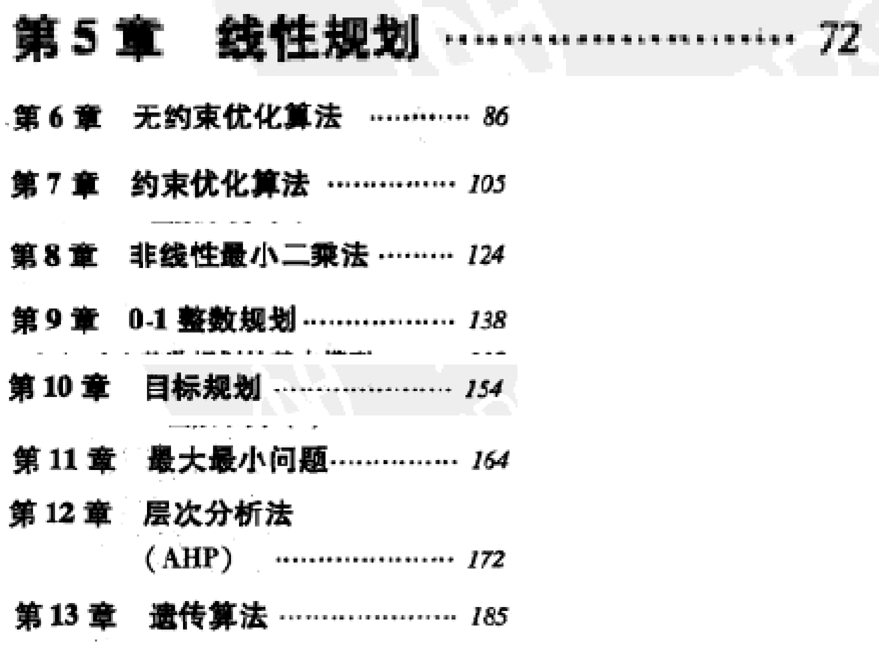
- **Operational Research 本身有很多内容：**
  - 上面的运筹和优化，偏优化。实际运筹常见如下内容 – 在简单介绍优化后，按照应用来的
  - 那些规划：线性，非线性，整数，目标, 动态
  - 启发式优化：模拟退火，遗传, particle, 。。。
  - 存储问题，网络流，。。。

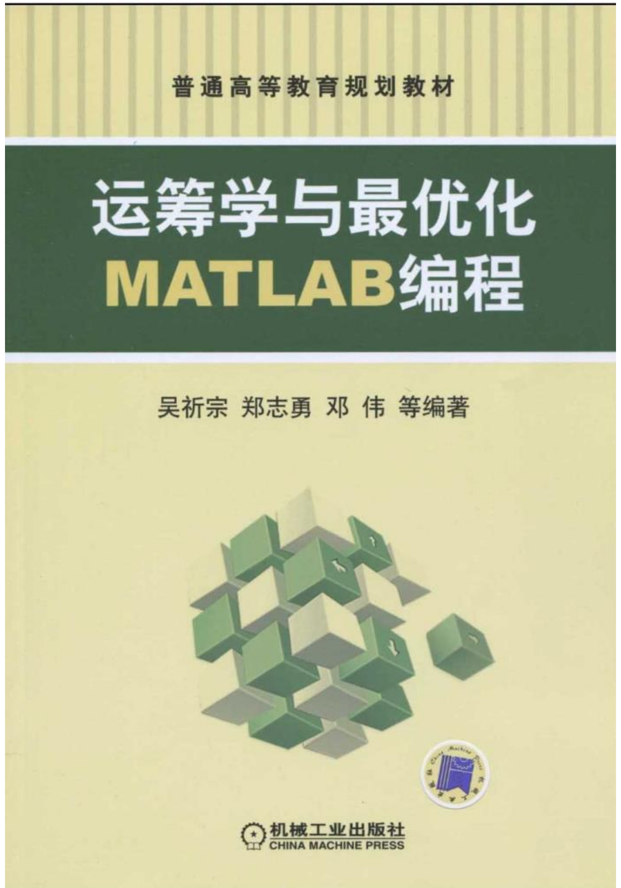


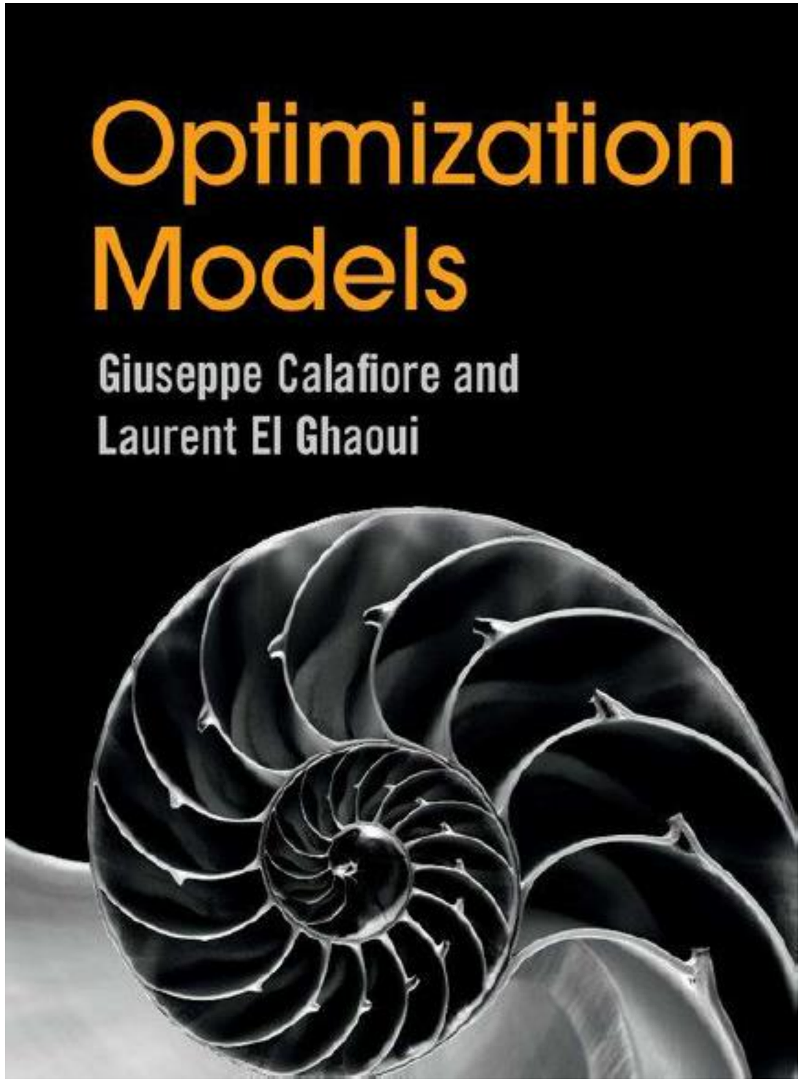
第1章　线性规划及单纯形法
第2章　线性规划的对偶理论
第3章　运输问题
第4章　整数规划与分配问题
第5章　目标规划
第6章　图与网络分析
第7章　计划评审方法和关键路线法
第8章　动态规划
第9章　存贮论
第10章　排队论
第11章　决策分析
第12章　博弈论

Operational Research 本身有很多内容:

- ,







- **Optimization Models**
- **Giuseppe Calafiore, Laurent El Ghaoui**

- **D:\My7\MyProjects\Books\01 商务智能-大企业生存之道\课程设计的资料\05 C 优化论**
  - Optimization-models.djvu
  - Optimization-Models.pdf