HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

Chapter 7: Unconstrained Minimization Methods

Vu Van Thieu, Dinh Viet Sang, Ban Ha Bang

SoICT
Hanoi University of Science and Technology

ONE LOVE. ONE FUTURE.

Introduction

1. Recall some concepts from calculus
2. Unconstrained nonlinear programming
3. Single-variable minimization methods
   - Single-extreme function
   - Fibonacci method
   - Golden section method
4. Unconstrained Optimization Methods
   - Gradient methods
   - Newton method

Recall some concepts from calculus

n-dimensional Euclidean space

Denote $\mathbb{R}^n$ - set of n-dimensional real vectors

$$R^n = \{x = (x_1, x_2, \cdots, x_n)^T : x_i \in \mathbb{R}, i = 1, 2, \cdots, n\}$$

where $\mathbb{R}$ is the set of real numbers. There we define the operations

- Addition of two vectors $u = (u_1, u_2, \cdots, u_n)^T$ abd $v = (v_1, v_2, \cdots, v_n)^T$

$$u + v = (u_1 + v_1, u_2 + v_2, \cdots, u_n + v_n)$$

- Vector multiplication with a real number α

$$\alpha u = (\alpha u_1, \alpha u_2, \cdots, \alpha u_n)^T$$

$\mathbb{R}^n$ and the operations just defined form a linear space. Elements of $\mathbb{R}^n$ are sometimes points.

n-dimensional Euclidean space (cont.)

- If we consider the concept of the scalar product of two vectors $u, v \in \mathbb{R}^n$:

$$< u, v >= \sum_{i=1}^{n} u_i \times v_i$$

then $\mathbb{R}^n$ together with the scalar product becomes an n-dimensional Euclidean space.

- The length of vector $u \in \mathbb{R}^n$ is the number

$$||u|| =< u, u >^{1/2}= \left(\sum_{i=1}^{n} u_i^2 \right)^{1/2}$$

n-dimensional Euclidean space (cont.)

- The distance between two points $u, v \in \mathbb{R}^n$

$$\rho(u, v) = ||u - v|| = \left(\sum_{i=1}^{n} (u_i - v_i)^2 \right)^{1/2}$$

- For $u, v, w \in \mathbb{R}^n$ we have the triangle inequality

$$||u - v|| \leq ||u - w|| + ||w - v||$$

n-dimensional Euclidean space (cont.)

- Assume $\{u^k, k = 1, 2, \cdots\}$ set of points in $\mathbb{R}^n$, that means $u^k \in \mathbb{R}^n$, $k = 1, 2, \cdots$, the point v is called the critical point of sequence $\{u^k\}$ if we could find a subsequencce $\{u^{k(i)}\}$ converges to v.
- Sequence $\{u^k\}$ is said to be bounded if there exists a constant $M \geq 0$ such that $||u^k|| \leq M$, for all $k = 1, 2, \cdots$
- Set $O(x, \epsilon) = \{u \in \mathbb{R}^n : ||u - x|| < \epsilon\}$ is the sphere centered at x and radius $\epsilon > 0$ is called the **neighbor** ϵ of x.
- The point $v \in \mathbb{R}^n$ is called the **critical point** of set $U \subset \mathbb{R}^n$, if all its neighbors ϵ always contains point of U that is different from v.

Recall some concepts from calculus

n-dimensional Euclidean space (cont.)

- Point $x \in X$ is said to be an **interior point** of set X if there exists a neighbor ϵ of which lies entirely in X. The set of interior points of X is denoted by $int(X)$.
- Point $x \in X$ is said to be a **boundary point** of set X if among all its neighbor ϵ there are points in X and not in X. The set of boundary points of X is denoted by $\partial(X)$.
- Set X is said to be the **open set** if every point $x \in X$ is an interior point of X.
- Set X in $\mathbb{R}^n$ is said to be **bounded**, if there exists a constant $L > 0$ such that $||u|| \leq L$ for all $u \in X$.

Recall some concepts from calculus

n-dimensional Euclidean space (cont.)

- Set X in $\mathbb{R}^n$ is said to be **closed set** if it contains all boundary points.
- Assume $\{x^k\}$ is a point in the closed set X and $\lim_{k\to+\infty} x = \overline{x}$, then $\overline{x} \in X$
- Set X is said to be **compact** if it is closed and bounded.
- Assume $\{x^k\}$ is a point in the compact set X. Then from $\{x^k\}$ we can always we can always extract the convergent subsequence $\{x^{k(i)}\}$ such that $\lim_{k(i)\to+\infty} x^{k(i)} = \overline{x}$, then $\overline{x} \in X$

n-dimensional Euclidean space

Differential of the multivariable function

Definition 1 : Assume function f is dertermined at neighbor $O(x, \epsilon)$ of the point x. We say that the function f is differentiable at x if there exists a vector $f'(x) \in \mathbb{R}^n$ such that the increment of the function at x : $\Delta f(x) = f(x + \Delta x) - f(x), ||\Delta x|| \leq \epsilon$ could be stated as following

$$\Delta f(x) =< f'(x), \Delta x > +o(x, \Delta x)$$

where $\lim_{||\Delta x|| \to 0} \frac{o(x,\Delta x)}{||\Delta x||} = 0$.
The function $f'(x)$ is called the gradient of the function f at x and denoted by $\nabla f(x)$.

Differential of the multivariable function (cont.)

Definition 2 : Assume function f is determined at neighbor $O(x, \epsilon)$ of the point x. We say the function We say that the function f is twice differentiable at x if along with vector $f'(x)$, there exists a symmetric matrix $f''(x) \in \mathbb{R}^{n \times n}$ such that the increment of function at x can be written as

$$\Delta f(x) = f(x + \Delta x) - f(x) = < f'(x), \Delta x > + \frac{< f''(x)\Delta x, \Delta x >}{2} + o(x, \Delta x)$$

where $\lim_{||\Delta x|| \to 0} \frac{o(x, \Delta x)}{||\Delta x||^2} = 0$.
Matrix $f''(x)$ is called second-order derivative matrix (a.k.a. the Hessian) of function f at x and is sometimes denoted by $\nabla^2 f(x)$

Differential of the multivariable function (cont.)

Definition 3 : Assume function f is determined on open set X. We say that the function f is continuously differentiable over the set X if f is differentiable at every point x of X and

$$||f'(x+\Delta x) - f'(x)|| \to 0 \text{ khi } ||\Delta x|| \to 0, \quad \forall x, x+\Delta x \in X$$

The set of functions satisfying this property is denoted by $C^1(X)$.

Definition 4 : Assume function f is determined on open set X. We say that the function f is twice continuously differentiable on set X if f is twice differentiable at every point x of X and

$$||f''(x+\Delta x) - f''(x)|| \to 0 \text{ khi } ||\Delta x|| \to 0, \quad \forall x, x+\Delta x \in X$$

The set of functions satisfying this property is denoted by $C^2(X)$.

Differential of the multivariable function (cont.)

Taylor's formula : Assume $f(x)$ is twice continuously differentiable at a neighbor ϵ of x^o, then

$$\begin{aligned} f(x) =& f(x^o) + < f'(x^o), x - x^o > \\ &+ \frac{1}{2} < f"(x^o)(x - x^o), x - x^o > + \alpha(x, x^o)||x - x^o||^2 \end{aligned}$$

where $\lim_{x \to x^o} \alpha(x, x^o) = 0$, the error can be written as $o(||x - x^o||^2)$

Differential of the multivariable function (cont.)

The finite-increments formula : Suppose function f is continuously differentiable on the open set S and x is a vector of S. Then for any vector y satisfying $x + y \in S$, one could always find a number $\alpha \in [0, 1]$ such that

$$f(x+y) - f(x) =< f'(x+\alpha y), y >= \int_0^1 < f'(x+ty), y > dt$$

if f is twice continuously differentiable, then we have:

$$f(x+y) - f(x) =< f'(x), y > + \frac{1}{2} < f''(x+\alpha y)y, y > .$$

Differential of the multivariable function

Extrema of Multivariable Functions

Considering optimization problem

$$f(x) \to \min, x \in X$$

where $X \subset \mathbb{R}^n$, and f is the function determined on X.
Defintion 5 :The point $x^* \in X$ is called **global minimum** point of f on X if $f(x^*) \leq f(x), \quad \forall x \in X$.

- Value $f(x^*)$ is the minimum value of f on X and we denote it by $\min\{f(x) : x \in X\}$
- The point $x^* \in X$ is called **local minimum** point of f on X if there exists neighbor $O(x, \epsilon), \epsilon > 0$ such that $f(x^*) \leq f(x)$, for $x \in O(x, \epsilon) \cap X$.

Extrema of Multivariable Functions (cont.)

Definition 6 : Suppose function f is bounded on X. Value f^* is called lower bound of f on X if

1. $f^* \leq f(x)$ for all $x \in X$
2. For every value $\epsilon > 0$, there always exists $u^\epsilon \in X$ such that $f(u^\epsilon) < f^* + \epsilon$

Then we denote: $\inf_{x \in X} f(x) = f^*$

Note

Obviously if function f has global minimum on X then

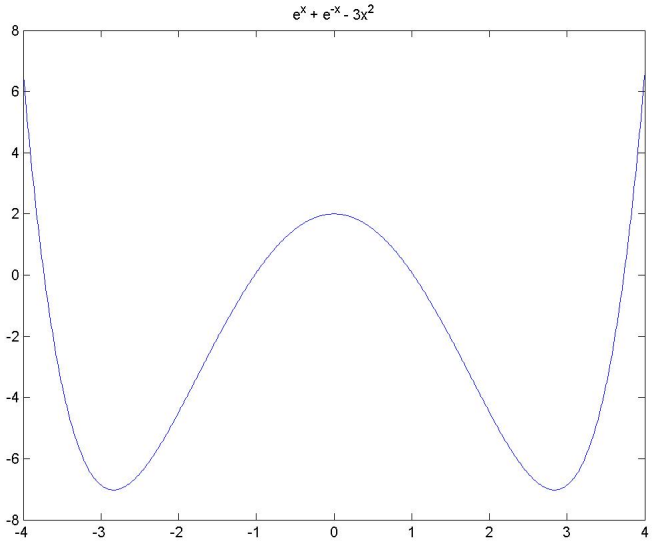
$$\inf_{x \in X} f(x) = \min_{x \in X} f(x)$$

Recall some concepts from calculus

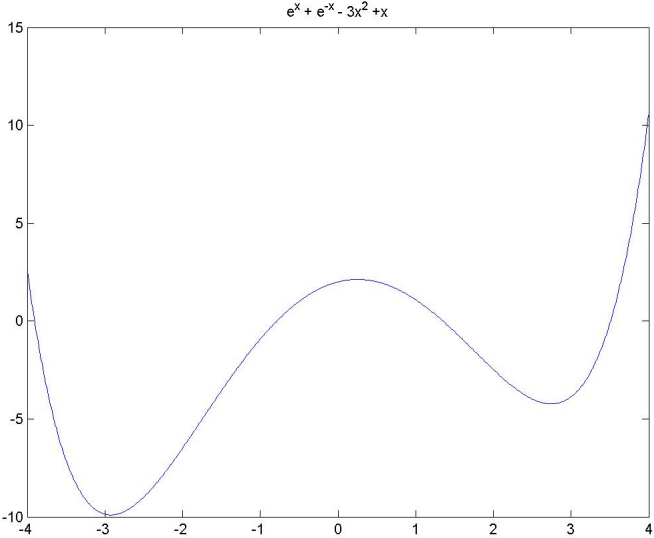
Some examples

- $f(x) = (x-1)^2$ has global minimum at $x^* = 1$ with $f(x^*) = 0$.
- $f(x) = e^x + e^{-x} - 3x^2$. The optimal value of function $f(x) = -7.02$. The problem has global minimum at two points $x = \pm 2.84$, has no local minimum.
- $f(x) = e^{-x}$ has a lower bound of zero but not achieved. There is no local minimum or global minimum.
- $f(x) = -x + e^{-x}$ The objective function is not bounded below, has neither a local minimum nor a global minimum.
- $f(x) = e^x + e^{-x} - 3x^2 + x$ The problem has local minimum $x_1 = -2.9226$ and $x_2 = 2.7418$, in which x_1 is global minimum point. The optimal value of this function is -9.9040

Recall some concepts from calculus



Recall some concepts from calculus



Definition

Consider the Unconstrained nonlinear programming:

$$\min\{f(x) : x \in \mathbb{R}^n\} \tag{1}$$

where $f(x)$ is continuously differentiable.

Theorems

Theorem 1 (Necessary condition for optimality) : The necessary condition for x^0 to be a local minimum is

$$\nabla f(x^0) = 0 \tag{2}$$

Condition (2) is called the stationary condition, the point x^0 satisfying (2) is called stationary point. Therefore, solving the problem (1) can be reduced to to solving equations (2).

Some examples

- $f(x) = x^2 - 3x - 1$: equation $f'(x) = 2x - 3 = 0$ has the only solution $x^0 = 3/2$ that is a local minimum and at the same time a global minimum.
- $f(x_1, x_2) = x_1^2 + x_2^2 - 2x_1x_2 + x_1$: equation $\nabla f(x) = \begin{pmatrix} 2x_1 - 2x_2 + 1 \\ -2x_1 + 2x_2 \end{pmatrix} = 0$ has the only solution $x^0 = (-1/4, 1/4)$. However, solution x^0 is not optimal solution of the problem $\min\{f(x) : x \in \mathbb{R}^2\}$ because $f(-1/4, 1/4) = -1/8 > -1 = f(0, 1)$.

Theorem (cont.)

Theorem 2 (Sufficient condition for optimality) : Suppose f is twice continuously differentiable. The stationary point x^0 is local minimum if matrix $f''(x^0)$ is a positive definite matrix.

To know whether a matrix is positive definite or not, the following Sylvester's criterion can be used.

Sylvester's criterion : Matrix $A = (a_{ij})_{n\times n}$ a negative definite matrix (positive semidefinite matrix) if and only if all of its sub-determinants are non-negative.

$$\triangle_{i_1,i_2,\cdots,i_k} = det \begin{vmatrix} a_{i_1,i_1} & a_{i_1,i_2} & \cdots & a_{i_1,i_k} \\ a_{i_2,i_1} & a_{i_2,i_2} & \cdots & a_{i_2,i_k} \\ \cdots & & & \\ a_{i_k,i_1} & a_{i_k,i_2} & \cdots & a_{i_k,i_k} \end{vmatrix} \geq 0$$

where $\forall 1 \leq i_1 < i_2 < \cdots < i_k \leq n, \forall k = 1, 2, \cdots, n$

Some examples

- Consider $f(x_1, x_2) = e^{x_1^2+x_2^2}$: solve equations

$$\nabla f(x) = \begin{pmatrix} 2x_1 e^{x_1^2+x_2^2} \\ 2x_2 e^{x_1^2+x_2^2} \end{pmatrix} = 0$$

has the only solution $x^0 = (0, 0)$ because $f''(0, 0) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$ has determinant $det|f''(x^0)| > 0$ so x^0 is a local minimum and at the same time the optimal solution of the problem.

Some examples (cont.)

- Consider $f(x_1, x_2) = -x_1^2 + x_2^2 - 2x_1x_2 - x_1$: solve equations

$$\nabla f(x) = \begin{pmatrix} -2x_1 - 2x_2 - 1 \\ -2x_1 + 2x_2 \end{pmatrix} = 0$$

has the only solution $x^0 = (-1/4, -1/4)$ because $f''(x^0) = \begin{pmatrix} -2 & -2 \\ -2 & 2 \end{pmatrix}$ has determinant $det|f''(x^0)| < 0$ so x^0 is not minimum of function $f(x)$.

Unimodal function

Unimodal function is a function with only one maximum or minimum point on the specified interval.

Definition : If x^* is the single minimum point of $f(x)$ in the range $a \leq x \leq b$ then $f(x)$ is unimodal on the interval if and only if for any two points x_1 and x_2,

- $x^* \leq x_1 \leq x_2$ implies that $f(x^*) \leq f(x_1) \leq f(x_2)$
- $x^* \geq x_1 \geq x_2$ implies that$f(x^*) \leq f(x_1) \leq f(x_2)$

Note

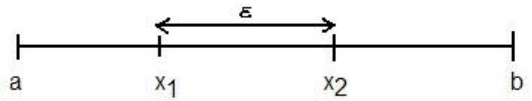
The search methods (to be introduced) all are worked for unimodal function.

Example

Assuming the interval contains minimum [0,1], we calculate the value of function at two points $x_1 < x_2 : f(x_1) = f_1, f(x_2) = f_2$, there are 3 possibilities:

1. $f_1 < f_2$: The minimum point x cannot be to the right of x_2, thus the interval containing the minimum is $[0, x_2]$
2. $f_1 > f_2$: The minimum point x cannot be to the left of x_1, thus the interval containing the minimum is $[x_1, 1]$
3. $f_1 = f_2$: Two intervales $[0, x_1)$ and $(x_2, 1]$ can be removed, and the intervalcontaining the minimum is $[x_1, x_2]$

Searching diagram



Suppose we have the initial interval $[a, b]$ containing the minimum

1. Calculate $x_1 = a + (b - a)/2 - e/2$ abd $x_2 = a + (b - a)/2 + e/2$ with e is the precision.
2. Calculate $f_1 = f(x_1)$ và $f_2 = f(x_2)$
3.
   - If $f_1 < f_2$ then setting $b = x_2$ (remove segment $x > x_2$);
   - If $f_1 > f_2$ then setting $a = x_1$ (remove segment $x < x_1$);
   - If $f_1 = f_2$ then setting $a = x_1$,$b = x_2$ (remove segment $x < x_1$ và $x > x_2$);
4. If $|b - a| < 2e$ then terminates; otherwise go back to step 1

Example :

Write the script to calculate the minimum of $f(x) = x(x - 1.5)$ on the interval $(a, b) = (0, 1)$ with the precision $e = 0.01$ compared to the absolute solution $x^* = 0.75$

```
f = inline('x.*(x-1.5)','x');
eps = 0.01;
a = 0; b = 1; k = 0;
while abs(b-a)>= 2*eps
        x1=a + (b-a)/2 - eps/2; x2=a + (b-a)/2 + eps/2;
        f1=f(x1); f2=f(x2); k=k+1;
        if f1<f2 b=x2;
                elseif f1>f2 a=x1;
                else a=x1;b=x2;
        end
end
...
```

Example (cont.) :

```
...
fprintf('Number of iterations k= %d ',k);
fprintf('Length of segment : b-a = %d',b-a);
fprintf('x= %d',x1);
```

.

Result

» Number of iterations k=7

Length of segment : b-a = 1.773438e-002

x=7.502344e-001

Fibonacci method

Definition : Fibonacci sequence is defined recursively as following

$$F_0 = 1; F_1 = 1;$$
$$F_k = F_{k-1} + F_{k-2}, k \geq 2;$$

We need to determine the number of iterations N before calculating the minimum. The selection of two points $x_1^{(k)}$ and $x_2^{(k)}$ at iteration k is determined by the formula :

$$x_1^{(k)} = \frac{F_{N-1-k}}{F_{N+1-k}}(b_k - a_k) + a_k, k = 0, 1, \cdots, N-1$$
$$x_2^{(k)} = \frac{F_{N-k}}{F_{N+1-k}}(b_k - a_k) + a_k, k = 0, 1, \cdots, N-1$$

Golden section method

To improve the Fibonacci method, without given number of iterations N, we always apply a fixed ratio when dividing the interval $b_k - a_k$

$$\lim_{N\to\infty} \frac{F_{N-1}}{F_{N+1}} = 0.382; \quad \lim_{N\to\infty} \frac{F_N}{F_{N+1}} = 0.618$$

Thus, the golden section method propose to choose two initial points $x_1^{(k)}$ và $x_2^{(k)}$ according to the updated formula at a iteration step as follows.

$$x_1^{(k)} = 0.382(b_k - a_k) + a_k,$$
$$x_2^{(k)} = 0.618(b_k - a_k) + a_k, \; k = 0, 1, 2, \cdots$$

Example

Implement the golden section method to find the minimum of function in the interval [0,2]

$$f(x) = x^2 - 2x + 1$$

```
f = inline('x^2-2*x+1');
a = 0; b = 2; eps = 0.00001;
x1 = a + (b-a)*0.382;
x2 = a + (b-a)*0.618;
f1 = f(x1);
f2 = f(x2);
while abs(b-a)>2*eps
        if f1 > f2
                a = x1; x1 = x2; f1 = f2;
                x2 = a + (b-a)*0.618;
                f2 = f(x2);...
```

Example (cont.)

```
            ...
        else
            b = x2; x2 = x1; f2 = f1;
            x1 = a + (b-a)*0.382;
            f1 = f(x1);
        end
end
fprintf('The interval containing the minimum : [%f,%f]',a,b);
»
The interval containing the minimum : [0.999990,1.000004]
```

Single-variable minimization methods

Introduction

Consider the unconstrained nonlinear programming

$$f(x) \to \min, x \in \mathbb{R}^n \tag{3}$$

where $f(x)$ is continuously differentiable. To solve (3), if the solution exist, it could be found among solutions of equations

$$\nabla f(x) = 0 \tag{4}$$

However, solving equations (4) in general case is is still quite complecate. This leads us to use efficient methods to solve (3).

Introduction (cont.)

A common approach to solving (3) is to use methods that iterate from an initial value x^0 then move 'toward' the optimal value x^*, at each iteration we calculate :

$$x^{k+1} = x^k + \alpha_k p^k, \quad k = 1, 2, \cdots \tag{5}$$

where

- p^k is the displacement vector from the point x^k.
- α_k is the length of the move in the direction p^k.

Obviously the procedure (5)is deterministic when we determine the direction p^k of the move and the way to calculate the length of the move α_k.

Introduction (cont.)

Depending on the different constructions of p^k and α^k we have iterative procedures with different properties.We are particularly interested in the following two properties: :

- The value variation of objective function f of sequence $\{x^k\}$
- The convergence of the sequence $\{x^k\}$ to solution x^*.

Also note that defining p^k and α_k differently also requires different amounts of computation.

Gradient methods

We select direction p^k such that

$$< \nabla f(x^k), p^k >< 0 \tag{6}$$

Because when select α^k small enough, we have

$$f(x^{k+1}) = f(x^k + \alpha_k) = f(x^k) + \alpha_k < \nabla f(x^k), p^k > + o(\alpha_k) < f(x^k)$$

that is, moving in the direction of p^k with a sufficiently small length, we will get to the point x^{k+1} with a smaller objective function value. So the direction p^k satisfying (6) is called the descent direction of the objective function $f(x)$.

Gradient methods (cont.)

One of the vectors satisfying the inequality (6) can be chosen as the gradient vector of the function f at x^k :

$$p_k = -\nabla f(x^k), k = 0, 1, 2, \cdots$$

Then we have an iterative procedure

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k), \alpha_k > 0, k = 0, 1, 2, \cdots \quad (7)$$

Iterative procedures that follow the formula (7) are called **gradient methods**.

Gradient methods (cont.)

Since the search direction is fixed, the gradient methods differ due to the choice of α_k. We list out some of the basic choices below

- Procedure : To select α_k we perform the following process
  1. Set $\alpha = \overline{\alpha} > 0$
  2. Set $u = x^k - \alpha \nabla f(x^k)$, calculate $f(u)$
  3. Check
  $$f(u) - f(x^k) \leq -\epsilon\alpha ||\nabla f(x^k)||^2, \text{ with } 0 < \epsilon < 1 \quad (8)$$
  4. If inequality (8) is satisfied then set $\alpha_k = \alpha$, otherwise set $\alpha = \alpha/2$ and go back to step 2.

Theorems with gradient methods

- Theorem 1 : Suppose $f(x)$ is bounded below and its gradient $f'(x)$ satisfies the Lipchitz condition :

$$||f'(x) - f'(y)|| \leq L||x - y||$$

for all $x, y \in \mathbb{R}^n$, the selection of α_k can be performed according to the procedure 2. Then following the iterative formula (7) will produce the sequence $\{x^k\}$ satisfying the condition

$$||f'(x)|| \to 0, k \to \infty$$

for all initial points x^0

Theorems with gradient methods (cont.)

Theorem 2 : Suppose function f is twice continuously differentiable and its Hessian matrix satisfies the condition

$$m||y||^2 \leq < H(x)y, y > \leq M||y||^2, M \geq m > 0$$

for all $x, y \in \mathbb{R}^n$, sequence $\{x^k\}$is constructed according to the iterative procedure (7) where α_k is determined according to procedure 2. Then for all initial points x^0 we have

$$x^k \to x^*, f(x^k) \to f(x^*), \text{ when } k \to \infty$$

where x^* is minimum point of $f(x)$, we have also the following:

$$f(x^k) - f(x^*) \leq q^k[f(x^0) - f(x^*)], \ ||x^k - x^*|| \leq Cq^{1/2}$$

where $0 < C < +\infty, 0 < q < 1$ are constants.

Example :

Consider the unconstrained minimization problem

$$f(x_1, x_2) = \frac{1}{2}(x_1^2 + \gamma x_2^2) \quad (\gamma > 0)$$

Perform the gradient method starting from the initial solution $x^0 = (\gamma, 1)$, the step length is determined according to the procedure 1, we get a sequence of approximate solutions $x^k = (x_1^k, x_2^k)$ where

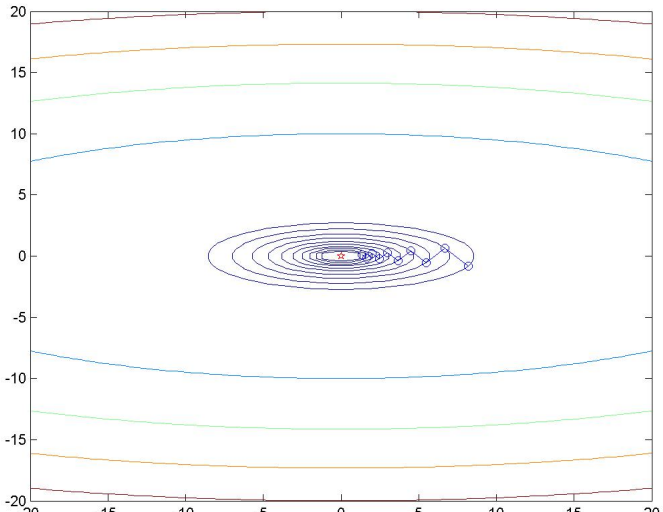
$$x_1^k = \gamma \left(\frac{\gamma - 1}{\gamma + 1}\right)^k, x_2^k = \left(-\frac{\gamma - 1}{\gamma + 1}\right)^k$$

The approximation sequence converges slowly to the optimal solution $x^* = (0, 0)$ when $\gamma >> 1$ and $\gamma << 1$

Gradient Descent Algorithm

Given stopping condition ϵ

1. Step 1: Given x^0 and set $k \leftarrow 0$
2. Step 2: $p^k = -\nabla f(x^k)$
   If $\|\nabla f(x^k)\|_2^2 < \epsilon$, then stop.
3. Step 3: Set $x^{k+1} = x^k + \alpha^k p^k, k \leftarrow k + 1$.
   Go to Step 1.


20
15
10
5
0
-5
-10
-15
-20
-20
-15
-10
-5
0
5
10
15
20

Newton method

In the case that the function f is twice continuously differentiable and the calculation of $f'(x)$ and $f''(x)$ is not difficult, we can use the quadratic term of the Taylor-series expansion.

$$f_k(x) \approx f(x^k) + < f'(x^k), x - x^k > + \frac{1}{2} < H(x^k)(x - x^k), x - x^k > \quad (9)$$

is a quadratic approximation of the function f at the neighbor of point x^k.

Newton method (cont.)

It is easy to see that when x^k is very close to x^* then $f(x^k)$ is very close to 0, and so the quadratic term of the (9) will give us more precise information about the variation of the function f in the neighbor of x^k.
We determine the approximation vector u^k from the condition

$$f_k(u^k) = \min f_k(x) \tag{10}$$

and build the next approximation solution

$$x^{k+1} = x^k + \alpha_k(u^k - x^k) \tag{11}$$

So depending on the options of choosing α_k, we have different methods. If $\alpha_k = 1$, for every k, we have Newton's method.

Newton method (cont.)

From (11) we have $x^{k+1} = u^k$, when choosing $\alpha_k = 1$, the condition (10) becomes

$$f_k(x^{k+1}) = \min f_k(x), k = 1, 2, \cdots \tag{12}$$

From (12) we infer that x^{k+1} is the breakpoint of the function $f_k(x)$, i,e,

$$\nabla f_k(x) = \nabla f(x^k) + H(x^k)(x - x^k) = 0$$

So if $H(x^k)$ is not degenerate, then we have the following Newton's formula

$$x^{k+1} = x^k - [H(x^k)]^{-1} \nabla f(x^k), k = 1, 2, \cdots \tag{13}$$

Newton method (cont.)

Convergence theorem : Suppose $H(x)$ is invertible, $[H(x)]^{-1}$ is bounded, $H(x)$ satisfies Liptchitz condition

$$||H(x) - H(y)|| \leq L||x - y||, \forall x, y \in \mathbb{R}^n$$

Then the sequence $\{x^k\}$ built on (13) will satisfy

$$||x^{k+1} - x^*|| \leq c||x^k - x^*||^2$$

where x^* is the solution to equation $f'(x) = 0$. So the convergence rate of Newton's method is the square of the distance after each iteration.

Newton Algorithm

Given stopping condition ϵ

1. Step 1: Given x^0 and set $k \leftarrow 0$
2. Step 2: $p^k = -H(x^k)^{-1} \nabla f(x^k)$
   If $p^k < \epsilon$, then stop.
3. Step 3: Set $x^{k+1} = x^k + p^k, k \leftarrow k + 1$.
   Go to Step 1.

Example

Illustrate Newton's method for the following objective function without constraints

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

in the algorithm there are functions $f(x), gf(x)$ and $hf(x)$.

```
f = inline('100*(x(2)-x(1)^2)^2 + (1-x(1))^1');
gf = inline('[100*2*(x(2)-x(1)^2)*(-2*x(1)) + 2*x(1)-2;200*(x(2)-x(1)^2)]');
hf = inline('[1200*x(1)^2 - 400*x(2)+ 2, -400*x(1);-400*x(1),200]');
```

Example (cont.)

```
fval = f(x);gval = gf(x);H = hf(x);ng = norm(gval);nf = 1;tol = 0.05;iter = 0; alpha
= 1;
while ng >= tol
  iter = iter + 1;nf = 0;alpha = 1;p = -inv(H)*gval;pass = 0;
  while pass == 0
    ftest = f(x+alpha*p);
    nf = nf+1;
    if ftest <= fval + 0.01*alpha*gval'*p
       pass = 1;
       x = x+alpha*p;
       fval = ftest;
       gval = gf(x);
       H = hf(x);
       ...
```

Example (cont.)

```
        ...
        ng = norm(gval);
    else
        alpha = alpha/2;
    end
  end
fprintf('%3i %3.2e %3.2e %3.2e %3.2e %i',iter,fval,ng,norm(x-xstart),alpha,nf);
end
```

Unconstrained Optimization Methods

iter	f	||g||	||x-x*||	alpha	nf
1	2.18e+000	4.64e+000	2.21e+000	1.00e+000	1
2	2.08e+000	1.10e+001	2.06e+000	6.25e-002	5
3	2.00e+000	1.51e+001	1.93e+000	2.50e-001	3
4	1.91e+000	1.63e+001	1.83e+000	5.00e-001	2
5	1.79e+000	1.68e+001	1.72e+000	1.00e+000	1
6	1.46e+000	7.79e+000	1.65e+000	1.00e+000	1
7	1.36e+000	8.04e+000	1.59e+000	5.00e-001	2
8	1.23e+000	8.23e+000	1.50e+000	1.00e+000	1
9	9.87e-001	3.70e+000	1.40e+000	1.00e+000	1
10	9.82e-001	1.06e+001	1.23e+000	1.00e+000	1
11	6.72e-001	1.17e+000	1.12e+000	1.00e+000	1
.....					
16	1.42e-001	4.59e+000	2.82e-001	1.00e+000	1
17	9.04e-002	4.41e-001	1.96e-001	1.00e+000	1
18	2.24e-002	2.13e+000	4.88e-002	1.00e+000	1

Searching methods

In practice the function f is not always smooth and twice differentiable. In addition, the calculation of these derivative requires large computations, so it is not suitable. So we need methods that only require the calculation of the function value.

$$f(x) \to \min, x \in \mathbb{R}^n \tag{14}$$

Denote

$$e^i = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

is a vector consisting of only zeros and ones in the ith row. Of course $e^i \in \mathbb{R}^n$, this is the unit

Algorithm

Suppose we have x^0 as a starting approximation. Choose $\alpha_0 > 0$ as a parameter of the algorithm. In iterations $k = 0, 1, 2, \cdots$ at approximations x^k and $\alpha_k > 0$.
Set

$$p^k = e^{i_k}, \quad i_k = k - \left[\frac{k}{n}\right] n + 1 \tag{15}$$

where $[k/n]$ is the largest integer not exceeding k/n.
Calculate $f(x^k + \alpha_k p^k)$ if

$$f(x^k + \alpha_k p^k) < f(x^k) \tag{16}$$

is satisfied then assign

$$x^{k+1} = x^k + \alpha_k p^k, \quad \alpha_{k+1} = \alpha_k$$

then go to step $k + 1$.

Algorithm (cont.)

If inequality (16) is not satisfied, then calculate $f(x^k - \alpha_k p^k)$ and check if the inequality

$$f(x^k - \alpha_k p^k) < f(x^k) \tag{17}$$

is satisfied then assign

$$x^{k+1} = x^k - \alpha_k p^k, \quad \alpha_{k+1} = \alpha_k$$

then go to step $k + 1$.
An iteration succeeds if either (16) and (17) are satisfied. Otherwise, we do the following

$$x^{k+1} = x^k,$$

Algorithm (cont.)

$$\alpha_{k+1} = \begin{cases} \lambda\alpha_k, \text{ if } i_k = n, x^k = x^{k-n+1} \\ \alpha_k, \text{ if } i_k \neq n, \text{ hay } x^k \neq x^{k-n+1} \end{cases}$$

where $0 < \lambda < 1$ is parameter of the algorithm.

- Formula (15) ensure a cyclical transition of the searching direction

$$p^0 = e^1, p^1 = e^2, \cdots, p^{n-1} = e^n, p^n = e^1, \cdots, p^{2n-1} = e^n, p^{2n} = e^1,$$

- The length $\alpha_{k+1} = \lambda\alpha_k$ needs to be changed according to the above formula only if after n consecutive cycles we have not succeeded in reducing the objective function value when satisfying either option (16) and (17).

Searching methods (cont.)

Convergence Theorem: Suppose f is a continuously differentiable convex function on $\mathbb{R}^n$, and x^0 is chosen such that the set

$$M(x^0) = \{x : f(x) \leq f(x^0)\}$$

is bounded. Then the sequence $\{x^k\}$ constructed by the method described above is the minimize sequence of the problem (14), that is

$$f(x^k) \to f^*, k \to \infty$$

where f^* is optimal value of the problem (14).

Exercise :
try to understand optimization functions in Matlab

- FMINBND
- FMINUNC
- FMINSEARCH
- OPTIMSET
- INLINE

Consider unconstraint minimization problem:

$$f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$$

Initial point $x^0 = (-1, 1)$

Consider the the unconstrained nonlinear programming:

$$f(x_1, x_2) = (3x_1 - 9)^2 + (4x_2 - 10)^2 \to min, x = (x_1, x_2) \in \mathbb{R}^2.$$

Implement gradient algorithm to solve the problem, starting from the initial solution $x^0 = (1, 2)$. To determine the step length use procedure 1 (which requires solving a problem of minimize the function of one variable).