Modelling Transportation Systems

Nonlinear Programming

Dr Zhiyuan Liu
Email: zhiyuanl@seu.edu.cn

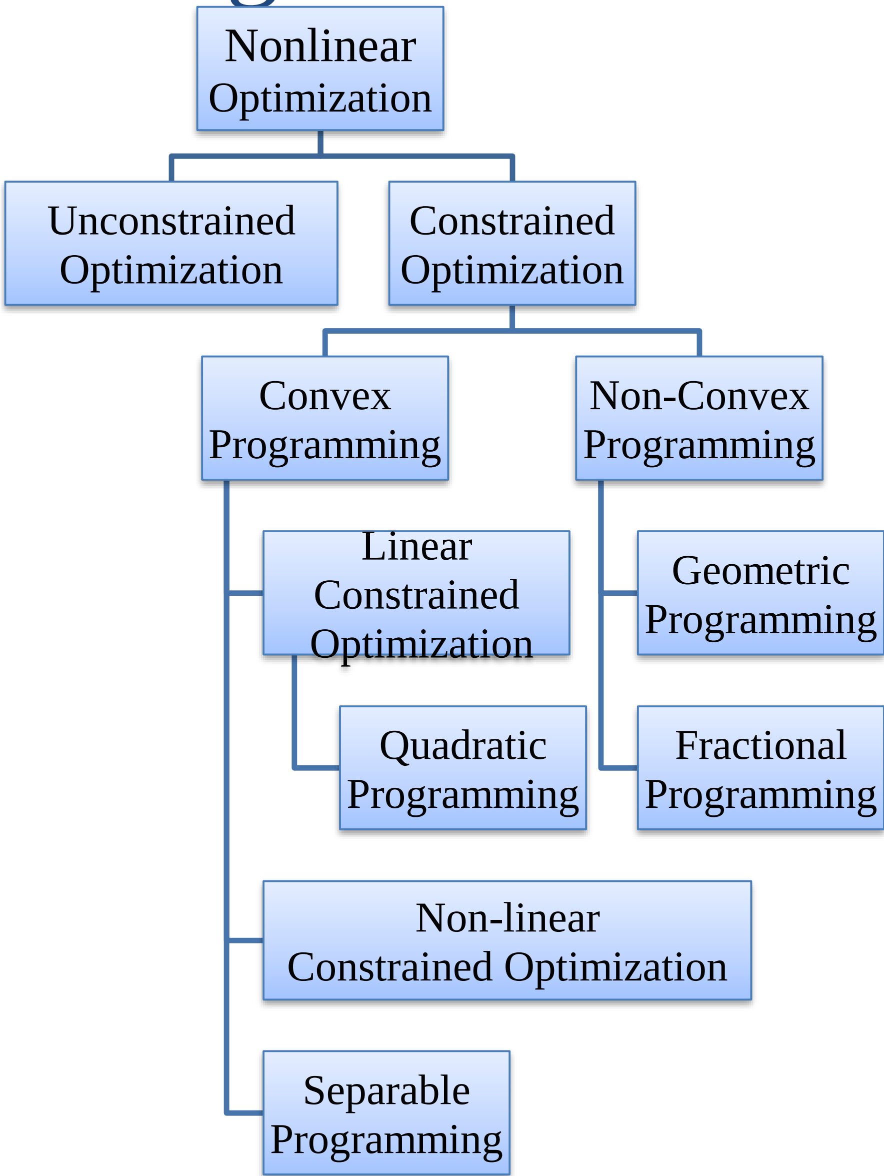
Nonlinear Programming (NLP)

❑ **Learning Objectives**

- Understand the formulation and classification of NLP.
- Familiar with the first- and second-order necessary optimality conditions.
- Learn the Karush-Kuhn-Tucker (KKT) conditions.
- Know how to use KKT conditions to find a local optimum.
- Understand the steepest descent direction method for multidimensional unconstrained minimization problems.
- Know the Frank-Wolfe method for multidimensional constrained minimization problems and **traffic assignment problems**.

Mathematical Programming

- Linear Programming
- Nonlinear Programming
- Dynamic Programming
- Graphic Theory
- Stochastic Programming
- ……



Formulating a NLP

❑A Regression Example

Consumers' behavior prediction model

Suppose in a given month, a consumer allots E dollars to spend on 3 different products (i=1,2,3). Assume that the consumer will purchase a certain minimal amount q_i of product i with price p_i. Out of the remaining money a certain fixed fraction a_i of funds is allocated to purchase more of product i. Parameter a_i may be interpreted as the marginal propensity to consume commodity i out of uncommitted funds. Once these numbers can be estimated, we have the following model to predict the consumer's behavior:

$$p_i q_i + a_i[E - (p_1 q_1 + p_2 q_2 + p_3 q_3)] = p_i q_i + a_i(E - \sum_{j=1}^{3} p_j q_j),\ i = 1, 2, 3$$

Our problem is to infer parameters $\boldsymbol{q_i}$ and $\boldsymbol{a_i}$ (i=1,2,3) from her or his monthly purchase record.

Formulating a NLP (con't)

Purchase records over four months

Month m	E_m	Product 1		Product 2		Product 3	
		p_{m1}	Q_{m1}	p_{m2}	Q_{m2}	p_{m3}	Q_{m3}
1	1500	2.5	198	4.0	124	5.0	100
2	2000	2.7	200	3.5	200	4.9	110
3	1500	2.8	100	4.0	80	5.6	60
4	1200	2.6	152	3.6	115	4.5	65

Note: p_{mi} , price of product *i* at month *m*; Q_{mi} , amount of product *i* purchased at month *m*

Difference between the realistic and estimated expenses for product *i* at month *m* :

$$e_{mi} = p_{mi}Q_{mi} - \left[p_{mi}q_i + a_i(E_m - \sum_{j=1}^{3} p_{mj}q_j) \right], \; m = 1,2,3,4; i = 1,2,3$$

Formulating a NLP (con’t)

❑Nonlinear programming model

- Decision variables:

$a_i, q_i,\ i=1,2,3,$ namely, $a_1, a_2, a_3, q_1, q_2, q_3$

- Objective function:

Minimize sum of square of e_{mi} for all the products over the four months (Least Square Method)

$$\min_{a,q} \ f(a,q)=\sum_{m=1}^{4}\sum_{i=1}^{3}e_{mi}^2=\sum_{m=1}^{4}\sum_{i=1}^{3}\left\{p_{mi}Q_{mi}-\left[p_{mi}q_i+a_i(E_m-\sum_{j=1}^{3}p_{mj}q_j)\right]\right\}^2$$

where $a=(a_1,a_2,a_3)^T, q=(q_1,q_2,q_3)^T$

- Constraints:

(1) $a_i \geq 0;\ q_i \geq 0,\ i=1,2,3$

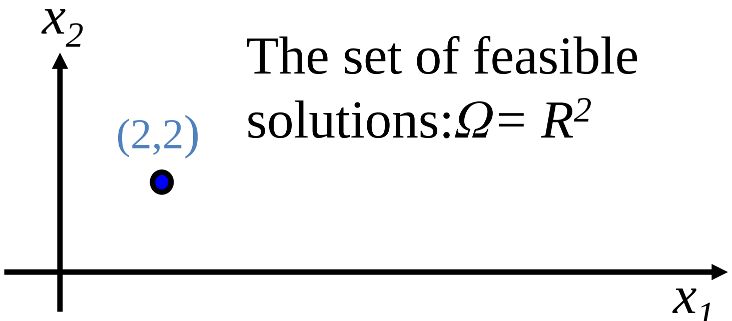
(2) $a_1+a_2+a_3=1$

(3) $q_i \leq Q_{mi},\ \ i=1,2,3,\ m=1,2,3,4$

Defining a Nonlinear Programming Problem

- **Example 1**

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

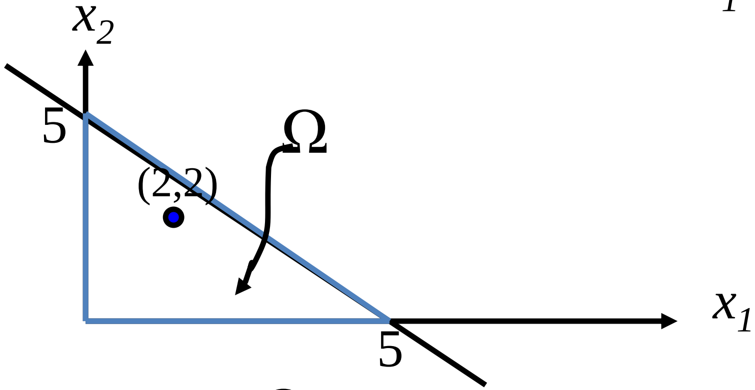


- **Example 2**

$$\min f(x) = (x_1 - 2)^2 + (x_2 - 2)^2$$

subject to

$$x_1 + x_2 - 5 \le 0$$
$$x_1, x_2 \ge 0$$

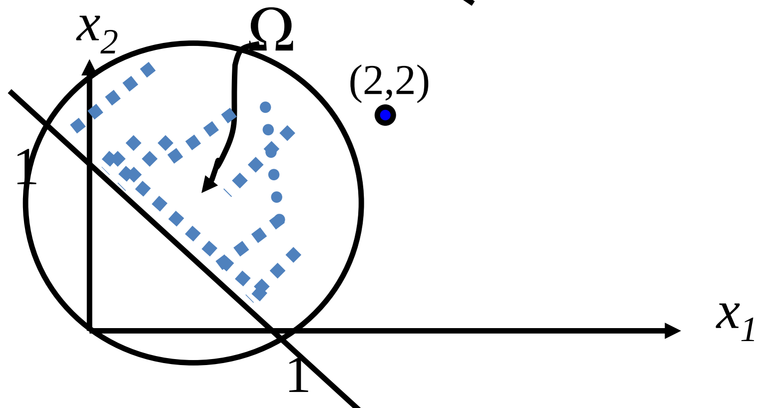


- **Example 2**

$$\min f(x) = (x_1 - 2)^2 + (x_2 - 2)^2$$

subject to

$$(x_1 - 1)^2 + (x_2 - 1)^2 - 1 \le 0$$
$$x_1 + x_2 \ge 1$$
$$x_1, x_2 \ge 0$$



Defining a Nonlinear Programming Problem (con't)

❑Problem statement

Determine a set of values of the decision variables that optimizes an objective function subject to several constraints described by appropriate functions

$$\min \ (\text{or max})\ f(x_1, \ldots, x_n)$$

Objective Function

subject to

$$g_1(x_1, \cdots, x_n) \le 0$$
$$g_2(x_1, \cdots, x_n) \le 0$$
$$\vdots$$
$$g_{m_1}(x_1, \cdots, x_n) \le 0$$

Inequality constraints

$$h_1(x_1, \cdots, x_n) = 0$$
$$h_2(x_1, \cdots, x_n) = 0$$
$$\vdots$$
$$h_{m_2}(x_1, \cdots, x_n) = 0$$

Equality constraints

Defining a Nonlinear Programming Problem (con't)

- **Vectors:**
  - Decision variables: $x = \left(x_1, \cdots, x_n\right)^T$
  - Inequality constraints: $g = \left(g_1, \cdots, g_{m_1}\right)^T$
  - Equality constraints: $h = \left(h_1, \cdots, h_{m_2}\right)^T$
- **Assumptions**

  f(*x*), *g*(*x*) and *h*(*x*) are continuously differentiable functions
- **Alternative representations** (use vectors)

$$\min_x f\left(x\right)$$

Subject to:

$$g\left(x\right) \leq 0$$

$$h\left(x\right) = 0$$

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

where

$$\Omega = \left\{x \middle| g\left(x\right) \leq 0, h\left(x\right) = 0\right\}$$

- Inequality constraints are more common to see

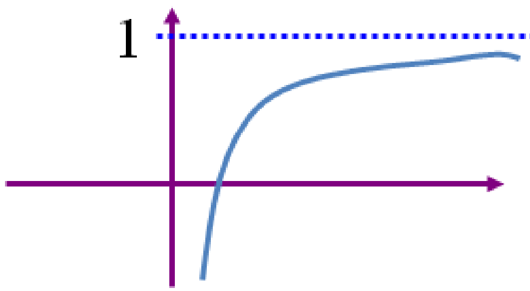
Preliminary knowledge

❑Difficulties with NLPs

1. Local optimum ≠ global optimum
2. Unlike LP, optimum may not occur at extreme point (vertex)
3. Even if *f(x)* is bounded, there may not be an optimal solution

Example

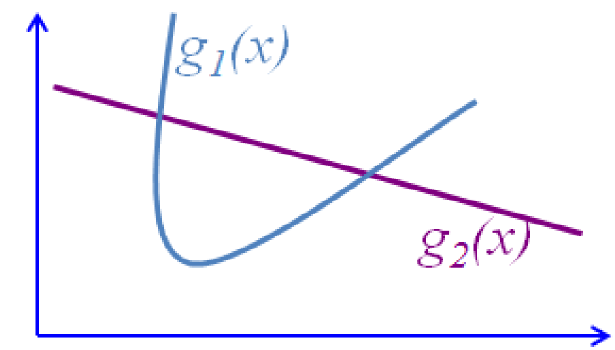
$$f(x)=1-\frac{1}{x},\Omega=\{x:x\geq 0.5\},\Rightarrow \max_{x\in\Omega} f(x)=?$$



4. Feasible solution region may not be connected

Example

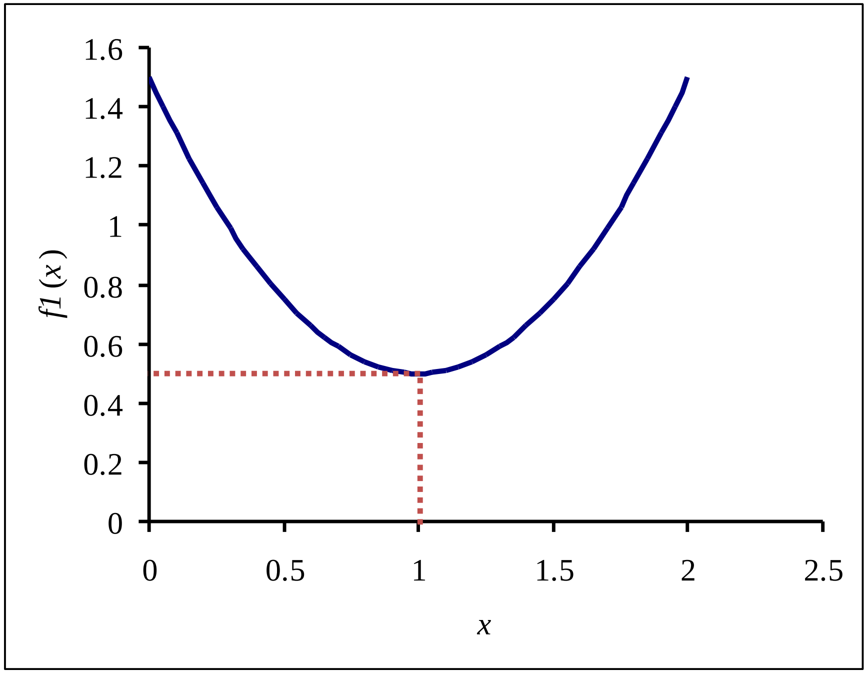
$$\Omega=\{x:g_1(x)\geq g_2(x)\}$$



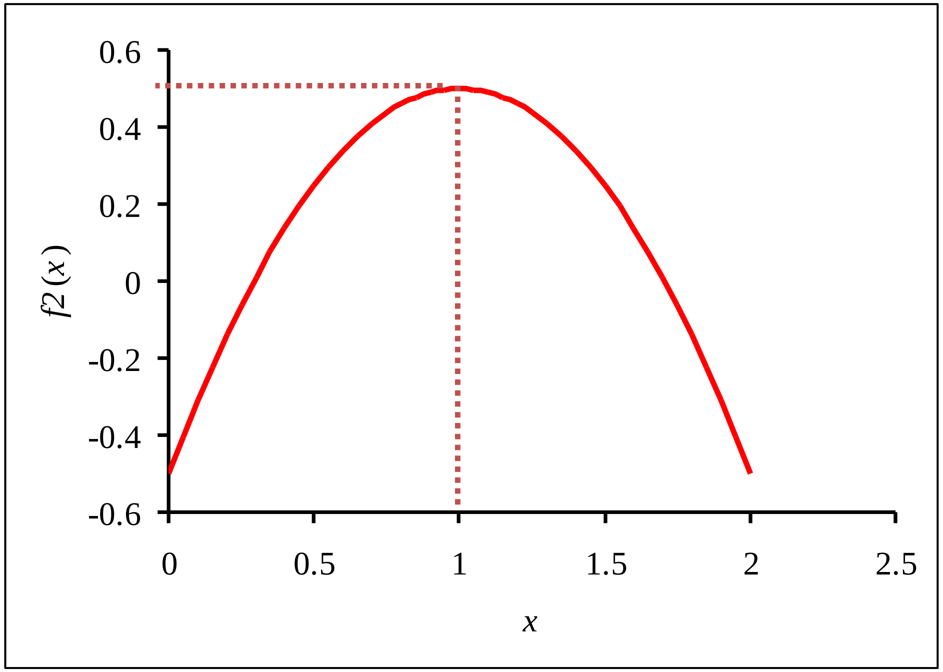
Preliminary knowledge (con't)

❑ Maximum and Minimum Solutions

- $x^*=1$ is the global minimum solution of function $f_1(x)$
- $x^*=1$ is the global maximum solution of function $f_2(x)$



(a) Function $f_1(x)=(x-1)^2+0.5$



(b) Function $f_2(x)=0.5-(x-1)^2$

Preliminary knowledge (con't)

- **Global and Local Minimum Solutions for Minimization Problem:** $\min\limits_{x \in \Omega} f(x)$

- **Global minimum solution *x****

(i) (Feasibility) x^* is a feasible solution of the minimization problem, i.e., $x^* \in \Omega$

(ii) (Optimality) $f(x) \geq f(x^*)$ for any $x \in \Omega$

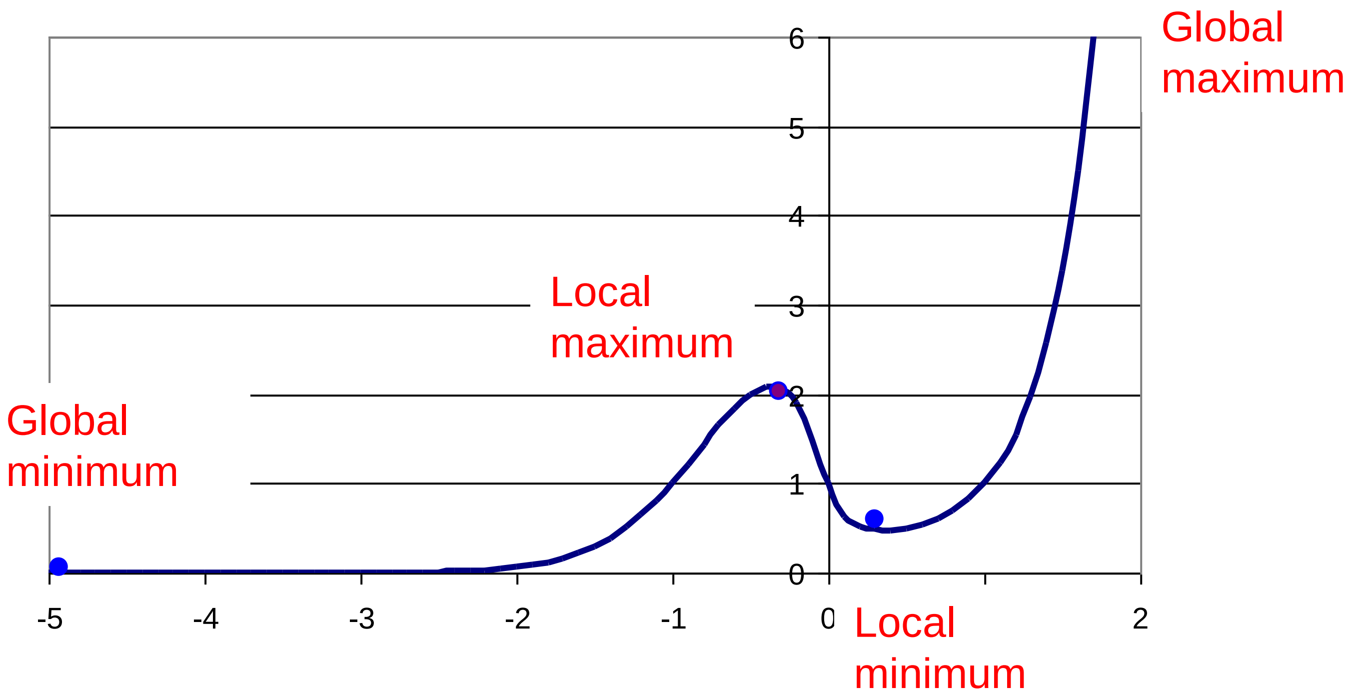
- **Local minimum solution *x****

(i) (Feasibility) x^* is a feasible solution of the minimization problem, i.e., $x^* \in \Omega$

(ii) (Optimality) $f(x) \geq f(x^*)$ for any x in a neighborhood of x^*

Preliminary knowledge (con't)

- What are the global minimum and global maximum solutions of function $f_3(x)$?
- What are the global minimum and maximum solutions of function $f_3(x)$ subject to $-5 \le x \le 2$?



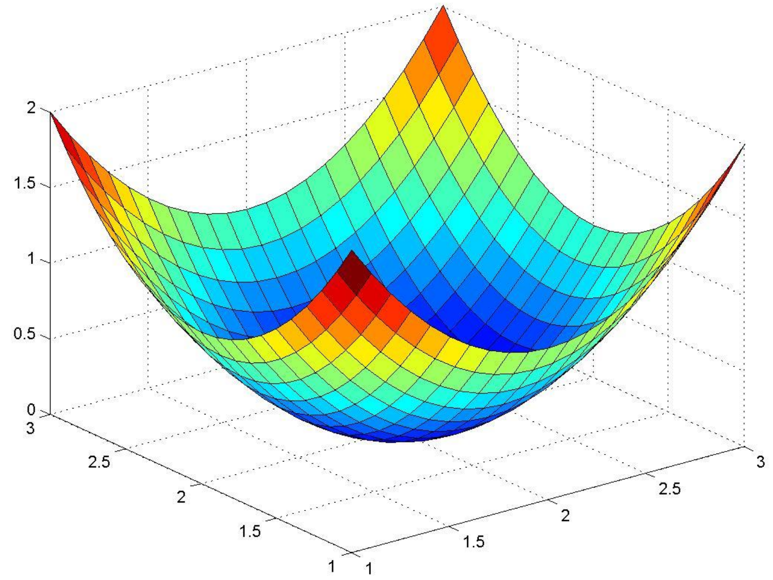
(c) Function $f_3(x) = \left(x^2\right)^x$

Multivariate Functions

- **Examples**

(1) $f_1(x) = (x_1 - 2)^2 + (x_2 - 2)^2$

where vector: $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \Re^2$



(2) $f_2(x) = 2x_1 + 3x_2 + 0.5x_3 - x_4$

where vector: $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \in \Re^4$

Multivariate Functions (con't)

- **Gradient of a n-dimensional function *f(x)* at a solution *x=x****

$$\nabla f\left(x^*\right) = \begin{pmatrix} \partial f\left(x^*\right)/\partial x_1 \\ \partial f\left(x^*\right)/\partial x_2 \\ \vdots \\ \partial f\left(x^*\right)/\partial x_n \end{pmatrix}$$

$f : R^n \to R^1$

what if $R^n \to R^n$?

Jacobian

- **Examples**

(1) $f_1\left(x\right) = \left(x_1 - 2\right)^2 + \left(x_2 - 2\right)^2$

$$\nabla f_1\left(x^*\right) = \begin{pmatrix} 2\left(x_1^* - 2\right) \\ 2\left(x_2^* - 2\right) \end{pmatrix}$$

(2) $f_2\left(x\right) = 2x_1 + 3x_2 + 0.5x_3 - x_4$

$$\nabla f_2\left(x^*\right) = \begin{pmatrix} 2 \\ 3 \\ 0.5 \\ -1 \end{pmatrix}$$

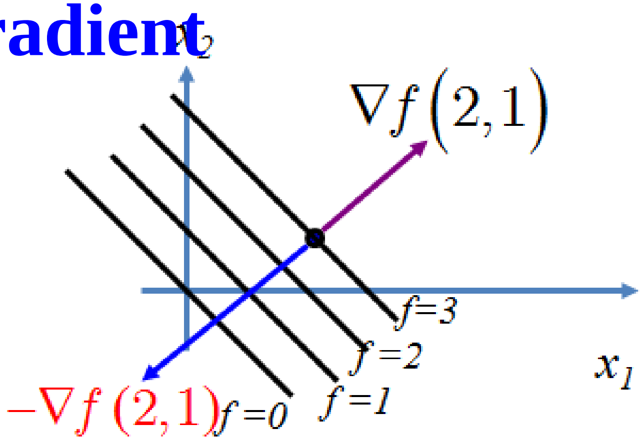
Multivariate Functions (con't)

❑Geometric interpretation of a gradient

Given a function $f(x)=x_1+x_2$,

at point $x=(2,1)$,

$$\nabla f(2,1) = (1,1)^T$$



✓ Conclusions

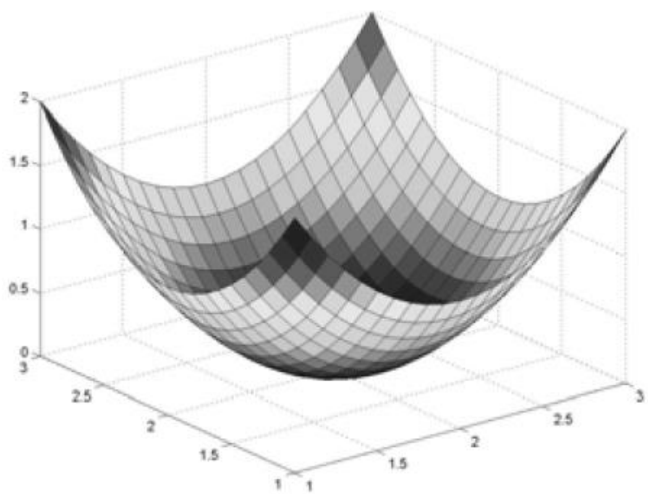
1. The function value will **increase** when moving away from point x along the gradient direction:

2. The function value will **decrease** when moving away from point x along the opposite direction:

The opposite direction of a gradient provides an important clue for minimizing a function.

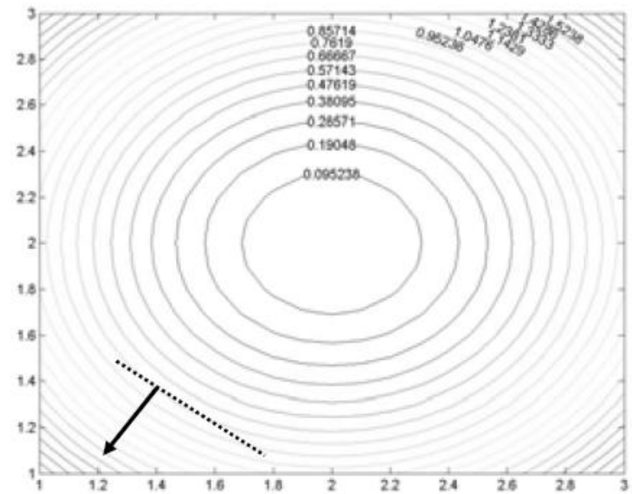
Multivariate Functions (con’t)

❑Geometric interpretation of a gradient



$$z = (x_1-2)^2+(x_2-2)^2$$

$$\nabla z(1.4,1.4) = (-1.2,-1.2)$$



Representation in contours

Multivariate Functions (con't)

- **Hessian Matrix of a n-dimensional Function *f(x)* at a Solution *x=x****

$$\nabla^2 f\left(x^*\right)=\begin{bmatrix} \partial^2 f\left(x^*\right)/\partial^2 x_1 & \partial^2 f\left(x^*\right)/\partial x_1 \partial x_2 & \ldots & \partial^2 f\left(x^*\right)/\partial x_1 \partial x_n \\ \partial^2 f\left(x^*\right)/\partial x_2 \partial x_1 & \partial^2 f\left(x^*\right)/\partial^2 x_2 & \ldots & \partial^2 f\left(x^*\right)/\partial x_2 \partial x_n \\ \vdots & \vdots & \vdots & \vdots \\ \partial^2 f\left(x^*\right)/\partial x_n \partial x_1 & \partial^2 f\left(x^*\right)/\partial x_n \partial x_2 & \ldots & \partial^2 f\left(x^*\right)/\partial^2 x_2 \end{bmatrix}_{n\times n}$$

- **Examples**

$$(1)\ f_1\left(x\right)=\left(x_1-2\right)^2+\left(x_2-2\right)^2$$

$$\nabla^2 f\left(x^*\right)=\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

$$(2)\ f_2\left(x\right)=2x_1+3x_2+0.5x_3-x_4$$

$$\nabla^2 f_2\left(x^*\right)=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Definition of Positive Definite

Let A be a $n \times n$ matrix

- Positive definite matrix

For any non-zero vector x, we have $x^T A x > 0$

- Positive semi-definite matrix

For any non-zero vector x, we have $x^T A x \geq 0$

- A sufficient and necessary condition

Matrix A is positive definite if the values of the leading principal minor determinants are positive

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \qquad \left| a_{11} \right| > 0 \qquad \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0 \qquad \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} > 0$$

Preliminary knowledge (con't)

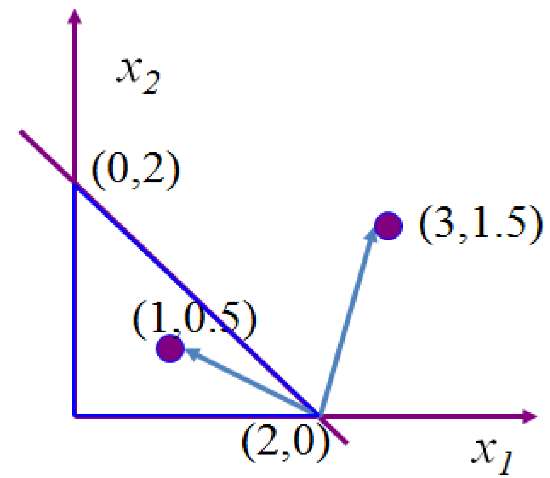
Feasible Directions of a NLP

$$\min_{x \in \Omega} f(x) \qquad \text{where } \Omega \subseteq R^n$$

- Feasible direction d at a point x of set Ω

A direction d (i.e., a n-dimensional vector) is referred to as a *feasible direction* at point x of set Ω iff there is a parameter such that

$$x + \alpha d \in \Omega \text{ for all } 0 \le \alpha \le \bar{\alpha}$$



Example: $\Omega=\{(x_1,x_2)|x_1+x_2\le 2,\ x_1\ge 0,\ x_2\ge 0\}$

✓ For interior point (1,0.5), any direction is a feasible direction *(Why?)*

✓ For vertex (2, 0),

Feasible direction: d = (1,0.5) - (2,0) = (-1,0.5)

Infeasible direction: d = (3, 1.5) - (2,0) = (1, 1.5)

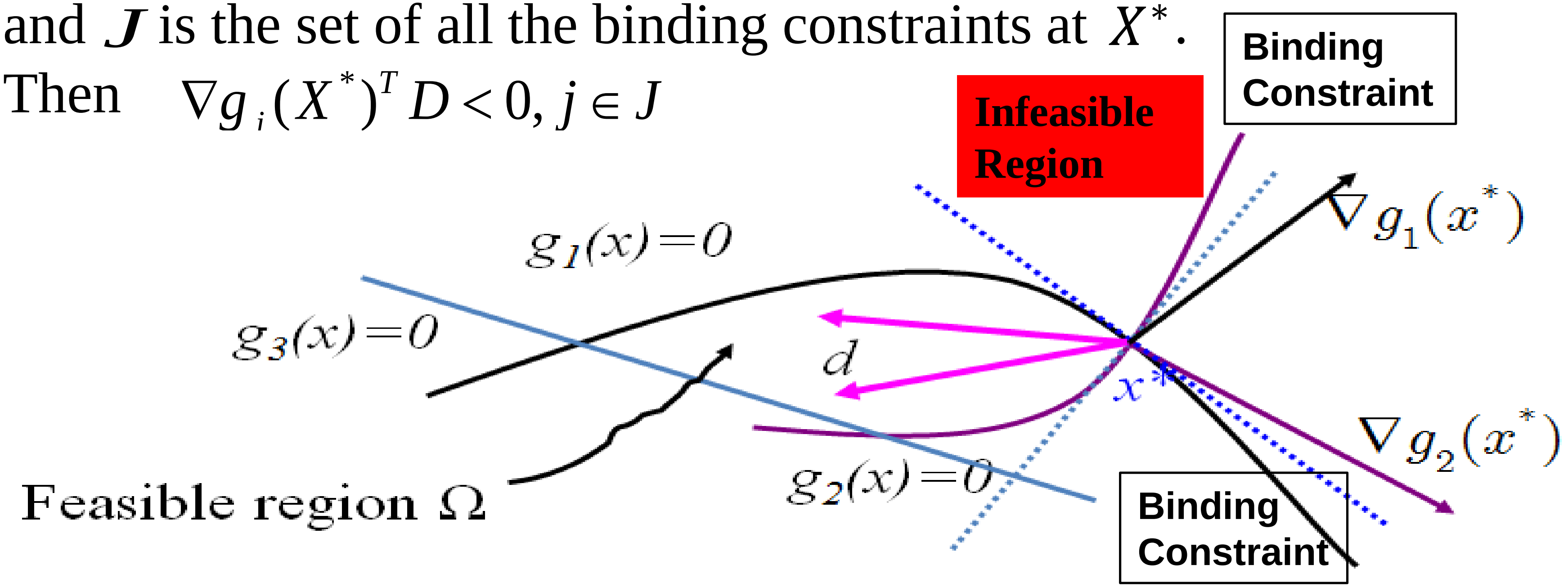
Preliminary knowledge (con't)

❑Feasible directions of a NLP

- We first talk about the binding constraints

$$g_i(\mathbf{x}) \le 0 \quad i = 1, 2, \ldots, m$$

It is assumed that D is a feasible direction at X^*. The binding constraints at this points are $g_j(X^*) = 0$
and J is the set of all the binding constraints at X^*.
Then $\nabla g_j(X^*)^T D < 0, j \in J$



Preliminary knowledge (con't)

$X = X^* + \lambda D$

λ : step size

- **Feasible directions of a NLP**

Proof: using Taylor expansion

$$f(X) = f(X^*) + \nabla f\left(X^*\right)^{\mathrm{T}}\left(X - X^*\right) + o(X - X^*)$$

$$g_j(X^* + \lambda D) = g_j(X^*) + \lambda \nabla g_j(X^*)^T D + o(\lambda)$$

Since $g_j(X^*) = 0$

For a feasible direction D, we have: $g_j(X^* + \lambda D) \le 0,\ j \in J$

Thus, $g_j(X^*) + \lambda \nabla g_j(X^*)^T D + o(\lambda) \le 0 \Rightarrow \nabla g_j(X^*)^T D < 0,\ j \in J$

For nonbinding constraints, $g_i(X^{(0)}) < 0, i \notin J$

Then, for any direction D **, there always exist** $\lambda > 0$ **, satisfy**

$$g_i(X^{(0)} + \lambda D) \le 0, i \notin J$$

Preliminary knowledge (con't)

Descent directions of a NLP

- A vector D is a descent direction at $X^{(0)}$, if there exits a real number λ >0 , such that $f(X^{(0)}+\lambda D)< f(X^{(0)})$
- For any decent direction, we further have $\nabla f(X^{(0)})^T D<0$
- Proof: Using Taylor expansion of $f(X)$ at $X^{(0)}$, we have

$$f(X^{(0)}+\lambda D)= f(X^{(0)})+\nabla f\left(X^{(0)}\right)^{\mathrm{T}}\lambda D+o(X-X^{(0)})$$

$$f(X^{(0)}+\lambda D)- f(X^{(0)})<0 \Rightarrow \nabla f\left(X^{(0)}\right)^{\mathrm{T}}\lambda D<0$$

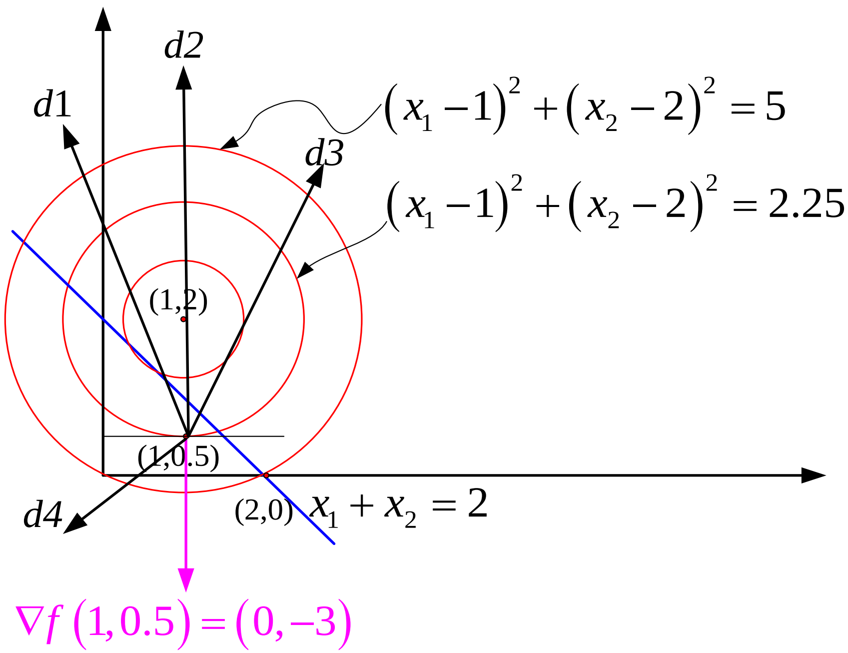
$$\nabla f(X^{(0)})^T D<0$$

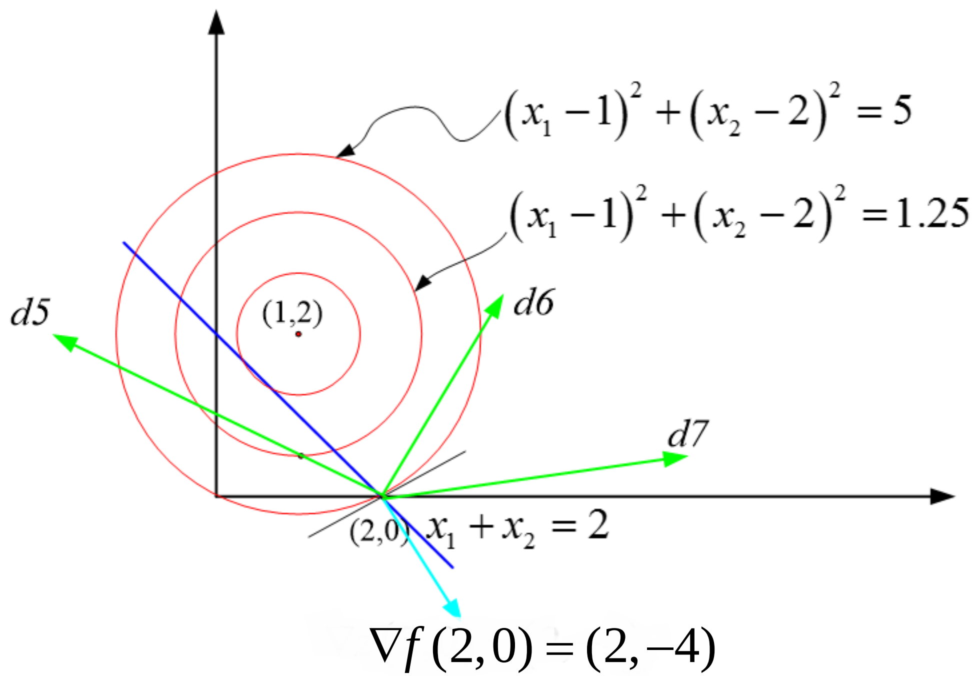
<u>Important conclusion:</u>
<u>negative gradient is the fastest descent direction</u>

Preliminary knowledge (con't)

❑Feasible Descent directions of a NLP

Example: $\Omega=\{(x_1,x_2)|x_1+x_2\leq 2,\ x_1\geq 0,\ x_2\geq 0\}$, and $f(x)=(x_1-1)^2+(x_2-2)^2$





At point (1,0.5)

Feasible descent directions: d_1, d_2, d_3

At point (2,0)

Feasible descent direction: d_5

Preliminary knowledge (con’t)

Feasible descent directions of a NLP

Literally ,

Feasible descent direction: both feasible and descent.

- If $X^{(0)}$ is not a local minimum, the search direction of the next iteration should be a *feasible descent direction at this point*;
- If $X^{(0)}$ is a local minimum, it dose not have feasible descent direction;
- On the other hand, if a point has feasible descent direction, it is definitely not the local minimum.

Preliminary knowledge (con't)

❑Feasible descent directions of a NLP

Mathematically ,

If point $X^{(0)}$ is not the local minimum, there must exist a direction D that satisfy the following inequalities simultaneously,

$$\begin{cases} \nabla f(X^{(0)})^T D < 0 \\ \nabla g_j(X^{(0)})^T D < 0, j \in J \end{cases}$$

Graphically ,

- the included angle between D and the *negative gradient of the objective function* is acute;
- the included angle between D and the *negative gradient of the binding constraints* is acute.

Convex Optimization

Convex Optimization

- Convex Programming – Convex Optimization
  - Minimize **Convex Function** over **Convex Set**
- A special subfield of nonlinear programming
  - The most commonly addressed NLP
  - Much easier to solve than other NLP models.

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

where

$$\Omega = \{x \mid g(x) \leq 0, h(x) = 0\}$$

Definition of Convex Set

❑Sets in the n-dimensional Real Space R^n

Let Ω be a set of a vector space over the real numbers $\Omega \subseteq R^n$

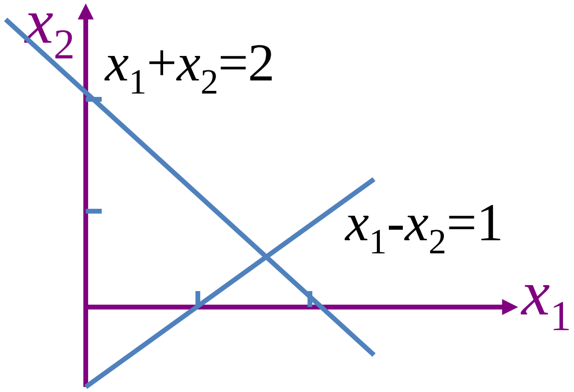
- Definition

A set Ω in R^n is said to be convex if, for all x_1 and x_2 in Ω and all α in the interval [0, 1], the point $\alpha x_1 + (1-\alpha) x_2$ also belongs to Ω .

$$\alpha x_1 + (1-\alpha) x_2 \in \Omega$$

- Example 1

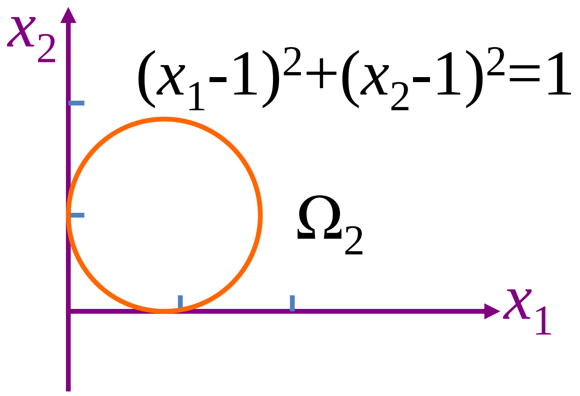
$\Omega_1 = \{(x_1, x_2) | \; x_1 + x_2 \leq 2; \; x_1 - x_2 \leq 1; \; x_1 \geq 0; x_2 \geq 0\}$



Definition of Convex Set

- Example 2

$$\Omega_2=\{(x_1,x_2)|\ (x_1-1)^2+(x_2-1)^2\leq 1\}$$

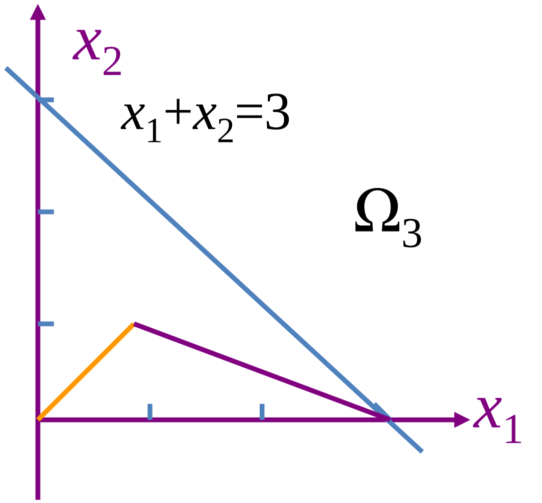


- Example 3

$$\Omega_3=\{(x_1,x_2)|\ x_1+x_2\leq 3;\ g(x)<=0;x_1\geq 0;x_2\geq 0\}$$

where

$$g(x)=\begin{cases} x_1-x_2, & 0\leq x_1\leq 1 \\ 3-x_1-2x_2, & x_1\geq 1 \end{cases}$$

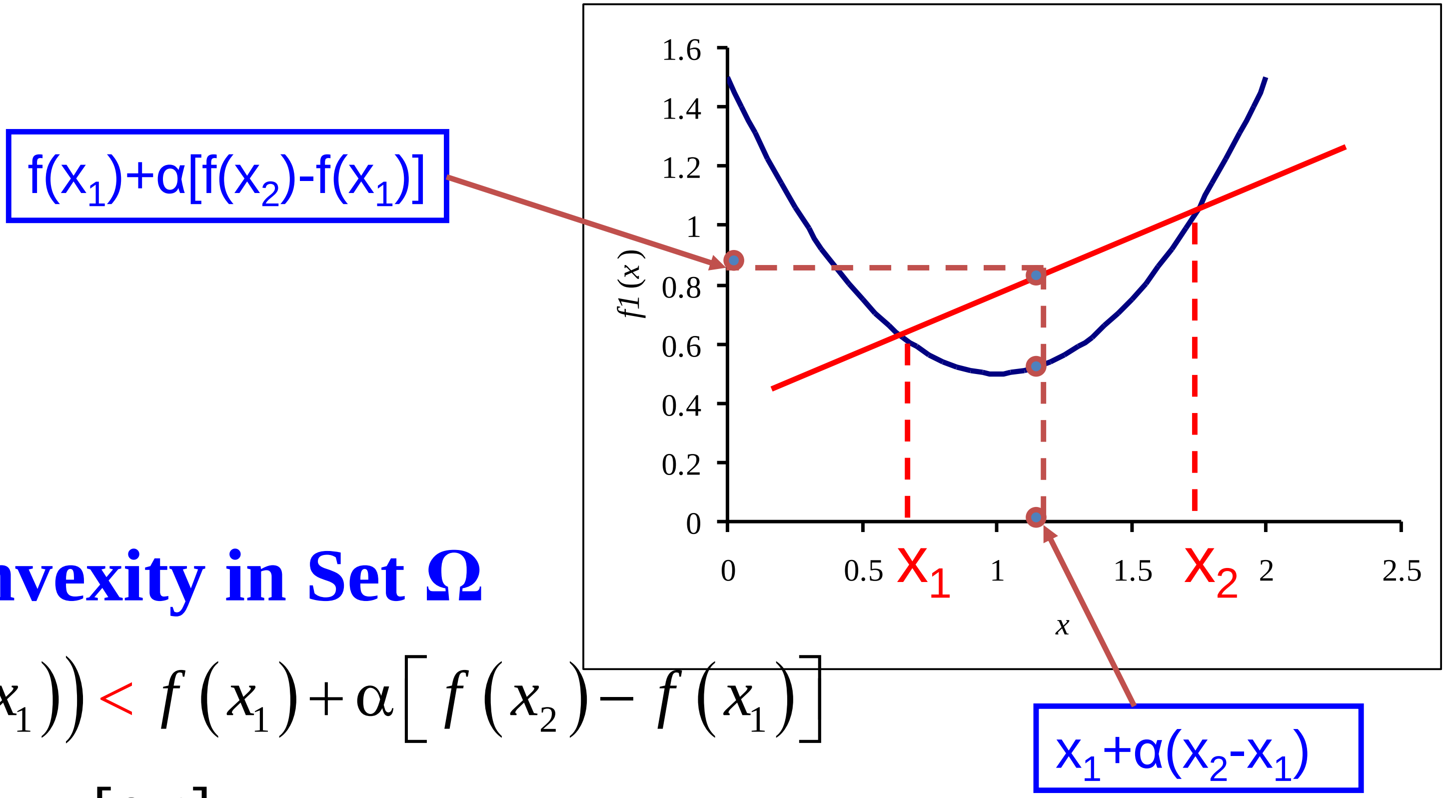


Definition of Convex Functions

Convexity in Set Ω

$$f\left(x_1+\alpha\left(x_2-x_1\right)\right)\leq f\left(x_1\right)+\alpha\left[f\left(x_2\right)-f\left(x_1\right)\right], \forall x_1, x_2 \in \Omega, \forall \alpha \in [0,1]$$

$$f: R^n \rightarrow R$$



- **Strict Convexity in Set Ω**

$$f\left(x_1+\alpha\left(x_2-x_1\right)\right) < f\left(x_1\right)+\alpha\left[f\left(x_2\right)-f\left(x_1\right)\right]$$

$$\forall x_1, x_2 \in \Omega, \forall \alpha \in [0,1]$$

First-order Condition

❑Conditions ensuring convexity of a function

Let C be a nonempty convex subset of R^n and let $f : R^n \to R$ be differentiable over an open set that contains C.

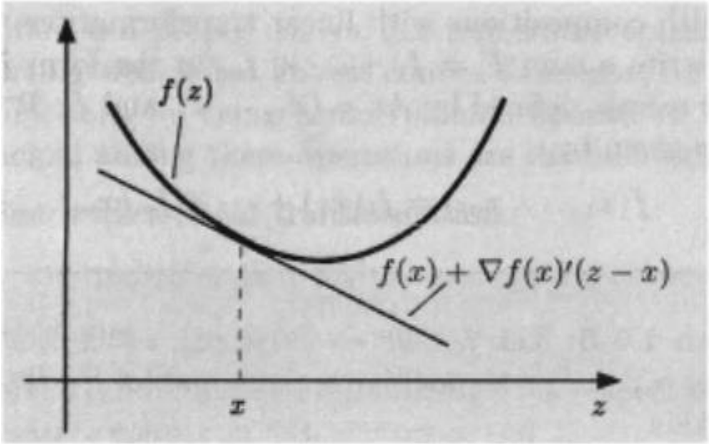
▪A sufficient condition

(i) f is **convex** over C iff

$$f(z) \geq f(x) + \nabla f(x)'(z - x), \forall x, z \in C$$

(ii) f is **strictly convex** over C iff

$$f(z) > f(x) + \nabla f(x)'(z - x), \forall x \neq z, x, z \in C$$



Second-order Condition

❑ Conditions checking convexity of a function

Assume that Hessian matrix $\nabla^2 f\left(x\right)$ **does exist** for any feasible x in set Ω

▪ **A second-order condition**

(i) Function *f(x)* is convex if and only if (iff) the Hessian matrix $\nabla^2 f\left(x\right)$ is **semi-positive definite**.

(ii) Function *f(x)* is **strictly convex** if and only if the Hessian matrix $\nabla^2 f\left(x\right)$ is **positive definite**.

Convex Functions

- **Examples**
  - Function $f_1(x) = (x_1 - 2)^2 + (x_2 - 2)^2$

    It is a strictly convex function

  - Function $f_2(x) = 2x_1 + 3x_2 + 0.5x_3 - x_4$

    It is a convex but not strictly function

  - Remark

    All the linear functions are convex. Are they concave?

Convex Functions (con't)

- Example 1:

Prove that $f(x) := \sum_{j=1}^{n} c_j x_j$ is a convex function on its natural domain

- Example 2:

Suppose that $c_j > 0,\ j = 1, 2 \ldots n.$

Prove that $f(x) := \sum_{j=1}^{n} c_j x_j^{\ 2}$ is a convex function (Hint: $2ab \le a^2 + b^2$)

- Example 3:

Give an example of the values of $c_j,\ j = 1, 2 \ldots n,$

such that $f(x) := \sum_{j=1}^{n} c_j x_j^{\ 2}$ is no longer a convex function

Convex Optimization

The following statements are true about the convex minimization problem:

- if a local minimum exists, then it is a global minimum.
- the set of all (global) minima is convex.
- for each strictly convex function, if the function has a minimum, then the minimum is unique.

Examples:

- Least squares
- Linear programming (convex or concave?)
- Convex quadratic minimization with linear constraints
- Conic optimization（锥优化）
- Entropy maximization with appropriate constraints

Local and Global Minimum of a convex optimization problem

Local minimum

If a x^* in a convex set Ω satisfies $\nabla f(x^*)^T d \geq 0$ for any feasible direction d at x^*, then x^* is a ***global minimum*** for the convex programming problem: $\min_{x \in \Omega} f\left(x\right)$

Proof.

Suppose that x^* is not a global minimum, then there is a point y in set Ω such that ***f(y)<f(x*).***

Since Ω is a convex set, for any $0 \leq \alpha \leq 1$, $x^* + \alpha(y - x^*)$ is still in set Ω. In other words, y-x^* is a feasible direction at point x^*.

As $f(x)$ is a convex function, it can be seen that

$$f(y) \geq f(x^*) + \nabla f(x^*)^T (y - x^*) \geq f(x^*)$$

This is a contradiction.

Karush-Kuhn-Tucker (KKT) Conditions

Karush-Kuhn-Tucker Conditions

- **How to verify x^* is an local optimum for the following nonlinear programming:**

$$\begin{aligned} &\min f(x) \\ \text{subject to} \quad & \\ & g_i(x) \le 0, \quad i = 1, \ldots, m_1 \\ & h_j(x) = 0, j = 1, \ldots, m_2 \end{aligned}$$

- ✓ Existing ***optimality conditions*** for optimization problems with constraints involve enumeration of all feasible directions.
- ✓ How to effectively handle these ***feasible directions***?

KKT Conditions (con't)

❑ A Necessary Optimality Condition

If x^* is a local minimum, then there exist Lagrangian multipliers $\{u_1, u_2, \ldots, u_{m1}\}$ and $\{\lambda_1, \lambda_2, \ldots, \lambda_{m_2}\}$ satisfy the following KKT conditions:

$$(1)\ \nabla f\left(x^*\right) + \sum_{i=1}^{m_1} u_i \nabla g_i\left(x^*\right) + \sum_{j=1}^{m_2} \lambda_j \nabla h_j\left(x^*\right) = 0$$ ← (KKT Equations)

$$(2)\ u_i \geq 0, i = 1, \cdots, m_1$$ ← (Non-negativity)

$$(3)\ u_i g_i\left(x^*\right) = 0, i = 1, \cdots, m_1$$ ← (Complementary slackness conditions)

$$(4)\ g_i\left(x^*\right) \leq 0, i = 1, \cdots, m_1, h_j\left(x^*\right) = 0, j = 1 \cdots, m_2$$ ← (Feasibility)

Constraint Qualification (CQ), or Regularity Condition

- For the KKT conditions to hold, $g_i(x)$ and $h_j(x)$ must satisfy certain regularity conditions (usually called ***constraint qualifications***).
- The most used CQs are listed below:
- **Linearity constraint qualification (LCQ):**

If $g_i(x)$ and $h_j(x)$ are affine functions, then no other condition is needed.

- **Linear independence constraint qualification (LICQ):**

$$\nabla g_i(x) \text{ and } \nabla h_j(x) \text{ are linearly independent.}$$

- ……

Reference:

- Chapter 5 of *Nonlinear Programming: Theory and Algorithms* by Bazaraa et al. (1993)
- Eustaquio, R. G., Karas, E. W., & Ribeiro, A. A. (2010). Constraint qualifications for nonlinear programming.

KKT Conditions (con't)

❑ Remarks

- The KKT conditions provide a necessary condition for the local minimum of a NLP.

- The KKT conditions are the necessary and sufficient conditions for the local minimum of **a convex programming problem.**

- **Any local minimum is also global minimum for a convex programming problem**.

- A strictly convex programming problem has **the unique optimal solution**.

KKT Conditions (con't)

❑ Example

Check that $x^*=(0.5,1.5)$ is the unique global optimal solution for the following minimization problem:

$$\min f(x)=(x_1-2)^2+(x_2-3)^2$$

subject to

$$x_1+x_2\le 2 \quad \longleftarrow u_1$$

$$x_1\ge 0 \quad \longleftarrow u_2$$

$$x_2\ge 0 \quad \longleftarrow u_3$$

(1) Verify the feasibility condition

$$x_1^*+x_2^*=2$$

$$x_1^*=0.5>0$$

$$x_2^*=1.5>0$$

KKT Conditions (con't)

(2) Verify the complementary slackness conditions

$$u_2 \times \left(-x_1^*\right) = 0, \quad u_3 \times \left(-x_2^*\right) = 0, \quad u_1^* \times \left(x_1^* + x_2^* - 2\right) = 0$$

It can be seen that $u_2=0, u_3=0$ and u_1 is a unknown and nonnegative multiplier.

(3) Solve the KKT equations

$$\begin{pmatrix} 2\left(x_1^* - 2\right) \\ 2\left(x_2^* - 3\right) \end{pmatrix} + u_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + u_2 \begin{pmatrix} -1 \\ 0 \end{pmatrix} + u_3 \begin{pmatrix} 0 \\ -1 \end{pmatrix} = 0 \Rightarrow u_1 = 3$$

(4) Check the non-negativity conditions

$$u_1 = 3 > 0, u_2 = 0, u_3 = 0$$

KKT Conditions (con't)

- **Example 1:**

Use the KKT conditions to solve the following NLP

$$\begin{cases} \min\ f(x) = (x-3)^2 \\ 0 \le x \le 5 \end{cases}$$

- Standard form:

$$\begin{cases} \min\ f(x) = (x-3)^2 \\ \qquad g_1(x) = -x \le 0 \\ \qquad g_2(x) = x-5 \le 0 \end{cases}$$

- Obtain the gradient of the objective function and constraints

$$\nabla f(x) = 2(x-3),$$

$$\nabla g_1(x) = -1, \nabla g_2(x) = 1$$

KKT Conditions (con't)

- **Example 1:**
- Introduce the multipliers

$$
\begin{cases}
\min\ f(x) = (x-3)^2 \\
g_1(x) = -x \le 0 & \longrightarrow \quad \gamma_1 \\
g_2(x) = x - 5 \le 0 & \longrightarrow \quad \gamma_2
\end{cases}
$$

Convex function
Convex set

$\Downarrow$

Convex optimization

- Based on the KKT conditions, let the local minimum be $x^*, \gamma_1^*, \gamma_2^*$
- Then, they should satisfy

$$
\begin{aligned}
&2(x^* - 3) - \gamma_1^* + \gamma_2^* = 0 \qquad && \gamma_1^*, \gamma_2^* \ge 0 \\
&\gamma_1^* x^* = 0 && -x^* \le 0 \\
&\gamma_2^*(x^* - 5) = 0 && x^* - 5 \le 0
\end{aligned}
$$

KKT Conditions (con't)

- **Example 1:**

$$2(x^* - 3) - \gamma_1^* + \gamma_2^* = 0 \qquad \gamma_1^*, \gamma_2^* \geq 0$$

$$\gamma_1^* x^* = 0 \qquad -x^* \leq 0$$

$$\gamma_2^*(x^* - 5) = 0 \qquad x^* - 5 \leq 0$$

- Discussions:
- (1) Let $\gamma_1^* \neq 0, \gamma_2^* \neq 0,$ no feasible solution;
- (2) Let $\gamma_1^* \neq 0, \gamma_2^* = 0 \Rightarrow x^* = 0, \gamma_1^* = -6,$ doesn't satisfy KKT;
- (3) Let $\gamma_1^* = 0, \gamma_2^* \neq 0 \Rightarrow x^* = 5, \gamma_2^* = -4,$ doesn't satisfy KKT;
- (4) Let $\gamma_1^* = \gamma_2^* = 0 \Rightarrow x^* = 3,$ satisfies KKT → a local minimum.
- The problem is a convex model, so the local minimum is a also a global minimum

Difficulties with NLPs

- The objective function and constraints must be continuously differentiable
- CQ must be satisfied at the points fulfilling KKT conditions.
- Finding points that fulfill KKT conditions is no easy task (involves many equalities and inequalities).
- Moreover, there may be many solutions fulfilling the KKT conditions. So, efficient algorithms are still needed, instead of solving the KKT conditions directly.

Solution Algorithms

Solution Algorithms

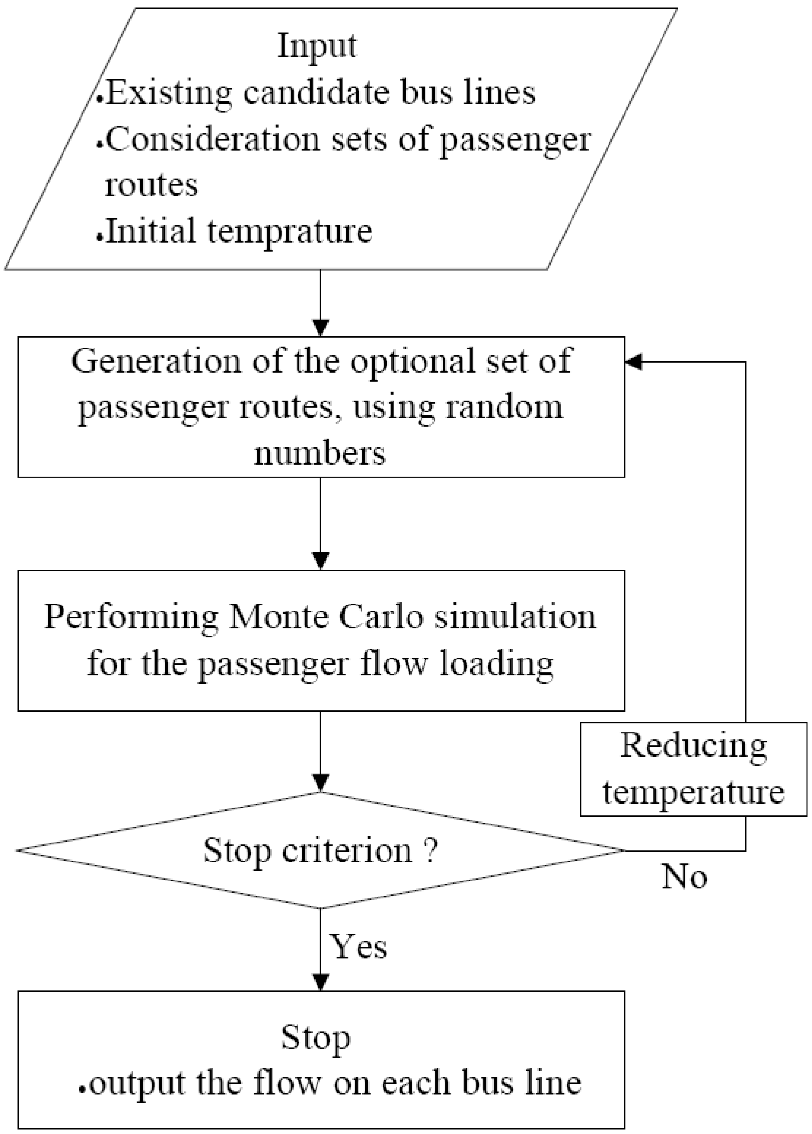
- What are your understanding of an algorithm?
- Processes constitute an algorithm for solving a mathematical model?
  - Input
  - Initiation
  - Descend direction
  - Line search
  - Stop Criterion:

$$\left| x_{k+1} - x_k \right| \leq \varepsilon$$

  - where ε is a given positive constant

Solution Algorithms

- How to draw a flowchart for an algorithm?



Solution Algorithms

- We first talk about methods for line search.
- After the descend direction is obtained, a line search is necessary to locate the optimal point for next step along this direction.
- So, line search problem usually only has one dimension.
- Three methods to solve such a one-dimensional optimization problem.
  - Bisection Algorithm
  - Golden Section Algorithm
  - Newton Method

Line Search Methods – Bisection Algorithm

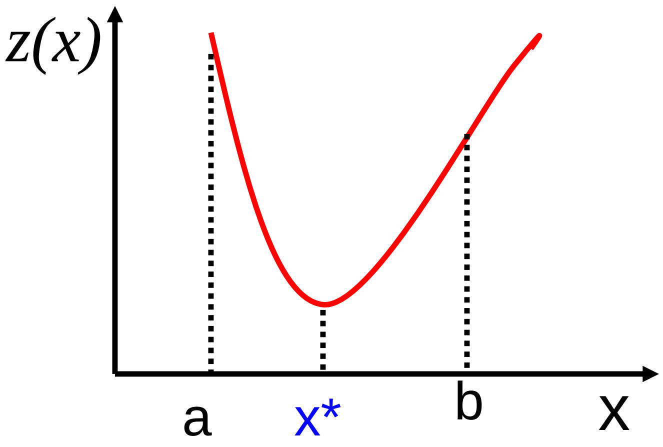
- **Assumption**

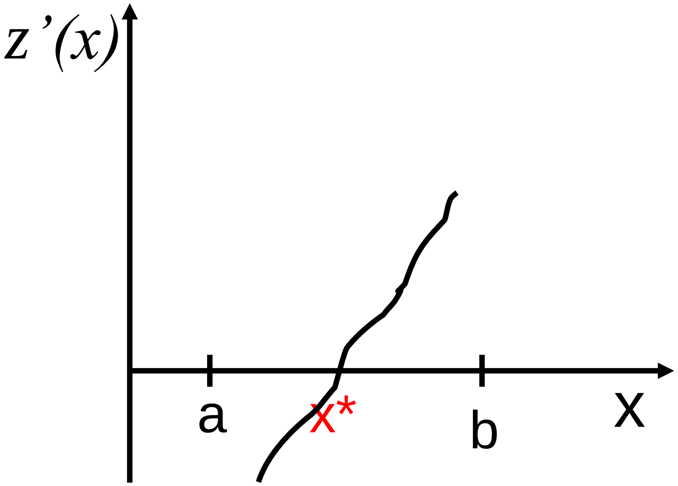
  z(x) is a convex and continuously differentiable function in the interval [a, b]

- **Motivation**

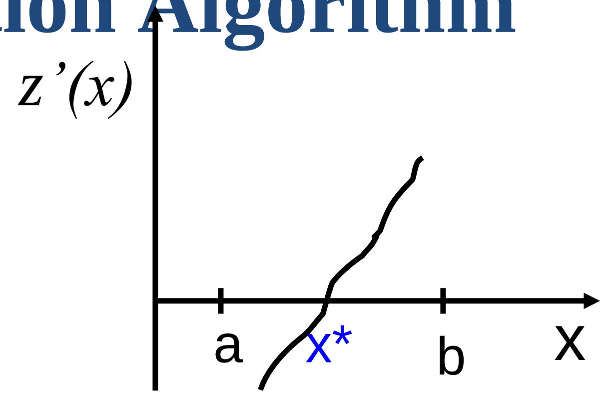
  Find a zero point of function *z'(x)* in the interval [a, b], i.e.,

  $z'(x)=0$ with $a \le x \le b$





Line Search Methods – Bisection Algorithm



❑ Algorithmic Framework

$$\boxed{(b_{k+1} - a_{k+1}) = 0.5 * (b_k - a_k)}$$

Step 0: Given an initial interval

$[a_0, b_0] \subseteq [a, b]$ wher $x^* \in [a, b]$ and $z'(a_0) < 0, z'(b_0) > 0$

Let k=0

Step 1: if $b_k - a_k \leq \varepsilon$, then stop and yield an optimal minimum $x^* = (b_k - a_k)/2$. Otherwise, go to Step 2

Step 2: Set $x_k = (b_k - a_k)/2$

If $z'(x_k) \geqslant 0$, then $b_{k+1} = x_k$, $a_{k+1} = a_k$

If $z'(x_k) < 0$, then $a_{k+1} = x_k$, $b_{k+1} = b_k$

Set k=k+1 and go to Step 1.

Line Search Methods – Bisection Algorithm

- **Example**

- Use the bisection algorithm to solve the following NLP.

$$\min z(x) = \sin x$$

$$s.t.\ 3 \le x \le 6$$

- Because $z'(x) = \cos x, a_0 = 3, b_0 = 6, x_0 = 4.5$
- Then $z'(3) = -0.99 < 0, z'(6) = 0.96 > 0, z'(x_0) = -0.2108$

$\Rightarrow$ $a_1 = 4.5, b_1 = 6, x_1 = 5.25$

Line Search Methods – Bisection Algorithm

- **Example**
- Iterative Process of Bisection Algorithm

k	a_k	b_k	x_k	$z'(x_k)$	$z'(a_k)$	$z'(b_k)$
0	3	6	4.5	-0.2108	-0.99	0.96017
1	4.5	6	5.25	0.51209	-0.2108	0.96017
2	4.5	5.25	4.875	0.1619	-0.2108	0.51209
3	4.5	4.875	4.6875	-0.0249	-0.2108	0.1619
4	4.6875	4.875	4.78125	0.06881	-0.0249	0.1619
5	4.6875	4.78125	4.73438	0.02198	-0.0249	0.06881
6	4.6875	4.73475	4.71113	-0.0013	-0.0249	0.02236

Comments on bisection search method

- In step 2, why don't we consider the case of $z(x_k)=0$ separately?
  - Due to problem structure and numerical errors, the possibility that $z(x_k)=0$ is very low.
- Which should be taken as the output?
  - $(a_k+b_k)/2$, *or* a_k *or* b_k
  - This does not matter much because the tolerance ε is generally very small.

Comments on bisection search method

- How is ε determined?
  - A larger ε gives to a faster convergent speed, but lower precision. Therefore, ε is chosen as a tradeoff between the efficiency and quality.
  - For example, if *x* represents the unit price at the retailer market, then it is acceptable to set ε at 1 cent.
- Any other stopping criteria?

– Yes. e.g., $|z(x_k)|<\varepsilon$ (the number of iterations required will change)

– We can also stop after a given number of iterations

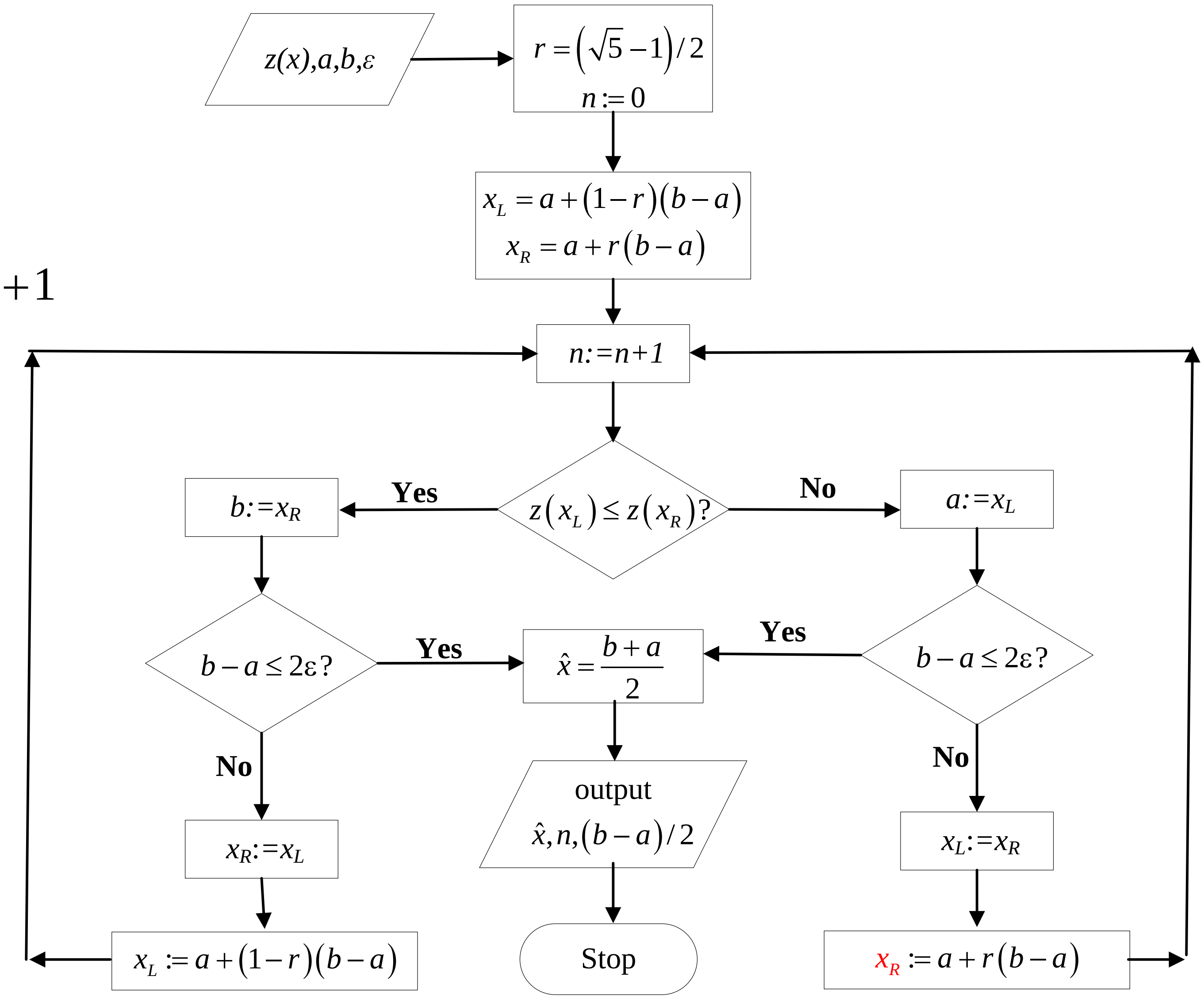
– We can also stop after a given CPU time

Conclusions

- Bisection algorithm is used to solve convex and continuously-differentiable problem.
  - Invalid if the problem is not differentiable.
- So, Golden section algorithm is used, and it doesn't even require the problem to be convex.
  - Read the textbook for more details of the Golden section algorithm.

Line Search Methods – Golden Section Algorithm

$$\min_{a \le x \le b} z(x)$$



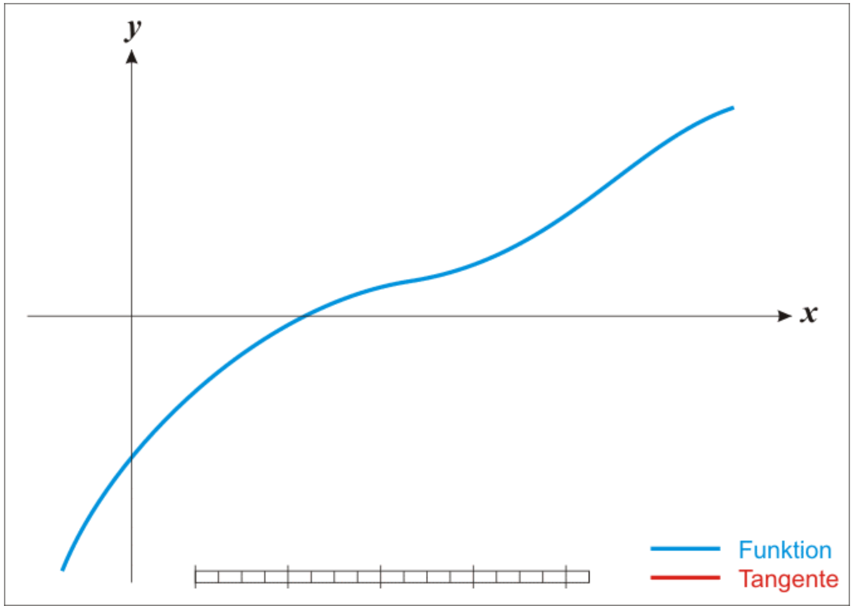
Example:

$$\min_{2 \le x \le 4} z(x)=(x-1)^2+1$$

Line Search Methods – Newton's Method

Idea of Newton Method $\min_{x \in R^n} z(x) \Rightarrow z'(x) = 0$

This method starts with an initial guess which is reasonably close to the true solution, then the function is approximated by its tangent line, and one computes the *x*-intercept of this tangent line. This *x*-intercept will typically be a better approximation to the function' solution than the original guess, and the method can be iterated.



$$z'(x) = 0$$

Newton's Method

$$\begin{cases} z'(x_n) = 0 \\ z''(x_n) \neq 0 \end{cases}$$

❑Newton Method for Line Search

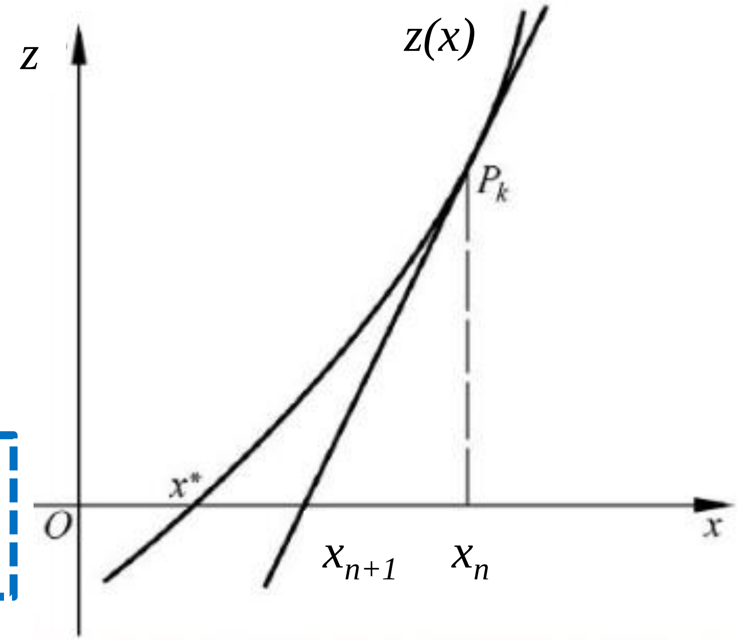
- we suppose the current solution is x_n and x_{n+1} is the next solution. The tangent line of $y = z''(x)$ at x_n is

$$y = z'(x_n) + z''(x_n)(x - x_n)$$

- The x-intercept of this line is regarded

as the next approximation root, that is,

$$x_{n+1} = x_n - \frac{z'(x_n)}{z''(x_n)}$$



This method will end until the solution satisfy a predetermined stop criterion.

Newton's Method

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

Assume that *z(x)* is second-order continuous differentiable.

Newton Method for Line Search

- Explanation based on the 2nd order Taylor approximation

$$z(x) \approx z(x_k) + z'(x_k)(x - x_k) + \frac{1}{2} z''(x_k)(x - x_k)^2$$

First order derivative: $z'(x) = z'(x_k) + z''(x_k)(x - x_k)$

- To minimize $z(x)$, first-order necessary optimality: $z'(x) = 0$

$$\Rightarrow \quad z'(x_k) + z''(x_k)(x - x_k) = 0 \Rightarrow x = x_k - \frac{z'(x_k)}{z''(x_k)}$$

- Thus, based on x_k, $x_{k+1} = x_k - \frac{z'(x_k)}{z''(x_k)}, k = 0,1...$

Gradient Descent

Gradient Descent

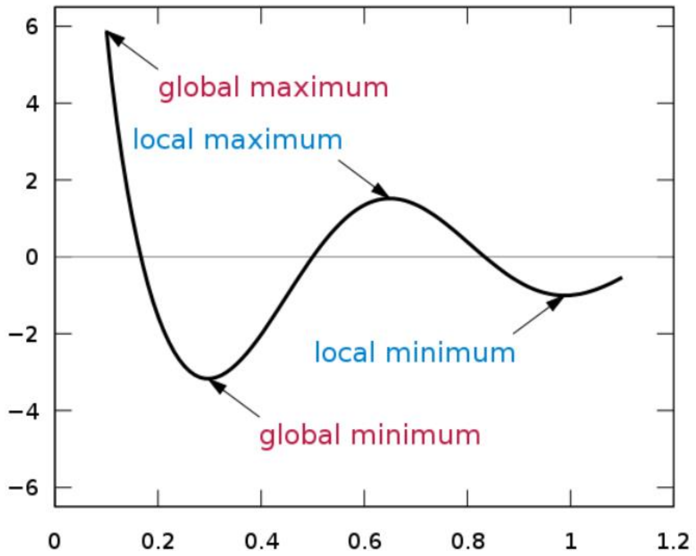
❑ Problem

Find a local minimum of a differentiable function $z(x), x \in R^n$.

x^* is a (global) minimum of $z(x)$ iff $z(x^*) \leq z(x), \forall x \in R^n$.

x^* is a local minimum of $z(x)$ iff there exists an $\varepsilon > 0$ and

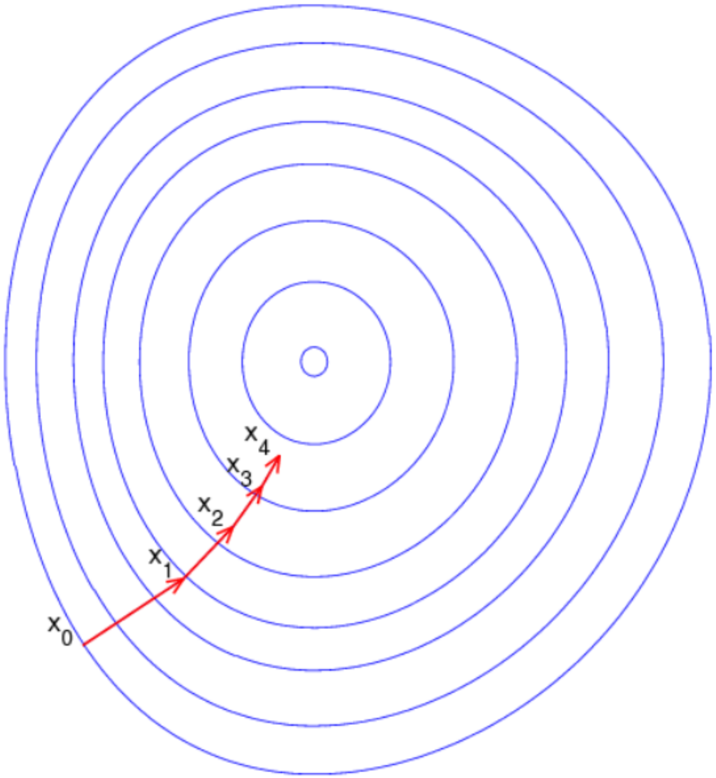
$z(x^*) \leq z(x), \forall x \in R^n$ and $\| x - x^* \|_2 < \varepsilon$.



Gradient Descent (con't)

❑ Idea of Descent Direction Methods

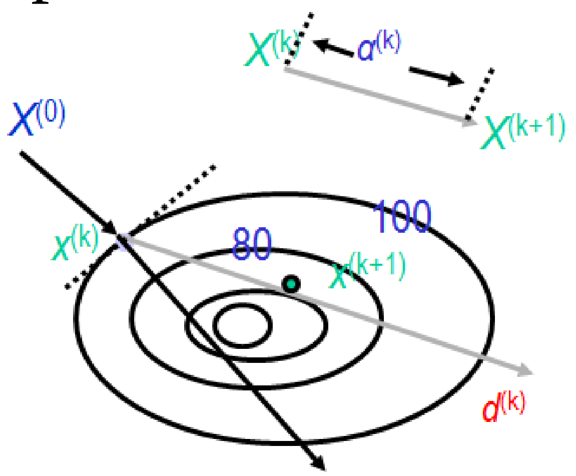
- Moving from one point to another, while reducing the objective value.



Gradient Descent (con't)

Framework of Descent Direction Methods

- Step 0: Choose an initial point $x^{(0)}$
- Step 1: Verify a stopping criterion
- Step 2: Find a feasible descent direction $d^{(k)}$ (feasibility is automatically ensured)
- Step 3: Identify an optimal/good step size $\alpha^{(k)}$ in connection with direction $d^{(k)}$
- Step 4: Set $x^{(k+1)}=x^{(k)}+\alpha^{(k)}d^{(k)}$



Gradient Descent (con't)

- **Steepest Descent/Gradient Descent**

- A natural idea is to use the negative gradient direction.

$$\nabla z(x)\Big|_{x=x^{(k)}} := \begin{pmatrix} \dfrac{\partial z}{\partial x_1} \\ \dfrac{\partial z}{\partial x_2} \\ \vdots \\ \dfrac{\partial z}{\partial x_n} \end{pmatrix}_{x=x^{(k)}} \quad \text{or } \nabla z\left(x^{(k)}\right) := \begin{pmatrix} \dfrac{\partial z}{\partial x_1^{(k)}} \\ \dfrac{\partial z}{\partial x_2^{(k)}} \\ \vdots \\ \dfrac{\partial z}{\partial x_n^{(k)}} \end{pmatrix}$$

Gradient Descent (con't)

❑ Gradient Descent Algorithm

Step 0: Choose an initial point $x^{(0)}$ and set $k = 0$. Define $\varepsilon > 0$

Step 1: If $|\nabla z(x^{(k)})| < \varepsilon,$ then stop. Otherwise, go to Step 2

Step 2: Set $d^{(k)} := -\nabla z(x^{(k)})$. Find an $\alpha^{(k)}$ which is the optimal solution of one-dimensional minimization problem: $\min\limits_{0 \le \alpha < \infty} z\left(x^{(k)} + \alpha d^{(k)}\right)$

Step 3: Set $x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}$, and let $k = k+1$, go to Step 1.

Gradient Descent (con’t)

Gradient Descent Algorithm

- In Step 0, in general, it is preferable to choose an initial point that is near to a local minimum. (e.g., use the optimal solution of a heuristic algorithm as the initial solution)
- In Step 1, other stop criteria are also possible.
- The sub-problem in Step 2 is a line search. It is much easier to solve than the original problem. Herein, we can use the KKT conditions, bisection search, or golden section.

Gradient Descent (con't)

- **Example**

$$\min_{x\in R^2} z(x) = (x_1 - 2)^2 + 10(x_2 - 2)^2$$

- Gradient $\nabla z(x) = (2x_1 - 4, 20x_2 - 40)^T$
- Line search

$$f(\alpha) = z\left(x^{(k)} - \alpha\nabla z\left(x^{(k)}\right)\right) = \left(x_1 + \alpha\left(4 - 2x_1^{(k)}\right) - 2\right)^2 + 10\left(x_2^{(k)} + \alpha\left(40 - 20x_2^{(k)}\right) - 2\right)^2$$

✓ A parabola, Let $f'(\alpha)=0$, it follows that:

$$0 \le \alpha_k = \frac{(4 - 2x_1^{(k)})^2 + \left(40 - 20x_2^{(k)}\right)^2}{2(4 - 2x_1^{(k)})^2 + 20\left(40 - 20x_2^{(k)}\right)^2}$$

Gradient Descent (con't)

❑ Example

- Iterative scheme (2 iterations)

k	$x^{(k)}$	$d^{(k)}$	$\lvert d^{(k)} \rvert$	$\alpha^{(k)}$	$z(x^{(k)})$
0	(-4,-3)	(12,100)	100.7	0.051	286
1	(-3.392,2.065)	(10.784,-1.294)	10.9	0.443	29.118
2	(1.389, 1.491)				

Gradient Descent (con't)

- **Comments**

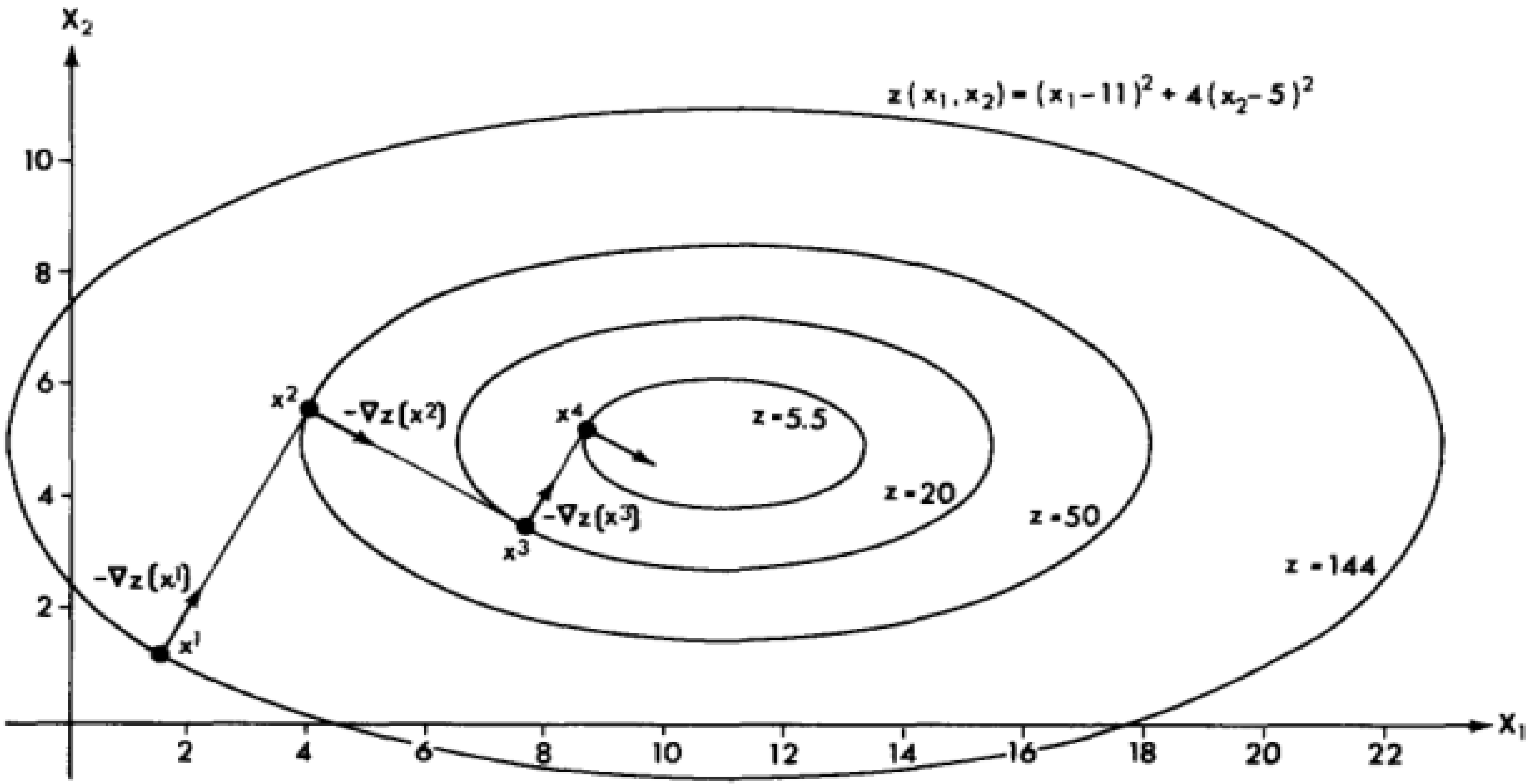
- When $z(x)$ is convex, gradient descent would end up at a global optimum.
- Slow convergence due to the "zigzaging" nature of the method.
  - Quadratic function
  - Rosenbrock function

$$f(x_1, x_2) = (1 - x_1)^2 + 100[x_2 - (x_1)^2]^2$$

Gradient Descent (con’t)

- **Comments**

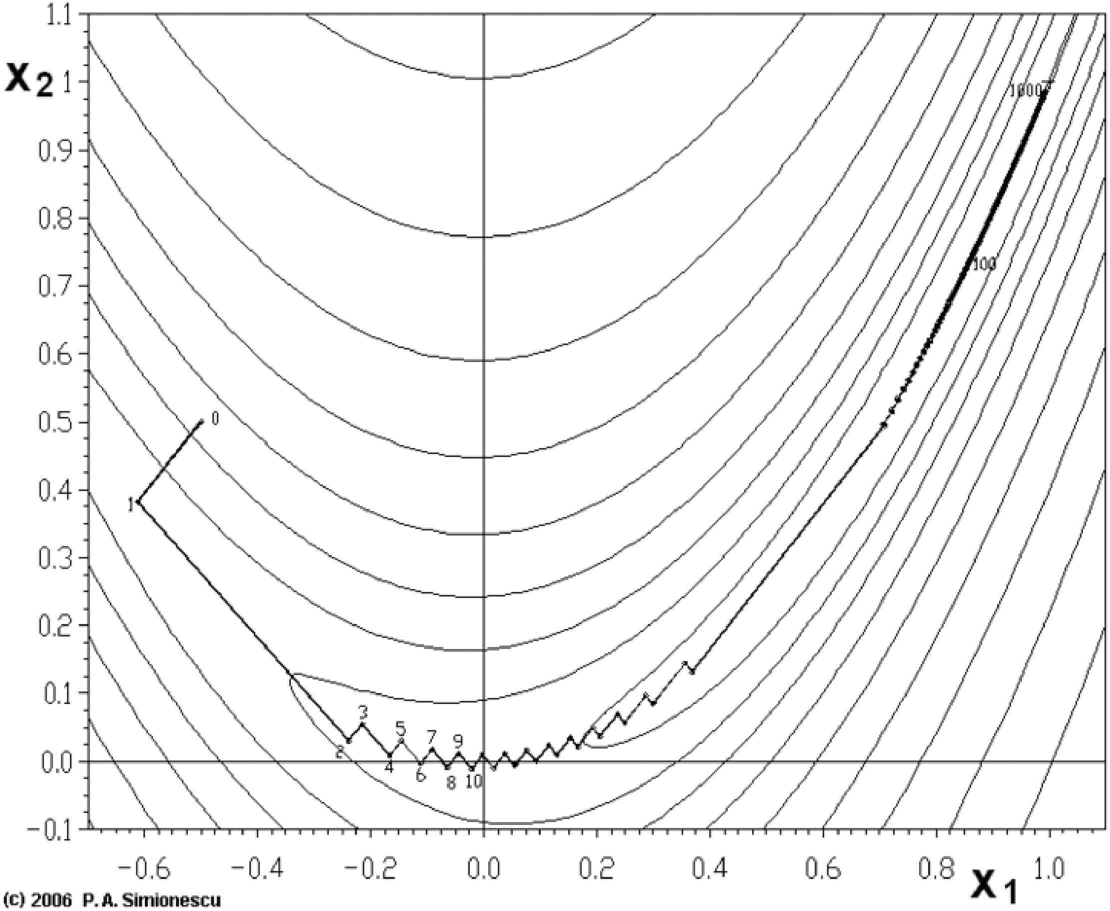
Zigzaging



Gradient Descent (con’t)

❑ Comments

- Rosenbrock function.



Frank-Wolfe Method

Frank-Wolfe Method

❑ Purpose

Find an optimal solution of the following convex programming problem by solving a series of linear programming (LP) problems.

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

subject to

$$g(x) \leq 0$$

$$h(x) = 0$$

▪ Frank, M. and Wolfe (1956) An algorithm for quadratic programming. Naval Research Logistics quarterly Research, Vol. 14, pp. 43-53.

Frank-Wolfe Method

❑ Algorithmic Framework

Step 0: Choose an initial feasible point $x^{(0)}$ and set k=0

Step 1: Find the descent direction $d^{(k)}=y^{(k)}-x^{(k)}$ where $y^{(k)}$ is the optimal solution of the LP:

$$\min_{y} y^{\mathrm{T}} \nabla f\left(x^{(k)}\right)$$

subject to

$$g(y) \leq 0$$

$$h(y) = 0$$

Frank-Wolfe Method

Step 2: If $\left|\sum_{i=1}^{n}\left[\left(\partial f\left(x^{(k)}\right)/\partial x_i\right)\left(y_i^{(k)}-x_i^{(k)}\right)\right]\right|\leq\varepsilon$,then stop. Otherwise, go to Step 3

Step 3: Find an α_k , which is the optimal solution of the following one-dimension minimization problem, by using a line search method

$$\min_{0\leq\alpha\leq1} f\left(x^{(k)}+\alpha\left(y^{(k)}-x^{(k)}\right)\right)$$

Step 4: Set $x^{(k+1)}=x^{(k)}+\alpha_k\left(y^{(k)}-x^{(k)}\right)$, and let k=k+1, got to Step 1

Frank-Wolfe Method

- **Example**

Find an optimal solution of the following minimization method by the Frank-Wolfe method with the stop tolerance ε=0.01 and the initial solution $x^{(0)}$ =(0,0)

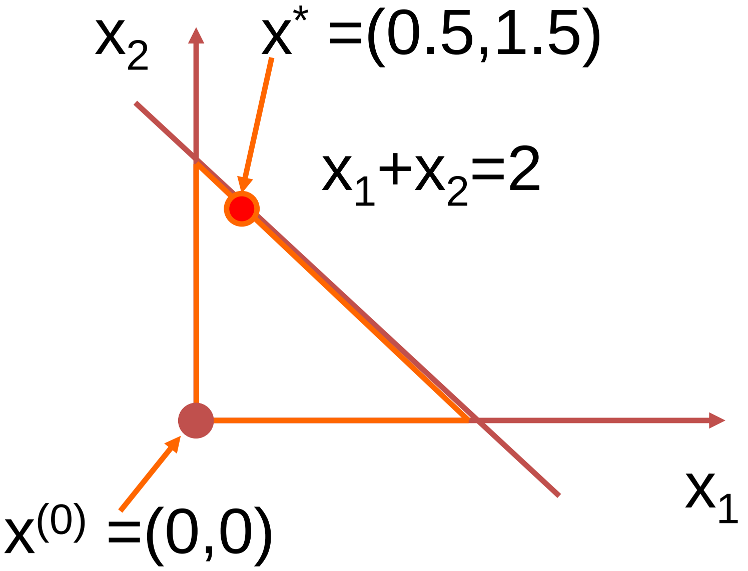
$$\min f(x) = (x_1 - 2)^2 + (x_2 - 3)^2$$

subject to

$$x_1 + x_2 \leq 2$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

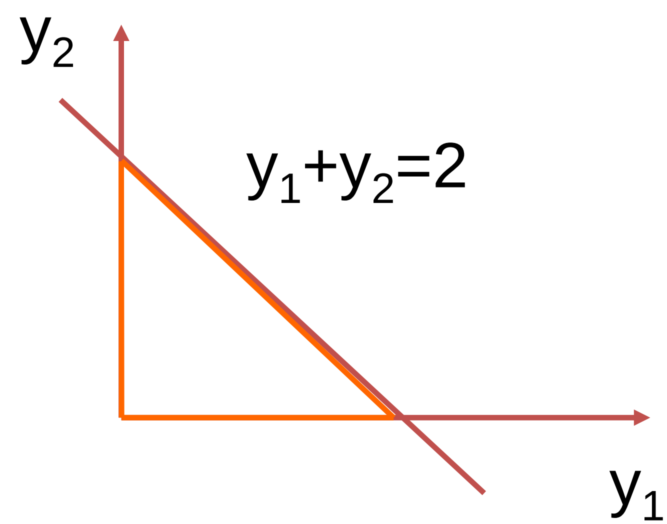


Frank-Wolfe Method

Iteration 1 (k=0)

$x^{(0)} = (0,0)^T$

$$\nabla f\left(x^{(0)}\right) = (-4,-6)^T$$



(1) Find the optimal solution $y^{(0)}$ of the LP: $\min z(y) = -4y_1 - 6y_2$

subject to
$$y_1 + y_2 \le 2$$
$$y_1 \ge 0$$
$$y_2 \ge 0$$

$\Rightarrow$ $y^{(0)} = (0,2)$

(2) Check the stop criterion:

$$\left|\sum_{i=1}^{n}\left[\left(\partial f\left(x^{(0)}\right)/\partial x_i\right)\left(y_i^{(0)} - x_i^{(0)}\right)\right]\right| = \left|\left[-4\times(0-0)+(-6)(2-0)\right]\right| > 0.01$$

Frank-Wolfe Method

(3) Find the optimal solution α_0 of the one-dimensional minimization problem by the bisection search method:

$$x^{(0)}+\alpha\left(y^{(0)}-x^{(0)}\right)=(0,0)^T+\alpha\left[(0,2)^T-(0,0)^T\right]=(0,2\alpha)^T$$

$$\min_{0\le\alpha\le1} f\left(x^{(0)}+\alpha\left(y^{(0)}-x^{(0)}\right)\right)=(0-2)^2+(2\alpha-3)^2$$

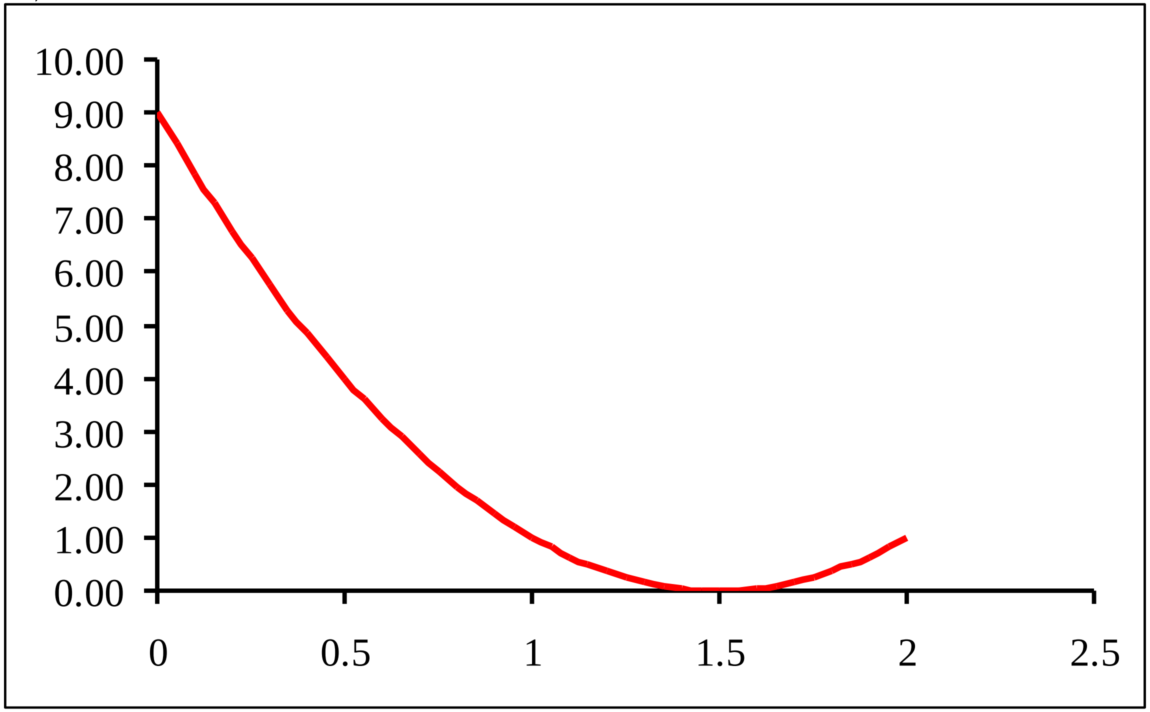
$\alpha_0=1$

(4) Update

$$x^{(1)}=x^{(0)}+\alpha_0\left(y^{(0)}-x^{(0)}\right)$$

$$=\begin{pmatrix}0+1\times(0\text{-}0)\\ 0+1\times(2\text{-}0)\end{pmatrix}$$

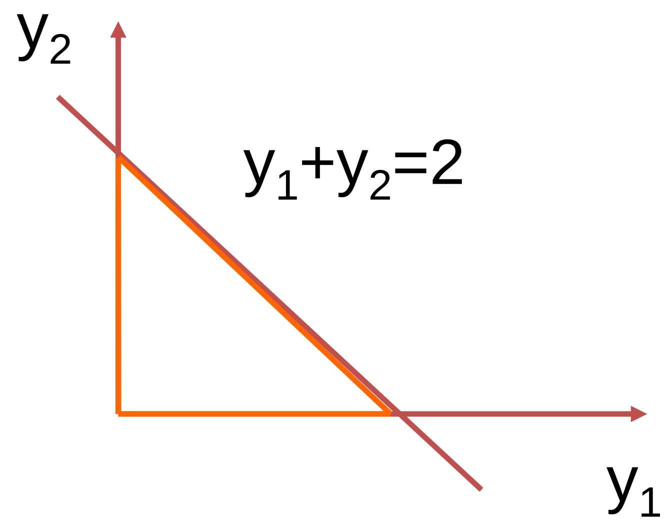
$$=\begin{pmatrix}0\\ 2\end{pmatrix}$$



Frank-Wolfe Method

<u>Iteration 2 (k=1)</u>

$$x^{(1)} = (0,2)$$



$$\nabla f\left(x^{(1)}\right) = (-4,-2)^T$$

(1) Find the optimal solution $y^{(1)}$ of the LP:

$$\min z(y) = -4y_1 - 2y_2$$

$$y_1 \geq 0$$

$$y_2 \geq 0$$

$$y_1 + y_2 \leq 2$$

$\Rightarrow$ $y^{(1)} = (2,0)$

(2) Check the stop criterion:

$$\left|\sum_{i=1}^{n}\left[\left(\partial f\left(x^{(1)}\right)/\partial x_i\right)\left(y_i^{(1)} - x_i^{(1)}\right)\right]\right| = \left|\left[-4\times(2-0)+(-2)(0-2)\right]\right| > 0.01$$

Frank-Wolfe Method

(3) Find the optimal solution α_0 of the one-dimensional minimization problem by the bisection search method:

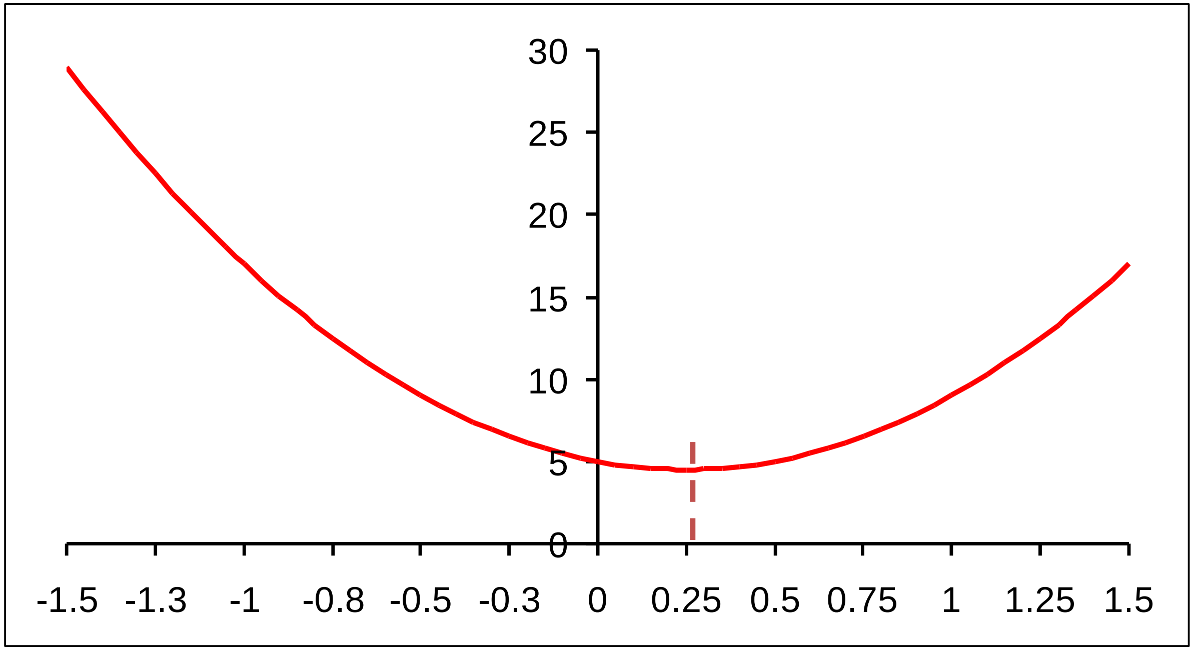
$$\min_{0\le\alpha\le1} f\left(x^{(1)}+\alpha\left(y^{(1)}-x^{(1)}\right)\right)=\left(2\alpha-2\right)^2+\left(-2\alpha-1\right)^2 \Rightarrow \alpha_1=0.25$$

(4) Update

$$x^{(2)}=x^{(1)}+\alpha_1\left(y^{(1)}-x^{(1)}\right)$$

$$=\begin{pmatrix}0+0.25\times(2\text{-}0)\\2+0.25\times(0\text{-}2)\end{pmatrix}$$

$$=\begin{pmatrix}0.5\\1.5\end{pmatrix}$$



Newton's Method (reading materials)

Newton's Method

$$\min_{x \in R^n} z(\mathbf{x})$$

❑Newton Method for multivariate functions

$$z(\mathbf{x}) \approx z(\mathbf{x}_k) + \nabla z(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_k)^T \nabla^2 z(\mathbf{x}_k)(x-\mathbf{x}_k)$$

- where

$$\nabla z(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial z}{\partial x_1} \\ \dfrac{\partial z}{\partial x_2} \\ \vdots \\ \dfrac{\partial z}{\partial x_n} \end{bmatrix} \qquad \nabla^2 z(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial^2 z}{\partial x_1^2} & \dfrac{\partial^2 z}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 z}{\partial x_1 \partial x_n} \\ \dfrac{\partial^2 z}{\partial x_2 \partial x_1} & \dfrac{\partial^2 z}{\partial x_2^2} & \cdots & \dfrac{\partial^2 z}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 z}{\partial x_n \partial x_1} & \dfrac{\partial^2 z}{\partial x_n \partial x_2} & \cdots & \dfrac{\partial^2 z}{\partial x_n^2} \end{bmatrix}_{n \times n}$$

Newton's Method

❑**Newton Method for multivariate functions**

$$z(\mathbf{x}) \approx z(\mathbf{x}_k) + \nabla z(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_k)^T \nabla^2 z(\mathbf{x}_k)(x-\mathbf{x}_k)$$

Then, it gives: $\nabla z(\mathbf{x}) \approx \nabla z(\mathbf{x}_k) + \nabla^2 z(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k)$

- The first order necessary optimality gives that

$$\nabla z(\mathbf{x}) = 0$$

$$\Rightarrow \quad \nabla z(\mathbf{x}_k) + \nabla^2 z(\mathbf{x}_k)(x-\mathbf{x}_k) = 0$$

- If $\nabla^2 z(\mathbf{x}_k)$ is nonsingular, then we have

$$x = \mathbf{x}_k - \frac{\nabla z(\mathbf{x}_k)}{\nabla^2 z(\mathbf{x}_k)}$$

Newton's Method (con't)

- **Algorithm**

Step 0: Choose an initial point $x^{(0)}$ and set $k=0$. Define $\varepsilon > 0$

Step 1: If $|\nabla z(x^{(k)})| < \varepsilon,$ then stop. Otherwise, go to Step 2

Step 2: Set $x^{(k+1)} = x^{(k)} - \left[\nabla^2 z\left(x^{(k)}\right)\right]^{-1} \nabla z\left(x^{(k)}\right),$

and let $k = k+1,$ go to Step 1.

- Pros
  - It is usually faster than gradient descent method. For quadratic optimization problems, it finds the optimal solution in one iteration.
- Cons
  - It may not converge to a KKT point.
  - The Hessian may be singular (not invertible).

Newton's Method (con't)

- **Example**

$$\min z(x) = (x_1 - 2)^4 + (x_1 - 2x_2)^2$$

$$\nabla z(x) = \left(4(x_1 - 2)^3 + 2x_1 - 4x_2, -4x_1 + 8x_2\right)$$

$$H(x) = \nabla^2 z(x) = \begin{pmatrix} 12(x_1 - 2)^2 + 2 & -4 \\ -4 & 8 \end{pmatrix}$$

- Use Newton method to solve the problem starting from point (0,3). Stop after 6 iterations.

Newton's Method (con't)

❑ Example

k	x_k $z(x^k)$	$\nabla z(x_k)$	$H(x_k)$	$H(x_k)^{-1}$	$-H(x^k)^{-1}\nabla z(x_k)$	x_{k+1}
0	(0.00, 3.00) 52.00	(-44.0, 24.0)	$\begin{pmatrix} 50.0 & -4.0 \\ -4.0 & 8.0 \end{pmatrix}$	$\frac{1}{384}\begin{pmatrix} 8.0 & 4.0 \\ 4.0 & 50.0 \end{pmatrix}$	(0.67, -2.67)	(0.67, 0.33)
1	(0.67, 0.33) 3.13	(-9.39,-0.04)	$\begin{pmatrix} 23.23 & -4.0 \\ -4.0 & 8.0 \end{pmatrix}$	$\frac{1}{169.84}\begin{pmatrix} 8.0 & 4.0 \\ 4.0 & 23.23 \end{pmatrix}$	(0.44, -0.23)	(1.11, 0.56)
2	(1.11, 0.56) 0.63	(-2.84, 0.04)	$\begin{pmatrix} 11.50 & -4.0 \\ -4.0 & 8.0 \end{pmatrix}$	$\frac{1}{76}\begin{pmatrix} 8.0 & 4.0 \\ 4.0 & 11.5 \end{pmatrix}$	(0.30, 0.14)	(1.41, 0.70)

Newton's Method (con't)

- Example

k	x_k $z(x^k)$	$\nabla z(x_k)$	$H(x_k)$	$H(x_k)^{-1}$	$-H(x^k)^{-1}\nabla z(x_k)$	x_{k+1}
3	(1.41, 0.70) 0.12	(-0.80, -0.04)	$\begin{pmatrix} 6.18 & -4.0 \\ -4.0 & 8.0 \end{pmatrix}$	$\frac{1}{33.44}\begin{pmatrix} 8.0 & 4.0 \\ 4.0 & 6.18 \end{pmatrix}$	(0.20, 0.10)	(1.61, 0.80)
4	(1.61, 0.80) 0.12	(-0.22,-0.04)	$\begin{pmatrix} 3.83 & -4.0 \\ -4.0 & 8.0 \end{pmatrix}$	$\frac{1}{14.64}\begin{pmatrix} 8.0 & 4.0 \\ 4.0 & 3.83 \end{pmatrix}$	(0.13, 0.07)	(1.74, 0.87)
5	(1.74, 0.87) 0.005	(-0.07, 0.00)	$\begin{pmatrix} 2.81 & -4.0 \\ -4.0 & 8.0 \end{pmatrix}$	$\frac{1}{6.48}\begin{pmatrix} 8.0 & 4.0 \\ 4.0 & 2.81 \end{pmatrix}$	(0.09, 0.04)	(1.83, 0.91)
6	(1.83, 0.91) 0.0009	(0.0003, -0.04)				

Algorithms for Multidimensional Unconstrained Minimization Problems

❑ Summary

➢ The feasible descent direction method

$$x_{k+1} = x_k + \alpha_k d_k \quad \text{with} \quad \nabla z(x_k)^T d_k < 0$$

where step size α_k can be obtained by a line search

➢ The steepest descent direction method: $d_k = -\nabla z(x_k)$

➢ Newton method: $d_k = -\left[\nabla^2 z(x_k)\right]^{-1} \nabla z(x_k)$ and $\alpha_k = 1$

➢ Other descent direction methods:

$d_k = -F_k \nabla z(x_k)$ where F_k is any positive definite matrix

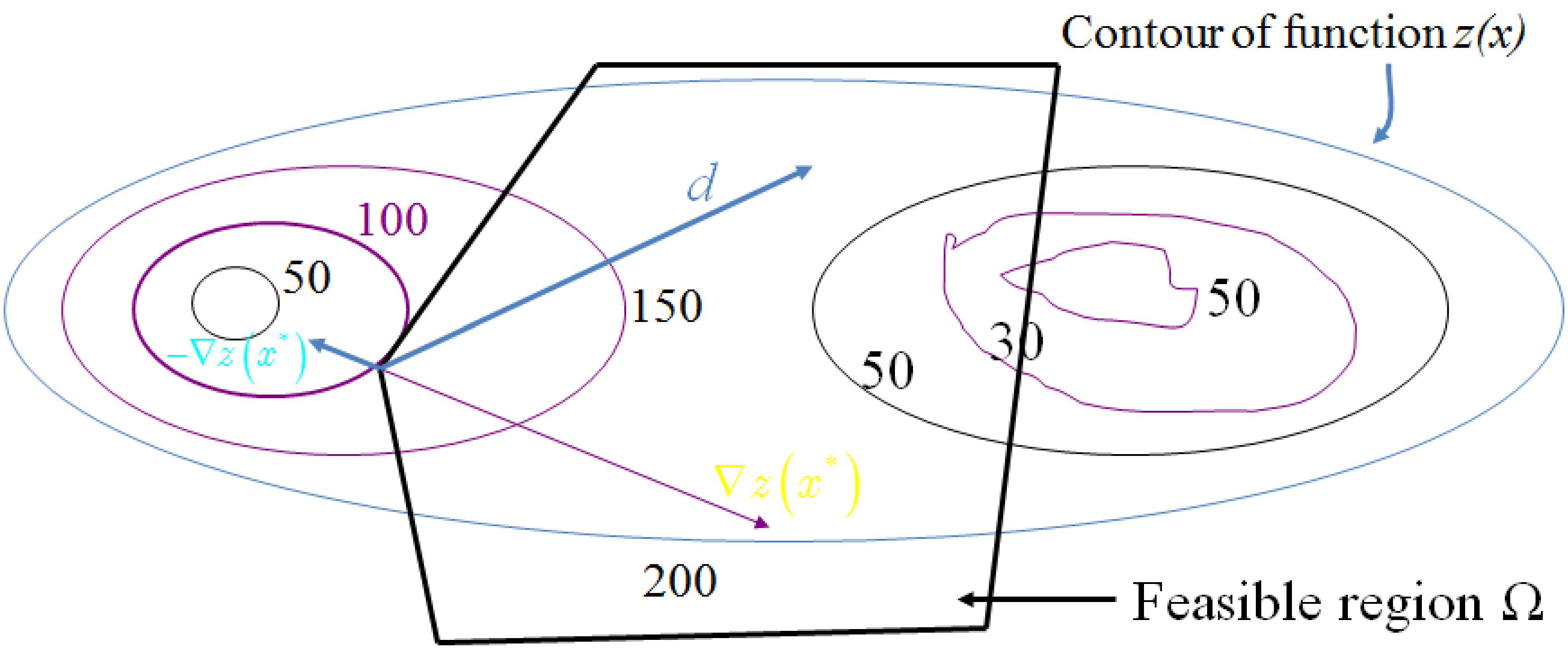
e.g. Quasi-Newton Method

Optimal Conditions for the Local Minimum
Necessary Condition vs. Sufficient Condition
(reading materials)

First-order Necessary Optimality Conditions

- **Constrained NLP:** $\min_{x\in\Omega} f(x)$

Given that $f(x)$ is a continuously differentiable function, if x^* is a local minimum, then any feasible direction d at x^* should satisfy the condition: $\nabla f(x^*)^T d \geq 0$



First-order Necessary Optimality Conditions (con't)

- **Proof by contradiction:**

Assume that the condition does not hold, i.e., there is a feasible direction *d* at *x** such that $\nabla f\left(x^*\right)^T d < 0$.

As *d* is a feasible direction at *x** and $\nabla f\left(x\right)$ is a continuous vector function, hence there is $\alpha_1 > 0$ such that

$$\nabla f\left(x^* + \alpha d\right)^T d < 0 \qquad \forall \alpha \in \left[0, \alpha_1\right]$$

Taking a very small but positive α in the interval $[0,\alpha_1]$, by Taylor's theorem, it follows that

$$f\left(x^* + \alpha d\right) = f\left(x^*\right) + \alpha \nabla f\left(x^* + \left(1-\theta\right)\alpha d\right)^T d$$

In other words, we have $f\left(x^* + \alpha\theta d\right) < f\left(x^*\right)$

It contradicts to the assumption that *x** is a local minimum

First-order Necessary Optimality Conditions (con't)

- **Unconstrained NLP:** $\min_{x \in R^n} f(x)$

Given that $f(x)$ is a continuously differentiable function, if x^* is a local minimum, then x^* must fulfill the condition:

$$\nabla f\left(x^*\right) = 0$$

Proof (using the first-order necessary condition for the constrained NLP):

First take an unit vector: $e_i = (0, \ldots, 1, 0, \ldots, 0)$.

As both e_i and $-e_i$ are the feasible direction of the unconstrained NLP, we have

$$\nabla z\left(x^*\right)^T e_i = \partial z\left(x^*\right) / \partial x_i \geq 0 \text{ and } \nabla z\left(x^*\right)^T \left(-e_i\right) = -\partial z\left(x^*\right) / \partial x_i \geq 0$$

$$\Longrightarrow \quad \partial z\left(x^*\right) / \partial x_i = 0, i = 1, \ldots, n$$

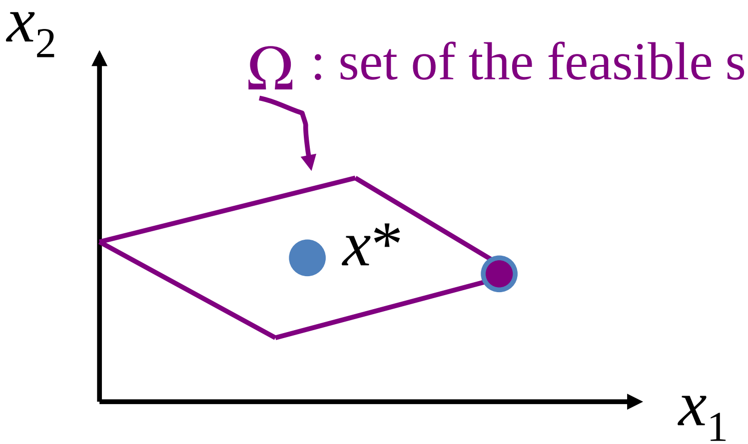
First-order Necessary Optimality Conditions (con't)

- **Corollary: For a <u>constrained</u> NLP**

Given that $f(x)$ is a continuously differentiable function, if x^* is a local minimum point and also an interior point in Ω, then we also have

$$\nabla f\left(x^*\right) = 0$$

$\nabla f\left(x^*\right)^T d \geq 0$ holds for any feasible direction d



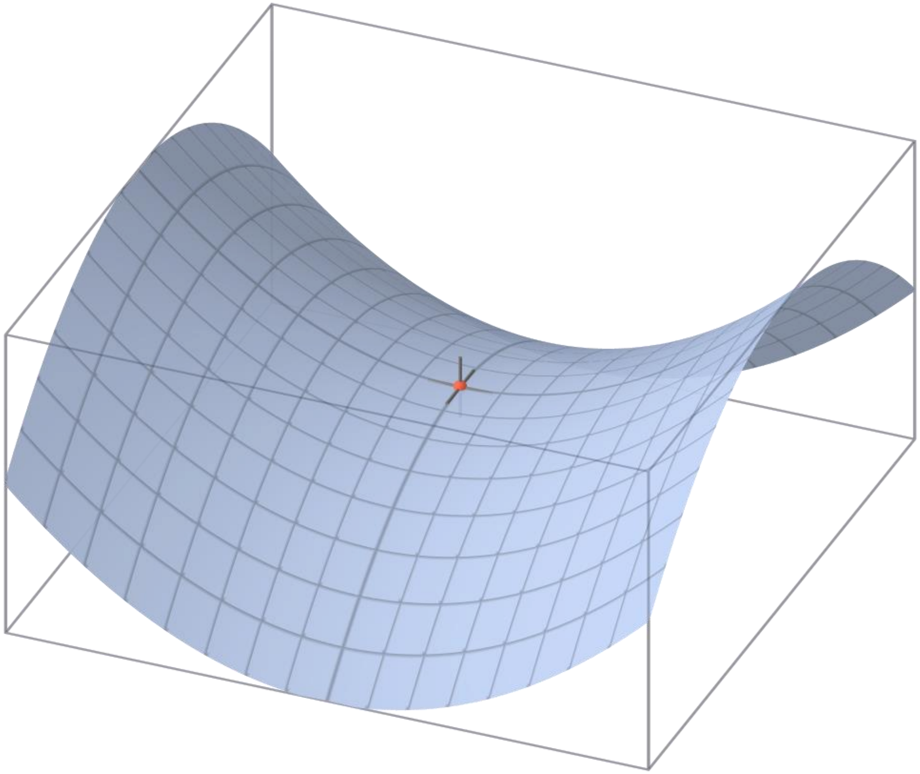
Second-order Necessary Optimality Conditions

- **Example**

At point *A*, we have

$$\nabla f\left(x\right) = 0$$

Point *A* is a saddle point rather than a local minimum



- **Conclusion**

The first-order partial derivatives of the objective functions are ***not sufficient*** to verify a local minimum.

Second-order Necessary Optimality Conditions (con't)

- **For a Constrained NLP:** $\min_{x \in \Omega \subseteq R^n} f(x)$

If $\nabla^2 f(x)$ is a continuous function on set Ω and x^* is a local minimum, then any feasible direction d at x^* should satisfy:

(i) $\nabla f(x^*)^T d \geq 0$

(ii) if $\nabla f(x^*)^T d = 0$, then $d^T \nabla^2 f(x^*) d \geq 0$

- **Unconstrained NLP:** $\min_{x \in R^n} f(x)$

If $\nabla^2 f(x)$ is a continuous function on R^n, and x^* is a local minimum, then we have:

(i) $\nabla f(x^*)^T = 0$

(ii) $\nabla^2 f(x^*)$ is positive semi-definite

Second-order Necessary Optimality Conditions (con’t)

- **Proof of the unconstrained NLP by contradiction**

Suppose the second-order necessary optimal condition does not hold, i.e., there is a direction *d* such that: $d^T \nabla^2 f\left(x^*\right) d < 0$

As $\nabla^2 f\left(x\right)$ is a continuous function, then there is a positive parameter $\alpha_1 > 0$ such that

$$d^T \nabla^2 f\left(x^* + \alpha d\right) d < 0, \forall \alpha \in \left[0, \alpha_1\right]$$

Taking a very small positive α in the interval $[0,\alpha_1]$, by Taylor’s theorem, it follows that

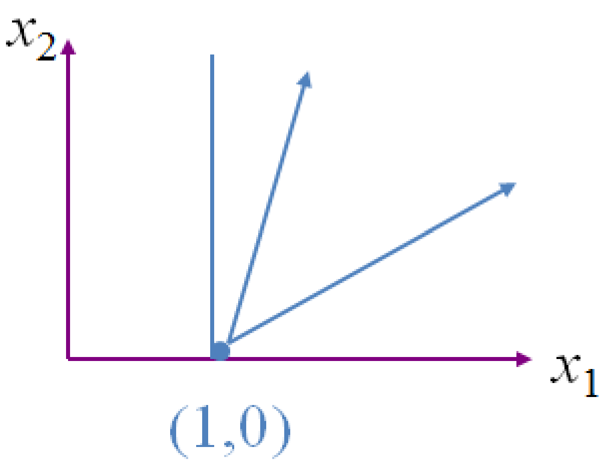
$$f\left(x^* + \alpha d\right) = f\left(x^*\right) + \alpha \nabla f\left(x^*\right)^T d + \frac{1}{2}\alpha^2 d^T \nabla^2 f\left(x^* + \theta \alpha d\right) d$$

Hence, $f\left(x^* + \alpha d\right) < f\left(x^*\right)$

It contradicts the fact that x^* is the optimum

❑ Example 1:

$$\min f(x) = x_1^2 + x_1 x_2 - x_1 - x_2,$$
$$\text{s.t. } x_1 \geq 1, x_2 \geq 0$$



Does the optimum $x^* = (1,0)$ satisfy

$\nabla f(x^*)^T d \geq 0$ for any feasible direction d?

❑ Example 2:

$$\min f(x) = x_1^2 + x_1 x_2 - x_1 - x_2$$

- Find a stationary point, that is, a point with zero gradient.
- Is the point a ***local minimum***?

Second-order Necessary Optimality Conditions (con't)

- **Example 1:**

$$\min f(x) = x_1^2 + x_1 x_2 - x_1 - x_2, \text{ s.t. } x_1 \geq 1, x_2 \geq 0$$

$$\left.\begin{array}{l} \text{Given } x_1^* = 1, x_2^* = 0 \Rightarrow \nabla f\left(x^*\right) = \left(1,0\right)^T \\ \text{any feasible direction at } \left(1,0\right), d = \left(d_1, d_2\right) \geq 0 \end{array}\right\} \Rightarrow \nabla f\left(x^*\right) d \geq 0$$

$$\begin{cases} if \ \nabla f\left(x^*\right)^T d = d_1 = 0 \\ then \ d^T \nabla^2 f\left(x^*\right)^T d = (d_1, d_2)\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = 2d_1^2 + 2d_1 d_2 = 0 \end{cases}$$

Hence, both first order and second order conditions are satisfied

Second-order Necessary Optimality Conditions (con't)

- **Example 2:**

Applying the second-order necessary optimality conditions to verify that (1,-1) is not a local minimum for the unconstrained minimization problem as follows:

$$\min f(x) = x_1^2 + x_1 x_2 - x_1 - x_2$$

Gradient function: $\nabla f(x) = (2x_1 + x_2 - 1, x_1 - 1)^T = (0,0)^T \Longrightarrow x_1 = 1,\ x_2 = -1$

So, (1, -1) is a stationary point.

Hessian matrix function: $\nabla^2 f(x) = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} \Longrightarrow \begin{vmatrix} 2 & 1 \\ 1 & 0 \end{vmatrix} = (2\times 0 - 1\times 1) = -1 < 0$

So, at (1, -1) the Hessian matrix is not positive definite, so it's not a local minimum.

Second-order Sufficient Optimality Conditions

Unconstrained minimization problem: $\min_{x \in R^n} z(x)$

Assume that $\nabla^2 z(x)$ is a continuous function over R^n. If a point x^* fulfills the following two conditions:

(i) $\nabla z(x^*) = 0$

(ii) $\nabla^2 z(x^*)$ is a positive definite matrix

Then, x^* is a local minimum of the unconstrained minimization problem