#001108

中国科学技术大学 University of Science and Technology of China

数学实验

实验十二：非线性规划问题

翟晓雅

Email: xiaoyazhai@ustc.edu.cn

Homepage: https://xiaoyazhai.github.io/

实验目的

- 了解非线性规划问题

- 能利用MATLAB求解非线性规划问题

12.1 基本定义

如果目标函数或约束条件中包含**非线性函数**，就称这种规划问题为**非线性规划问题**。

非线性规划目前还没有适于各种问题的一般算法，各个方法都有自己特定的适用范围。

12.1 基本定义

线性规划与非线性规划的区别

如果线性规划的最优解存在，其最优解**只能在其可行域的边界上达到（特别是可行域的顶点上达到）**；

而非线性规划的最优解（如果最优解存在）则可能**在其可行域的任意一点达到**。

12.1 基本解法

非线性规划的 Matlab 解法

Matlab 中非线性规划的数学模型写成以下形式

$$\min f(x)$$

$$\begin{cases} Ax \le B \\ Aeq \cdot x = Beq \\ C(x) \le 0 \\ Ceq(x) = 0 \end{cases},$$

其中 $f(x)$ 是标量函数，A, B, Aeq, Beq 是相应维数的矩阵和向量，$C(x), Ceq(x)$ 是非线性向量函数。

X=FMINCON(FUN,X0,A,B,Aeq,Beq,LB,UB,NONLCON,OPTIONS)

12.1 基本解法

optimoptions

创建优化选项

语法

```
options = optimoptions(SolverName)
options = optimoptions(SolverName,Name,Value)

options = optimoptions(oldoptions,Name,Value)

options = optimoptions(SolverName,oldoptions)

options = optimoptions(prob)
options = optimoptions(prob,Name,Value)
```

```
options = optimoptions('fmincon')
```

```
options =
  fmincon options:

   Options used by current Algorithm ('interior-point'):
   (Other available algorithms: 'active-set', 'sqp', 'sqp-legacy', 'trust-region-reflective')

   Set properties:
     No options set.

   Default properties:
                    Algorithm: 'interior-point'
           BarrierParamUpdate: 'monotone'
               CheckGradients: 0
          ConstraintTolerance: 1.0000e-06
                      Display: 'final'
         EnableFeasibilityMode: 0
      FiniteDifferenceStepSize: 'sqrt(eps)'
         FiniteDifferenceType: 'forward'
         HessianApproximation: 'bfgs'
                   HessianFcn: []
           HessianMultiplyFcn: []
                  HonorBounds: 1
       MaxFunctionEvaluations: 3000
                MaxIterations: 1000
               ObjectiveLimit: -1.0000e+20
          OptimalityTolerance: 1.0000e-06
                    OutputFcn: []
                      PlotFcn: []
                 ScaleProblem: 0
    SpecifyConstraintGradient: 0
     SpecifyObjectiveGradient: 0
```

```
options = optimoptions(@fmincon,'Algorithm','interior-point','Display','off');
```

12.1 基本解法

求下列非线性规划

$$\min\ f(x) = x_1^2 + x_2^2 + x_3^2 + 8$$

$$\text{s.t.}\quad x_1^2 - x_2 + x_3^2 \ge 0$$

$$x_1 + x_2^2 + x_3^3 \le 20$$

$$-x_1 - x_2^2 + 2 = 0$$

$$x_2 + 2x_3^2 = 3$$

$$x_1, x_2, x_3 \ge 0$$

```
function f=fun1(x);
f=sum(x.^2)+8;
```

```
function [g,h]=fun2(x);
g=[-x(1)^2+x(2)-x(3)^2
   x(1)+x(2)^2+x(3)^3-20]; %非线性不等式约束
h=[-x(1)-x(2)^2+2
x(2)+2*x(3)^2-3]; %非线性等式约束
```

```
options=optimset('largescale','off');
[x,y]=fmincon('fun1',rand(3,1),[],[],[],[],zeros(3,1),[], ...
'fun2', options)
```

X=FMINCON(FUN,X0,A,B,Aeq,Beq,LB,UB,NONLCON,OPTIONS)

12.1 基本定义

一些基本概念

整体最优值

记（NP）的可行域为 K。

若 $x^* \in K$，并且

$$f(x^*) \le f(x), \quad \forall x \in K$$

则称 x^* 是（NP）的整体最优解，$f(x^*)$ 是(NP)的整体最优值。

严格整体最优值

$$f(x^*) < f(x), \quad \forall x \in K, x \ne x^*$$

则称 x^* 是（NP）的严格整体最优解，$f(x^*)$ 是(NP)的严格整体最优值。

局部最优值

若 $x^* \in K$，并且存在 x^* 的邻域 $N_\delta(x^*)$，使

$$f(x^*) \le f(x), \quad \forall x \in N_\delta(x^*) \cap K,$$

则称 x^* 是（NP）的局部最优解，$f(x^*)$ 是(NP)的局部最优值。

严格局部最优值

$$f(x^*) < f(x), \quad \forall x \in N_\delta(x^*) \cap K$$

则称 x^* 是（NP）的严格局部最优解，$f(x^*)$ 是(NP)的严格局部最优值。

12.1 基本定义

解的最优性问题

由于线性规划的目标函数为线性函数，可行域为凸集，因而求出的最优解就是整个可行域上的全局最优解。

非线性规划却不然，有时求出的某个解虽是一部分可行域上的极值点，但却并不一定是整个可行域上的全局最优解。

12.1 基本定义

(a) 迭代法求解最优解

$$x^0 \in R^n \Rightarrow \{x^k\} \Rightarrow \begin{cases} \text{有穷点列，最后一个点是(NP)的最优解} \\ \text{无穷点列，极限点是(NP)的最优解} \end{cases}$$

$$x^{\mathrm{k}} \in R^n \Rightarrow x^{\mathrm{k+1}} \in R^n \qquad (1)$$

$$\boxed{x^{k+1} = x^k + t_k p^k}$$

t_k迭代步长

p^k迭代方向

12.1 基本定义

求解关键：如何构造每一轮的搜索方向和确定适当的步长。

设 $\bar{x} \in R^n, p \neq 0$，若存在 $\delta > 0$，使

$$f(\bar{x} + tp) < f(\bar{x}), \forall t \in (0, \delta),$$

称向量 p 是 f 在点 $\bar{x}$ 处的下降方向。

设 $\bar{x} \in R^n, p \neq 0$，若存在 $t > 0$，使

$$\bar{x} + tp \in K,$$

称向量 p 是点 $\bar{x}$ 处关于 K 的可行方向。

一个向量 p，若既是函数 f 在点 x 处的下降方向，又是该点关于区域 K 的可行方向，称之为函数 f 在点 x 处关于 K 的**可行下降方向**。

12.1 基本定义

求解的一般步骤

0° 选取初始点 x^0，令 $k := 0$。

1° 构造搜索方向，依照一定规划，构造 f 在点 x^k 处关于 K 的可行下降方向作为搜索方向 p^k。

2° 寻求搜索步长。以 x^k 为起点沿搜索方向 p^k 寻求适当的步长 t_k，使目标函数值有某种意义的下降。

3° 求出下一个迭代点。按迭代格式（1）求出

$$x^{k+1} = x^k + t_k p^k 。$$

若 x^{k+1} 已满足某种终止条件，停止迭代。

4° 以 x^{k+1} 代替 x^k，回到 1°步。

12.2 凸函数、凸规划

凸函数

设 $f(x)$ 为定义在 n 维欧氏空间 $E^{(n)}$ 中某个凸集 R 上的函数，若对任何实数 $\alpha(0<\alpha<1)$ 以及 R 中的任意两点 $x^{(1)}$ 和 $x^{(2)}$，恒有

$$f(\alpha x^{(1)}+(1-\alpha)x^{(2)})\le \alpha f(x^{(1)})+(1-\alpha)f(x^{(2)})$$

则称 $f(x)$ 为定义在 R 上的凸函数。

严格凸函数

若对每一个 $\alpha(0<\alpha<1)$ 和 $x^{(1)}\neq x^{(2)}\in R$ 恒有

$$f(\alpha x^{(1)}+(1-\alpha)x^{(2)})< \alpha f(x^{(1)})+(1-\alpha)f(x^{(2)})$$

则称 $f(x)$ 为定义在 R 上的严格凸函数。

12.2 凸函数、凸规划

考虑非线性规划

$$\begin{cases} \min\limits_{x\in R} f(x) \\ R=\{x \mid g_j(x)\le 0,\ j=1,2,\cdots,l\} \end{cases}$$

假定其中 $f(x)$ 为凸函数，$g_j(x)(j=1,2,\cdots,l)$ 为凸函数

凸规划的可行域为凸集，其局部最优解即为全局最优解，而且其最优解的集合形成一个凸集。当凸规划的目标函数 $f(x)$ 为严格凸函数时，其最优解必定唯一（假定最优解存在）。

12.3无约束问题

(1) 一维搜索方法

(a) 试探法（“成功—失败”，斐波那契法，0.618 法等）；

(b) 插值法（抛物线插值法，三次插值法等）；

(c) 微积分中的求根法（切线法，二分法等）；

12.3无约束问题

(1) 一维搜索方法-Fibonacci 法

考虑一维极小化问题,若 $f(t)$ 是$[a,b]$ 区间上的下单峰函数

$$\min_{a\le t\le b} f(t) \tag{2}$$

通过两个搜索点处目标函数值大小的比较，总可以获得缩短了的单峰区间。

斐波那契法使用对称搜索的方法，逐步缩短所考察的区间，它能以尽量少的函数求值次数，达到预定的某一缩短率。

12.3无约束问题

(1) 一维搜索方法-Fibonacci 法

$$\min_{a \le t \le b} f(t)$$

如用 F_n 表示计算 n 个函数值能缩短为单位长区间的最大原区间长度，可推出 F_n 满足关系

$$F_0 = F_1 = 1$$

$$F_n = F_{n-2} + F_{n-1}, \quad n = 2,3,\cdots,$$

数列 $\{F_n\}$ 称为 Fibonacci 数列，F_n 称为第 n 个 Fibonacci 数，相邻两个 Fibonacci 数之比 $\dfrac{F_{n-1}}{F_n}$ 称为 Fibonacci 分数。

12.3无约束问题

(1) 一维搜索方法-Fibonacci 法

当用斐波那契法以 n 个探索点来缩短某一区间时，区间长度的第一次缩短率为 $\frac{F_{n-1}}{F_n}$，其后各次分别为 $\frac{F_{n-2}}{F_{n-1}},\frac{F_{n-3}}{F_{n-2}},\cdots,\frac{F_1}{F_2}$。由此，若 t_1 和 $t_2(t_2<t_1)$ 是单峰区间 $[a,b]$ 中第 1 个和第 2 个探索点的话，那么应有比例关系

$$\frac{t_1-a}{b-a}=\frac{F_{n-1}}{F_n},\qquad \frac{t_2-a}{b-a}=\frac{F_{n-2}}{F_n}$$

$$t_1=a+\frac{F_{n-1}}{F_n}(b-a),\ t_2=a+\frac{F_{n-2}}{F_n}(b-a) \qquad (3)$$

它们关于 $[a,b]$ 确是对称的点。

12.3 无约束问题

(1) 一维搜索方法-Fibonacci 法

如果要求经过一系列探索点搜索之后，使最后的探索点和最优解之间的距离不超过精度δ > 0，这就要求最后区间的长度不超过δ，即

$$\frac{b-a}{F_n} \le \delta \tag{4}$$

应按照预先给定的精度δ，确定使上式成立的最小整数 n 作为搜索次数，直到进行到第 n 个探索点时停止。

12.3无约束问题

(1) 一维搜索方法-Fibonacci 法：具体步骤

1° 选取初始数据，确定单峰区间$[a_0,b_0]$，给出搜索精度$\delta>0$，由（4）确定搜索次数n。

2° $k=1,a=a_0,b=b_0$，计算最初两个搜索点，按（3）计算t_1和t_2。

3° while $k<n-1$

$$f_1=f(t_1),f_2=f(t_2)$$

if $f_1<f_2$

$$a=t_2;t_2=t_1;t_1=a+\frac{F(n-1-k)}{F(n-k)}(b-a)$$

else

$$b=t_1;t_1=t_2;t_2=b+\frac{F(n-1-k)}{F(n-k)}(a-b)$$

end

$k=k+1$

end

12.3无约束问题

(1) 一维搜索方法-Fibonacci 法：具体步骤

4° 当进行至$k=n-1$时，

$$t_1=t_2=\frac{1}{2}(a+b)$$

这就无法借比较函数值$f(t_1)$和$f(t_2)$的大小确定最终区间，为此，取

$$\begin{cases} t_2=\dfrac{1}{2}(a+b) \\ t_1=a+(\dfrac{1}{2}+\varepsilon)(b-a) \end{cases}$$

其中ε为任意小的数。在t_1和t_2这两点中，以函数值较小者为近似极小点，相应的函数值为近似极小值。并得最终区间$[a,t_1]$或$[t_2,b]$。

12.3无约束问题

练习:

试用斐波那契法求函数 $f(t) = t^2 - t + 2$ 的近似极小点，

要求缩短后的区间不大于区间[-1,3] 的 0.08 倍。

12.3无约束问题

(1) 一维搜索方法-0.618 法

若ω > 0， 满足比例关系

$$\frac{\omega}{1}=\frac{1-\omega}{\omega}$$

称之为黄金分割数，其值为 $\omega=\frac{\sqrt{5}-1}{2}=0.6180339887\cdots$。

黄金分割数 ω 和 Fibonacci 分数之间有着重要的关系

$$\omega=\lim_{n\to\infty}\frac{F_{n-1}}{F_n}。$$

用不变的区间缩短率 0.618，代替斐波那契法每次不同的缩短率

12.3无约束问题

(1) 一维搜索方法-0.618 法

用 0.618 法求解，从第 2 个探索点开始每增加一个探索点作一轮迭代以后，原单峰区间要缩短 0.618 倍。计算 n 个探索点的函数值可以把原区间 $[a_0, b_0]$ 连续缩短 $n-1$ 次，因为每次的缩短率均为 μ，故最后的区间长度为

$$(b_0 - a_0)\mu^{n-1}$$

这就是说，当已知缩短的相对精度为 δ 时，可用下式计算探索点个数 n：

$$\mu^{n-1} \le \delta$$

当然，也可以不预先计算探索点的数目 n，而在计算过程中逐次加以判断，看是否已满足了提出的精度要求。

0.618 法是一种等速对称进行试探的方法，每次的探索点均取在区间长度的 0.618 倍和 0.382 倍处。

12.3无约束问题

（2）二次插值法

$$\min_{a \le t \le b} f(t)$$

当 $f(t)$ 在$[a, b]$ 上连续时，可以考虑用多项式插值来进行一维搜索。

基本思想是：在搜索区间中，不断用低次（通常不超过三次）多项式来近似目标函数，并逐步用插值多项式的极小点来逼近（2）的最优解。

12.3无约束问题

(3) 无约束极值问题的解法

无约束极值问题可表示为：

$$\begin{aligned} &\min \ f(x) \\ &x \in E^{(n)} \end{aligned} \qquad (5)$$

解析法：用到函数的一阶导数或二阶导数

直接法：仅用到函数值

12.3无约束问题

(a) 解析法-梯度法（最速下降法）

对基本迭代格式

$$x^{k+1}=x^k+t_k p^k \qquad (6)$$

我们总是考虑从点 x^k 出发沿哪一个方向 p^k，使目标函数 f 下降得最快。微积分的知识告诉我们，点 x^k 的负梯度方向

$$p^k=-\nabla f(x^k),$$

是从点 x^k 出发使 f 下降最快的方向。为此，称负梯度方向 $-\nabla f(x^k)$ 为 f 在点 x^k 处的最速下降方向。

每轮的搜索方向都是目标函数在当前点下降最快的方向！

12.2无约束问题

(a) 解析法-梯度法（最速下降法）

其具体步骤如下：

1° 选取初始数据。选取初始点 x^0，给定终止误差，令 $k:=0$。

2° 求梯度向量。计算 $\nabla f(x^k)$，若 $\|\nabla f(x^k)\| \le \varepsilon$，停止迭代，输出 x^k。否则，进行 3° 。

3° 构造负梯度方向。取

$$p^k = -\nabla f(x^k).$$

4° 进行一维搜索。求 t_k，使得

$$f(x^k + t_k p^k) = \min_{t \ge 0} f(x^k + t p^k)$$

令 $x^{k+1} = x^k + t_k p^k, k := k+1$, 转 2° 。

12.2无约束问题

例：用最速下降法求解无约束非线性规划问题

$$\min f(x) = x_1^2 + 25x_2^2$$

其中 $x = (x_1, x_2)^T$，要求选取初始点 $x^0 = (2,2)^T$

```
function [f,df]=detaf(x);
f=x(1)^2+25*x(2)^2;
df=[2*x(1)
50*x(2)];
```

```
clc
x=[2;2];
[f0,g]=detaf(x);
while norm(g)>0.000001
p=-g/norm(g);
t=1.0;f=detaf(x+t*p);
while f>f0
t=t/2;
f=detaf(x+t*p);
end
x=x+t*p;
[f0,g]=detaf(x);
end
x,f0
```

12.2无约束问题

(a) 解析法-Newton法

考虑目标函数 f 在点 x^k 处的二次逼近式

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

假定海森阵正定。

$$\nabla^2 f(x^k) = \begin{bmatrix} \dfrac{\partial^2 f(x^k)}{\partial x_1^2} & \cdots & \dfrac{\partial^2 f(x^k)}{\partial x_1 \partial x_n} \\ \vdots & \cdots & \vdots \\ \dfrac{\partial^2 f(x^k)}{\partial x_n \partial x_1} & \cdots & \dfrac{\partial^2 f(x^k)}{\partial x_n^2} \end{bmatrix}$$

12.2无约束问题

(a) 解析法-Newton法

由于$\nabla^2 f(x^k)$正定，函数Q的驻点x^{k+1}是$Q(x)$的极小点。为求此极小点，令

$$\nabla Q(x^{k+1}) = \nabla f(x^k) + \nabla^2 f(x^k)(x^{k+1} - x^k) = 0,$$

即可解得

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$

对照基本迭代格式（1），可知从点x^k出发沿搜索方向。

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

并取步长$t_k = 1$即可得$Q(x)$的最小点x^{k+1}。通常，把方向p^k叫做从点x^k出发的Newton方向。从一初始点开始，每一轮从当前迭代点出发，沿Newton方向并取步长为1的求解方法，称之为Newton法。

12.2无约束问题

(a) 解析法-Newton法：具体步骤

1° 选取初始数据。选取初始点x^0，给定终止误差$\varepsilon > 0$，令$k := 0$。

2° 求梯度向量。计算$\nabla f(x^k)$，若$\|\nabla f(x^k)\| \le \varepsilon$，停止迭代，输出$x^k$。否则，进行3°。

3° 构造 Newton 方向。计算$[\nabla^2 f(x^k)]^{-1}$，取

$$p^k = -[\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$

4° 求下一迭代点。令$x^{k+1} = x^k + p^k, k := k+1$，转2°。

12.2无约束问题

例： 用Newton法求解：

$$\min f(x) = x_1^4 + 25x_2^4 + x_1^2 x_2^2$$

选取 $x^0 = (2,2)^T$ 。

解：（i） $\nabla f(x) = [4x_1^3 + 2x_1 x_2^2 \quad 100x_2^3 + 2x_1^2 x_2]^T$

$$\nabla^2 f = \begin{bmatrix} 12x_1^2 + 2x_2^2 & 4x_1 x_2 \\ 4x_1 x_2 & 300x_2^2 + 2x_1^2 \end{bmatrix}$$

```
function [f,df,d2f]=nwfun(x);
f=x(1)^4+25*x(2)^4+x(1)^2*x(2)^2;
df=[4*x(1)^3+2*x(1)*x(2)^2;100*x(2)^3+2*x(1)^2*x(2)];
d2f=[2*x(1)^2+2*x(2)^2,4*x(1)*x(2)
4*x(1)*x(2),300*x(2)^2+2*x(1)^2];
```

```
clc
x=[2;2];
[f0,g1,g2]=nwfun(x);
while norm(g1)>0.00001
p=-inv(g2)*g1;
x=x+p;
[f0,g1,g2]=nwfun(x);
end
x, f0
```

12.2无约束问题

如果目标函数是非二次函数，一般地说，用 Newton 法通过有限轮迭代并不能保证可求得其最优解。

为了提高计算精度，我们在迭代时可以采用变步长计算上述问题：

```
clc,clear
x=[2;2];
[f0,g1,g2]=nwfun(x);
while norm(g1)>0.00001
p=-inv(g2)*g1;p=p/norm(p);
t=1.0;f=nwfun(x+t*p);
while f>f0
t=t/2;f=nwfun(x+t*p);
end
x=x+t*p;
[f0,g1,g2]=nwfun(x);
end
x,f0
```

Newton 法的优点是收敛速度快；缺点是有时不好用而需采取改进措施，此外，当维数较高时，计算 $-\left[\nabla^2 f\left(x^k\right)\right]^{-1}$的工作量很大。

12.2无约束问题

(a) 解析法-变尺度法（DFP法）

变尺度法（ Variable Metric Algorithm）是近 20 多年来发展起来的，它不仅是求解无约束极值问题非常有效的算法，而且也已被推广用来求解约束极值问题。

既避免了**计算二阶导数矩阵及其求逆过程**，又比梯度法的**收敛速度快**，特别是对**高维问题**具有显著的优越性，因而使变尺度法获得了很高的声誉。

12.2 无约束问题

(a) 解析法-变尺度法（DFP法）

我们已经知道，牛顿法的搜索方向是 $-[\nabla^2 f(x^k)]^{-1}\nabla f(x^k)$ ，为了不计算二阶导数矩阵 $[\nabla^2 f(x^k)]$ 及其逆阵，我们设法构造另一个矩阵，用它来逼近二阶导数矩阵的逆阵 $[\nabla^2 f(x^k)]^{-1}$，这一类方法也称拟牛顿法（Quasi-Newton Method）。

12.2无约束问题

(a) 解析法-变尺度法（DFP法）

1° 给定初始点 x^0 及梯度允许误差 $\varepsilon > 0$。

2° 若 $\|\nabla f(x^0)\| \le \varepsilon$，则 x^0 即为近似极小点，停止迭代，否则，转向下一步。

3° 令

$$\overline{H}^{(0)} = I \text{（单位矩阵）},$$

$$p^0 = -\overline{H}^{(0)} \nabla f(x^0)$$

在 p^0 方向进行一维搜索，确定最佳步长 λ_0：

$$\min_{\lambda} f(x^0 + \lambda p^0) = f(x^0 + \lambda_0 p^0)$$

如此可得下一个近似点

$$x^1 = x^0 + \lambda_0 p^0$$

4° 一般地，设已得到近似点 x^k，算出 $\nabla f(x^k)$，若

$$\|\nabla f(x^0)\| \le \varepsilon$$

则 x^k 即为所求的近似解，停止迭代；否则，计算 $\overline{H}^{(k)}$：

$$\boxed{\overline{H}^{(k)} = \overline{H}^{(k-1)} + \frac{\Delta x^{k-1} (\Delta x^{k-1})^T}{(\Delta G^{(k-1)})^T \Delta x^{k-1}} - \frac{\overline{H}^{(k-1)} \Delta G^{(k-1)} (G^{(k-1)})^T \Delta H^{(k-1)}}{(\Delta G^{(k-1)})^T \overline{H}^{(k-1)} \Delta G^{(k-1)}}} \Rightarrow \text{尺度矩阵}$$

并令 $p^k = -\overline{H}^{(k)} \nabla f(x^k)$，在 p^k 方向上进行一维搜索，得 λ_k，从而可得下一个近似点

$$x^{k+1} = x^k + \lambda_k p^k$$

5° 若 x^{k+1} 满足精度要求，则 x^{k+1} 即为所求的近似解，否则，转回 4°，直到求出某点满足精度要求为止。

Newton迭代矩阵

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

12.2无约束问题

(b) 直接法

在无约束非线性规划方法中，遇到问题的目标函数不可导或导函数的解析式难以表示时，人们一般需要使用直接搜索方法。同时，由于这些方法一般都比较直观和易于理解，因而在实际应用中常为人们所采用。

Powell 方法：由基本搜索、加速搜索和调整搜索方向三部分组成。

12.2无约束问题

(b) 直接法

1° 选取初始数据。选取初始点 x^0， n 个线性无关初始方向，组成初搜索方向组 $\{p^0, p^1, \cdots, p^{n-1}\}$。给定终止误差 $\varepsilon > 0$，令 $k := 0$。

2° 进行基本搜索。令 $y^0 := x^k$，依次沿 $\{p^0, p^1, \cdots, p^{n-1}\}$ 中的方向进行一维搜索。对应地得到辅助迭代点 $y^1, y^2, \cdots, y^n$，即

$$y^j = y^{j-1} + t_{j-1} p^{j-1}$$

$$f(y^{j-1} + t_{j-1} p^{j-1}) = \min_{t \geq 0} f(y^{j-1} + t p^{j-1}), \quad j = 1, \cdots, n$$

12.2无约束问题

(b) 直接法

3° 构造加速方向。令 $p^n = y^n - y^0$，若 $\|p^n\| \le \varepsilon$，停止迭代，输出 $x^{k+1} = y^n$。否则进行 4° 。

4° 确定调整方向。按下式

$$f(y^{m-1}) - f(y^m) = \max\{f(y^{j-1}) - f(y^j) \mid 1 \le j \le n\}$$

找出 m。若

$$f(y^0) - 2f(y^n) + f(2y^n - y^0) < 2[f(y^{m-1}) - f(y^m)]$$

成立，进行 5° 。否则，进行 6° 。

5° 调整搜索方向组。令

$$x^{k+1} = y^n + t_n p^n : f(y^n + t_n p^n) = \min_{t \ge 0} f(y^n + tp^n).$$

同时，令

$$\{p^0, p^1, \cdots, p^{n+1}\}_{k+1} := \{p^0, \cdots, p^{m-1}, p^{m+1}, \cdots, p^{n-1}, p^n\},$$

$k := k+1$，转 2° 。

6° 不调整搜索方向组。令 $x^{k+1} := y^n, k := k+1$，转 2° 。

12.2无约束问题

MATLAB
fminunc/ fminsearch

MATLAB求无约束极值问题

Matlab 中 fminunc 的基本命令是

[X,FVAL]=FMINUNC(FUN,X0,OPTIONS,P1,P2, ...)

其中的返回值 X 是所求得的极小点，FVAL 是函数的极小值，其它返回值的含义参见相关的帮助。FUN 是一个 M 文件，当 FUN 只有一个返回值时，它的返回值是函数 $f(x)$；当 FUN 有两个返回值时，它的第二个返回值是 $f(x)$ 的梯度向量；当 FUN 有三个返回值时，它的第三个返回值是 $f(x)$ 的二阶导数阵（Hessian 阵）。X0 是向量 x 的初始值，OPTIONS 是优化参数，可以使用缺省参数。P1，P2 是可以传递给 FUN 的一些参数。

12.2无约束问题

例： 求函数 $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ 的最小值。

```
function [f,g]=fun2(x);
f=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
g=[-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1));200*(x(2)-x(1)^2)];
```

```
options = optimset('GradObj','on');
[x,y]=fminunc('fun2',rand(1,2),options)
```

作业12.1

- 移动最小二乘法并未对转换矩阵M进行条件限制，如果添加其他限制条件后，能得到不同形式的转换矩阵M，文章根据不同的转换矩阵M提出了三种变形方法：

仿射变形（Affine Deformation）

相似变形（Similarity Deformation）

刚性变形（Rigid Deformation）

复现论文Image Deformation Using Moving Least Squares，用移动最小二乘方法实现图像的变形

12.3 约束极值问题

求解约束极值问题要比求解无约束极值问题困难得多。

为了简化其优化工作，可采用以下方法：将约束问题化为无约束问题；将非线性规划问题化为线性规划问题，以及能将复杂问题变换为较简单问题的其它方法。

12.3 约束极值问题

（1）二次规划

若某非线性规划的目标函数为自变量 x 的二次函数，约束条件又全是线性的

$$\min \ \frac{1}{2}x^T Hx + f^T x,$$

$$\text{s.t.} \quad \begin{cases} Ax \le b \\ Aeq \cdot x = beq \end{cases}$$

这里 H 是实对称矩阵，f, b 是列向量，A 是相应维数的矩阵。

[X,FVAL]=QUADPROG(H,f,A,b,Aeq,beq,LB,UB,X0,OPTIONS)

12.3 约束极值问题

（1）二次规划

$$\begin{cases} \min f(x) = 2x_1{}^2 - 4x_1x_2 + 4x_2{}^2 - 6x_1 - 3x_2 \\ x_1 + x_2 \le 3 \\ 4x_1 + x_2 \le 9 \\ x_1, x_2 \ge 0 \end{cases}$$

```
h=[4,-4;-4,8];
f=[-6;-3];
a=[1,1;4,1];
b=[3;9];
[x,value]=quadprog(h,f,a,b,[],[],zeros(2,1))
```

$$x = \begin{bmatrix} 1.9500 \\ 1.0500 \end{bmatrix},\ \mathrm{Min}\ f(x) = -11.0250$$

12.3 约束极值问题

(2) 罚函数法

利用罚函数法，可将非线性规划问题的求解，转化为求解一系列无约束极值问题，因而也称这种方法为序列无约束最小化技术，简记为 SUMT (Sequential Unconstrained Minization Technique)

利用问题中的约束函数作出适当的罚函数，由此构造出带参数的增广目标函数，把问题转化为无约束非线性规划问题。

- 外罚函数法
- 内罚函数法

12.3 约束极值问题

(2) 罚函数法

$$\min f(x)$$

$$\text{s.t.}\ \begin{cases} g_i(x) \le 0,\ i = 1,\cdots,r, \\ h_j(x) \ge 0,\ j = 1,\cdots,\text{s}, \\ k_m(x) = 0,\ m = 1,\cdots,t \end{cases}$$

取一个充分大的数 $M > 0$ ，构造函数

$$P(x,M) = f(x) + M\sum_{i=1}^{r}\max(g_i(x),0) - M\sum_{i=1}^{s}\min(h_i(x),0) + M\sum_{i=1}^{t}|k_i(x)|$$

（或 $P(x,M) = f(x) + M\text{sum}\left(\max\begin{pmatrix} G(x) \\ 0 \end{pmatrix}\right) - M\text{sum}\left(\min\begin{pmatrix} H(x) \\ 0 \end{pmatrix}\right) + M\,\|K(x)\|$

这里 $G(x) = [g_1(x) \ \cdots \ g_r(x)]$，$H(x) = [h_1(x) \ \cdots \ h_s(x)]$，$K(x) = [k_1(x) \ \cdots \ k_t(t)]$，Matlab 中可以直接利用 max 、min 和 sum 函数。）则以增广目标函数 $P(x,M)$ 为目标函数的无约束极值问题

$$\min P(x,M)$$

的最优解 x 也是原问题的最优解。

12.3 约束极值问题

求下列非线性规划

$$\begin{cases} \min f(x) = x_1^2 + x_2^2 + 8 \\ x_1^2 - x_2 \geq 0 \\ -x_1 - x_2^2 + 2 = 0 \\ x_1, x_2 \geq 0. \end{cases}$$

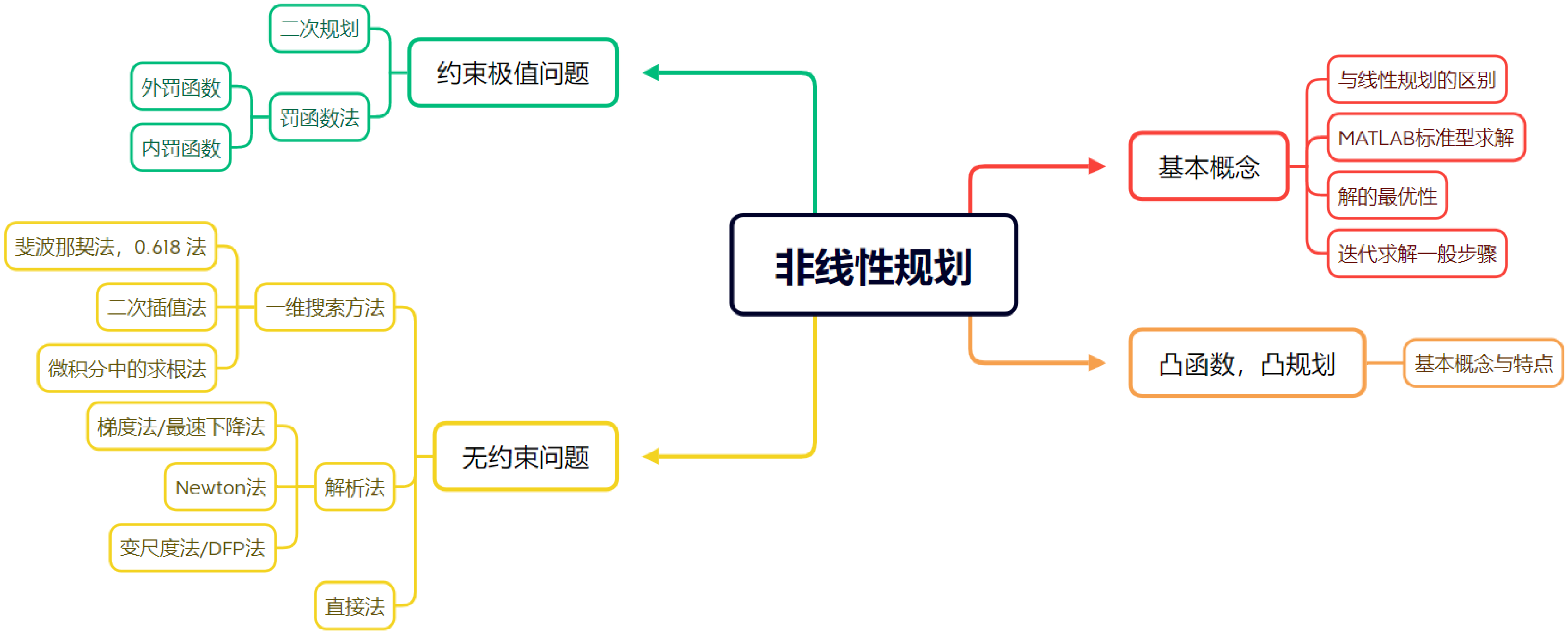
[x,y]=fminunc('test',rand(2,1))

```
function g=test(x);
M=50000;
f=x(1)^2+x(2)^2+8;
g=f-M*min(x(1),0)-M*min(x(2),0)-M*min(x(1)^2-x(2),0)+...
M*abs(-x(1)-x(2)^2+2);
```

```
function g=test(x);
M=50000;
f=x(1)^2+x(2)^2+8;
g=f-M*sum(min([x';zeros(1,2)]))-M*min(x(1)^2-x(2),0)+...
M*abs(-x(1)-x(2)^2+2);
```

```
function g=test(x);
M=50000;
f=x(1)^2+x(2)^2+8;
g=f-M*min(min(x),0)-M*min(x(1)^2-x(2),0)+M*(-x(1)-...
x(2)^2+2)^2;
```

总结



非线性规划
约束极值问题
二次规划
罚函数法
外罚函数
内罚函数
无约束问题
一维搜索方法
斐波那契法，0.618 法
二次插值法
微积分中的求根法
解析法
梯度法/最速下降法
Newton法
变尺度法/DFP法
直接法
基本概念
与线性规划的区别
MATLAB标准型求解
解的最优性
迭代求解一般步骤
凸函数，凸规划
基本概念与特点

作业12.1

- 移动最小二乘法并未对转换矩阵M进行条件限制，如果添加其他限制条件后，能得到不同形式的转换矩阵M，文章根据不同的转换矩阵M提出了三种变形方法：

仿射变形（Affine Deformation）

相似变形（Similarity Deformation）

刚性变形（Rigid Deformation）

复现论文Image Deformation Using Moving Least Squares，用移动最小二乘方法实现图像的变形

Q&A?

下节课内容

实验十三：现代优化算法

翟晓雅

Email: xiaoyazhai@ustc.edu.cn

Homepage: https://xiaoyazhai.github.io/

Lab: http://gcl.ustc.edu.cn/