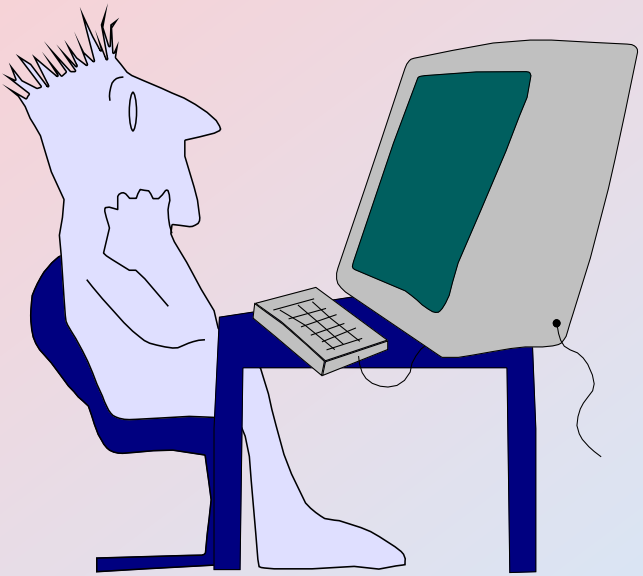
数学建模与数学实验

无约束最优化



实验目的

1、了解无约束最优化基本算法。

2、掌握用数学软件包求解无约束最优化问题。

实验内容

1、无约束优化基本思想及基本算法。

2、MATLAB优化工具箱简介

3、用MATLAB求解无约束优化问题。

4、实验作业。

无约束最优化问题

求解无约束最优化问题的的基本思想

*无约束最优化问题的基本算法

返回

求解无约束最优化问题的基本思想

标准形式：

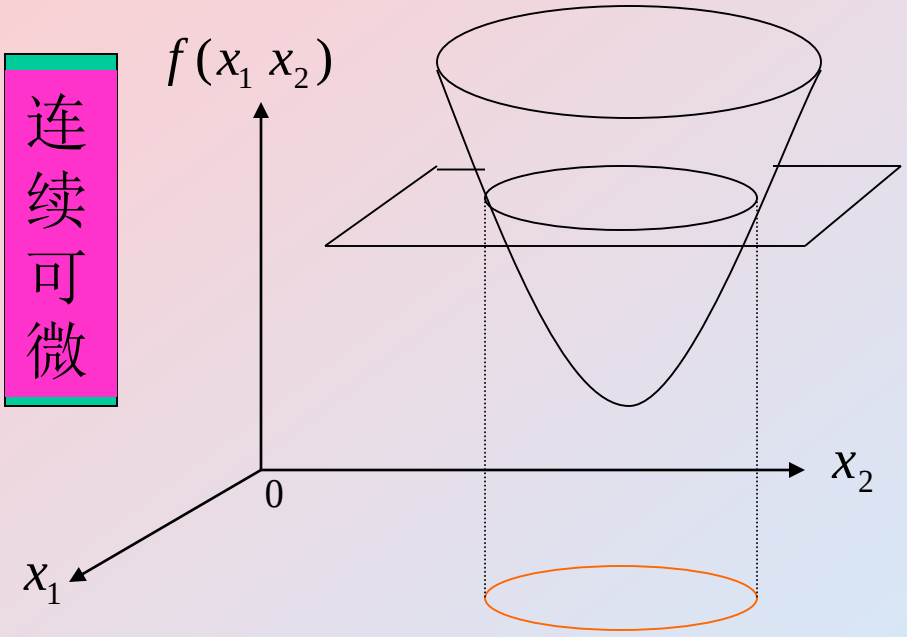
$$\min_{X \in E^n} f(X)$$

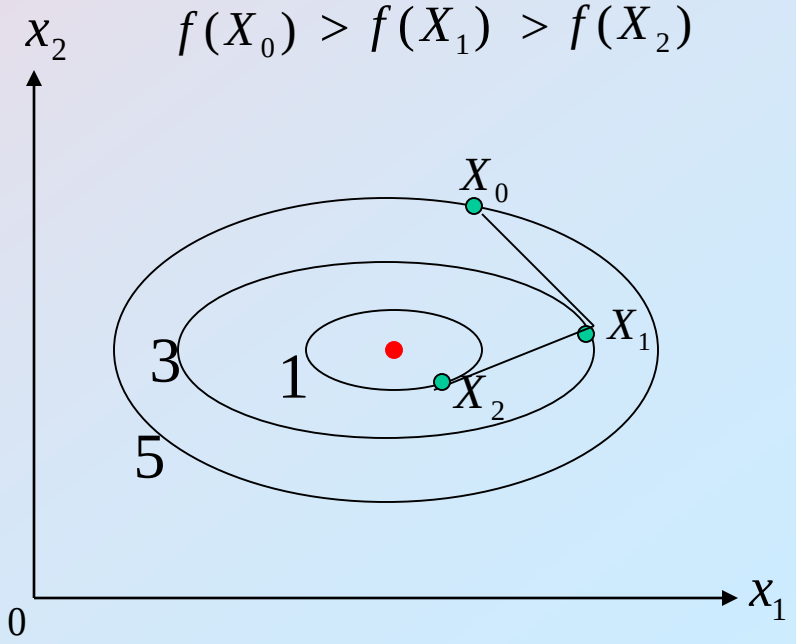
其中 $f : E^n \to E^1$

$$\max_{X \in E^n} f(X) = \min_{X \in E^n} [-f(X)]$$

求解的基本思想 （以二元函数为例）

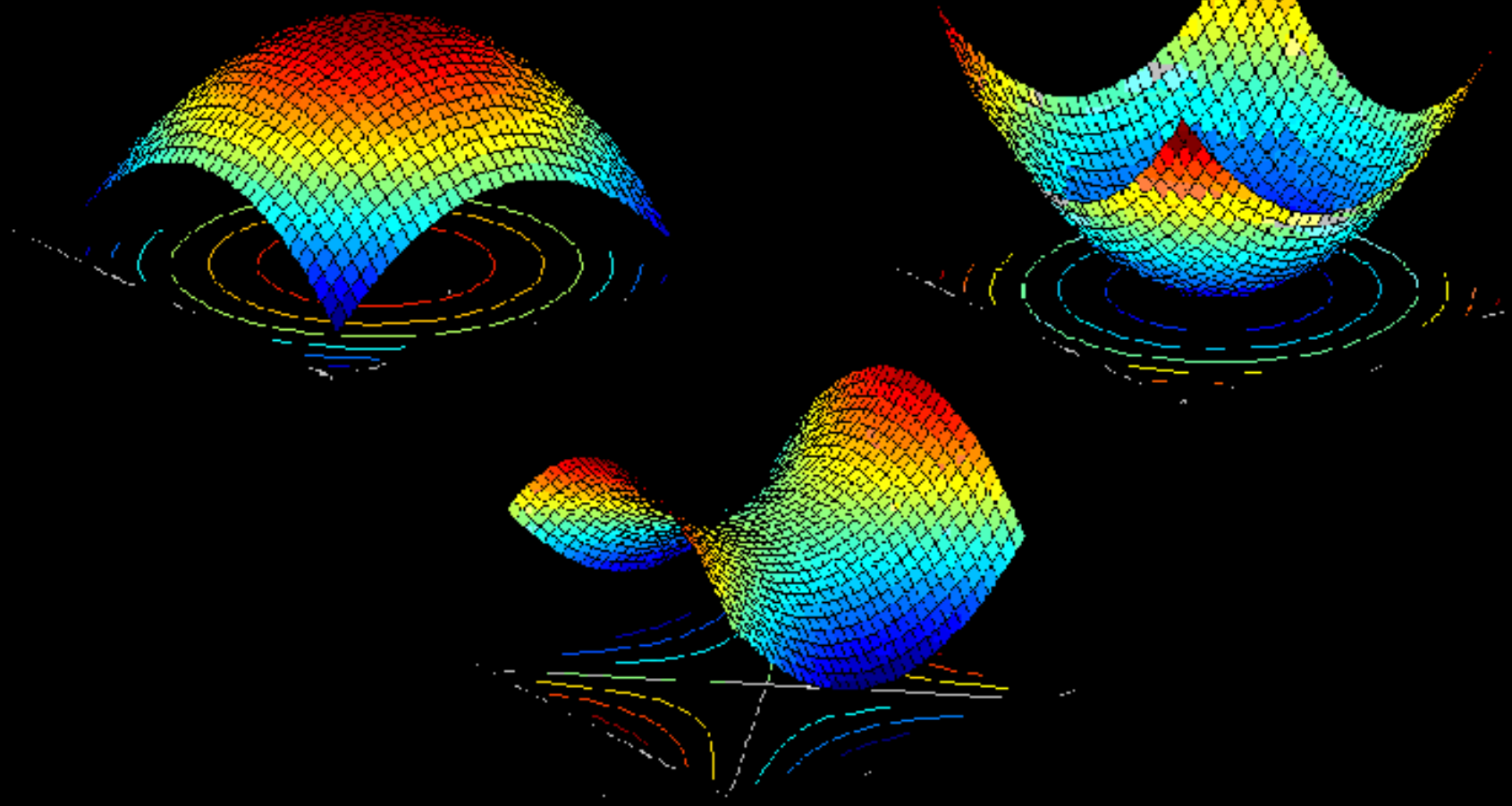
连续可微



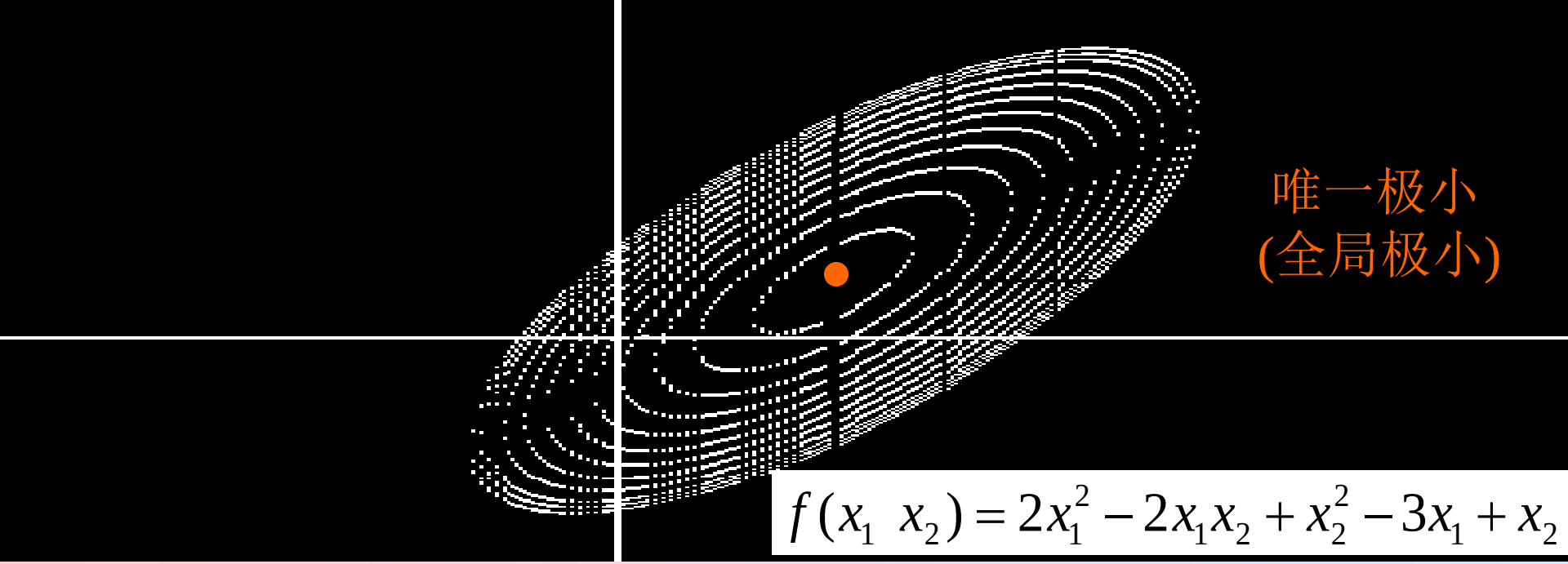


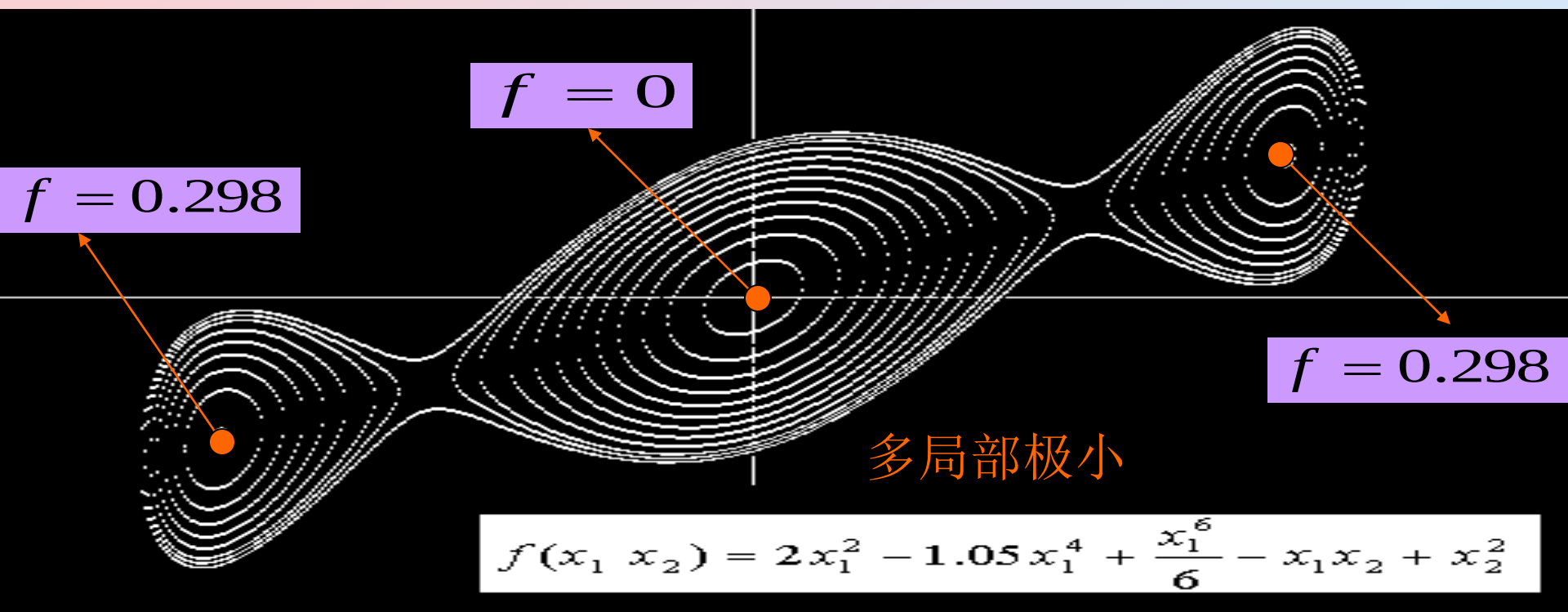
$f(x_1 \quad x_2) = -x_1^2 - x_2^2$

$f(x_1 \quad x_2) = x_1^2 + x_2^2$



$f(x_1 \quad x_2) = x_1^2 - x_2^2$


唯一极小
(全局极小)
$f(x_1\ x_2) = 2x_1^2 - 2x_1x_2 + x_2^2 - 3x_1 + x_2$


$f = 0$
$f = 0.298$
$f = 0.298$
多局部极小
$f(x_1\ x_2) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} - x_1x_2 + x_2^2$

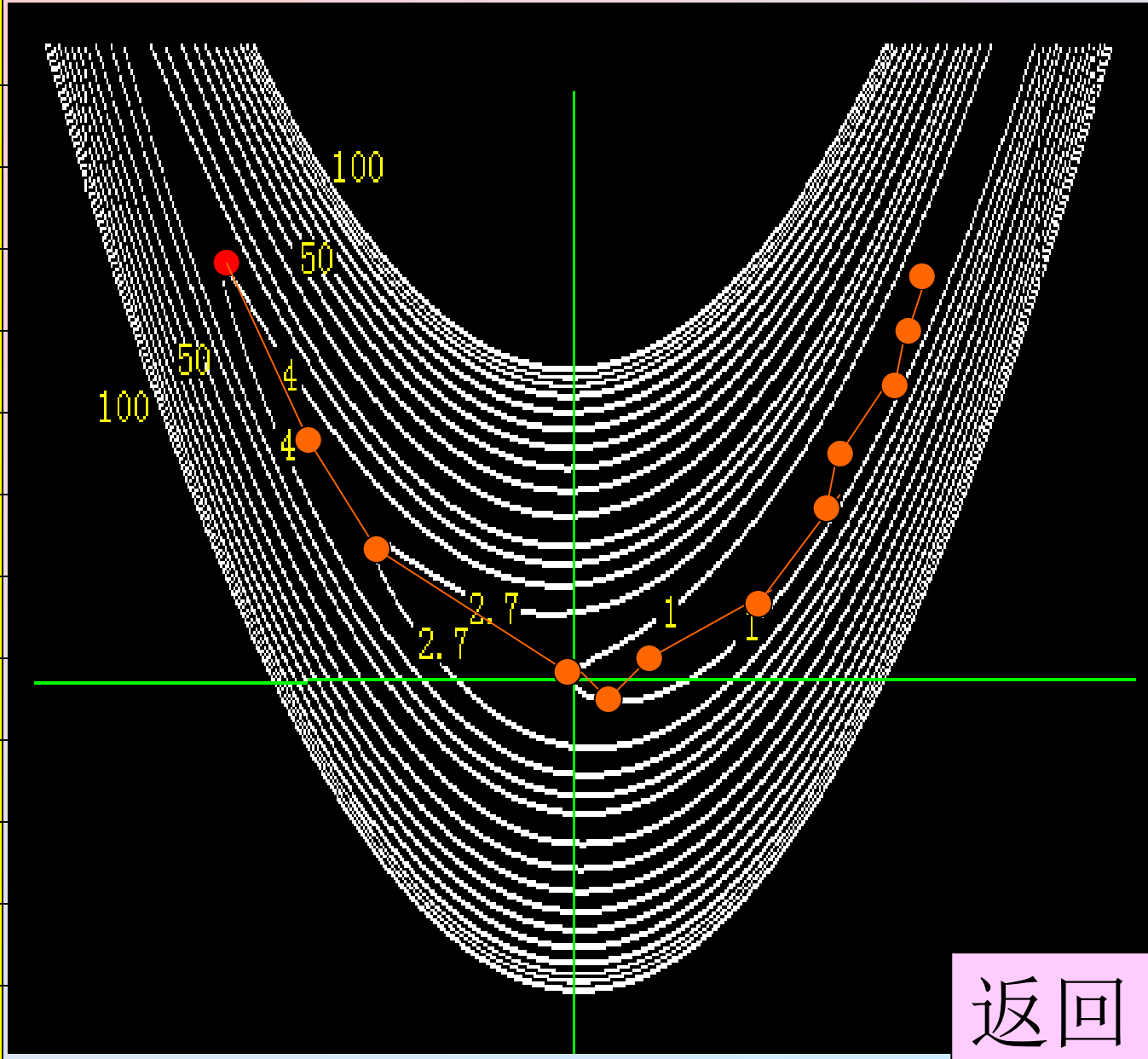
搜索过程

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

最优点 (1 1)

初始点 (-1 1)

x_1	x_2	f
-1	1	4.00
-0.79	0.58	3.39
-0.53	0.23	2.60
-0.18	0.00	1.50
0.09	-0.03	0.98
0.37	0.11	0.47
0.59	0.33	0.20
0.80	0.63	0.05
0.95	0.90	0.003
0.99	0.99	1E-4
0.999	0.998	1E-5
0.9997	0.9998	1E-8



返回

无约束优化问题的基本算法

1．最速下降法（共轭梯度法）算法步骤：

(1) 给定初始点 $X^0 \in E^n$，允许误差 $\varepsilon > 0$，令 k=0；

(2) 计算 $\nabla f\left(X^k\right)$；

(3) 检验是否满足收敛性的判别准则：

$$\left\|\nabla f\left(X^k\right)\right\| \le \varepsilon ,$$

若满足，则停止迭代，得点 $X^* \approx X^k$，否则进行(4)；

(4) 令 $S^k = -\nabla f\left(X^k\right)$，从 X^k 出发，沿 S^k 进行一维搜索，

即求 λ_k 使得：$\min\limits_{\lambda \ge 0} f\left(X^k + \lambda S^k\right) = f\left(X^k + \lambda_k S^k\right)$；

(5) 令 $X^{k+1} = X^k + \lambda_k S^k$，k=k+1 返回(2).

最速下降法是一种最基本的算法，它在最优化方法中占有重要地位. 最速下降法的优点是工作量小，存储变量较少, 初始点要求不高；缺点是收敛慢，最速下降法适用于寻优过程的前期迭代或作为间插步骤，当接近极值点时，宜选用别种收敛快的算法.

2. 牛顿法算法步骤：

(1) 选定初始点 $X^0 \in E^n$，给定允许误差 $\varepsilon > 0$，令 k=0；

(2) 求 $\nabla f(X^k)$, $(\nabla^2 f(X^k))^{-1}$, 检验：若 $\|\nabla f(X^k)\| < \varepsilon$，则停止迭代, $X^* \approx X^k$. 否则，转向(3)；

(3) 令 $S^k = -[\nabla^2 f(X^k)]^{-1} \nabla f(X^k)$（牛顿方向）；

(4) $X^{k+1} = X^k + S^k$, $k = k+1$, 转回(2).

如果f是对称正定矩阵A的二次函数，则用牛顿法经过**一次迭代**就可达到最优点,如不是二次函数，则牛顿法不能一步达到极值点，但由于这种函数在极值点附近和二次函数很近似,因此牛顿法的收敛速度还是很快的.

牛顿法的收敛速度虽然较快，但要求Hessian矩阵要可逆，要计算二阶导数和逆矩阵，就加大了计算机计算量和存储量.

3．拟牛顿法

为克服牛顿法的缺点，同时保持较快收敛速度的优点，利用第 k 步和第 k+1 步得到的 X^k ， X^{k+1} ， $\nabla f(X^k)$ ， $\nabla f(X^{k+1})$ ，构造一个正定矩阵 G^{k+1} 近似代替 $\nabla^2 f(X^k)$, 或用 H^{k+1} 近似代替 $(\nabla^2 f(X^k))^{-1}$ ，将牛顿方向改为：

$$G^{k+1} S^{k+1} = -\nabla f(X^{k+1}) \text{ ， } S^{k+1} = -H^{k+1} \nabla f(X^{k+1})$$

从而得到下降方向.

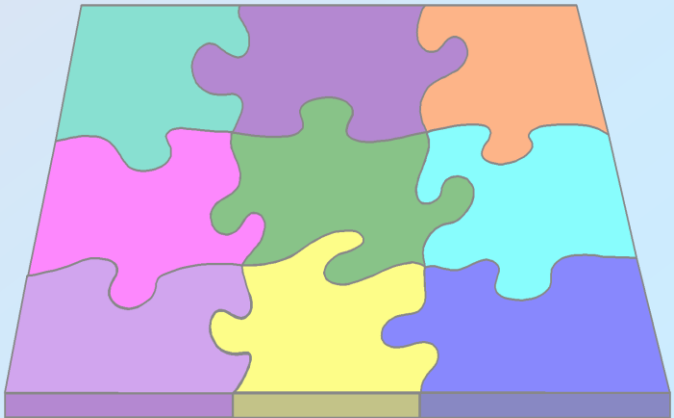
通常采用迭代法计算G^{k+1}，H^{k+1}，**迭代公式**为:

BFGS（Boryden-Fletcher-Goldfarb-Shanno）公式

$$G^{k+1}=G^{k}+\frac{\Delta f^{k}(\Delta f^{k})^{T}}{(\Delta f^{k})^{T}\Delta x^{k}}-\frac{G^{k}\Delta x^{k}(\Delta x^{k})^{T}G^{k}}{(\Delta x^{k})^{T}G^{k}\Delta x^{k}}$$

$$H^{k+1}=H^{k}+\left(1+\frac{(\Delta f^{k})^{T}H^{k}\Delta f^{k}}{(\Delta f^{k})^{T}\Delta x^{k}}\right)\frac{\Delta x^{k}(\Delta x^{k})^{T}}{(\Delta f^{k})^{T}\Delta x^{k}}$$

$$-\frac{\Delta x^{k}(\Delta f^{k})^{T}H^{k}-H^{k}\Delta f^{k}(\Delta x^{k})^{T}}{(\Delta f^{k})^{T}\Delta x^{k}}$$



DFP（Davidon-Fletcher-Powell）公式：

$$G^{k+1}=G^{k}+\left(1+\frac{(\Delta X^{k})^{T}G^{k}\Delta X^{k}}{(\Delta X^{k})^{T}\Delta f^{k}}\right)\frac{\Delta f^{k}(\Delta f^{k})^{T}}{(\Delta f^{k})^{T}\Delta X^{k}}$$

$$-\frac{\Delta f^{k}(\Delta X^{k})^{T}G^{k}-G^{k}\Delta X^{k}(\Delta f^{k})^{T}}{(\Delta X^{k})^{T}\Delta f^{k}}$$

$$H^{k+1}=H^{k}+\frac{\Delta X^{k}(\Delta X^{k})^{T}}{(\Delta f^{k})^{T}\Delta X^{k}}-\frac{H^{k}\Delta f^{k}(\Delta f^{k})^{T}H^{k}}{(\Delta f^{k})^{T}H^{k}\Delta f^{k}}$$

计算时可置$H^{1}=I$（单位阵），对于给出的X^{1}利用上面的公式进行递推.这种方法称为**拟牛顿法.**

返回

Matlab优化工具箱简介

1.MATLAB求解优化问题的主要函数

类　　型	模　　型	基本函数名
一元函数极小	Min F（x）s.t.x1<x<x2	x=fminbnd('F',x_1,x_2)
无约束极小	Min F(X)	X=fminunc('F',X_0) X=fminsearch('F',X_0)
线性规划	Min $c^T X$ s.t.AX<=b	X=linprog(c,A,b)
二次规划	Min$\frac{1}{2}$ x^THx+c^Tx s.t. Ax<=b	X=quadprog(H,c,A,b)
约束极小 （非线性规划）	Min F(X) s.t. G(X)<=0	X=fmincon('FG',X_0)
达到目标问题	Min r s.t. F(x)-wr<=goal	X=fgoalattain('F',x,goal,w)
极小极大问题	Min max {F_i(x)} X {Fi(x)} s.t. G(x)<=0	X=fminimax('FG',x_0)

2. 优化函数的输入变量

使用优化函数或优化工具箱中其它优化函数时，输入变量见下表：

变量	描　　述	调用函数
f	线性规划的目标函数f*X 或二次规划的目标函数X'*H*X+f*X 中线性项的系数向量	linprog,quadprog
fun	非线性优化的目标函数.fun必须为行命令对象或M文件、嵌入函数、或MEX文件的名称	fminbnd,fminsearch,fminunc, fmincon,lsqcurvefit,lsqnonlin, fgoalattain,fminimax
H	二次规划的目标函数X'*H*X+f*X 中二次项的系数矩阵	quadprog
A,b	A矩阵和b向量分别为线性不等式约束：$AX \le b$ 中的系数矩阵和右端向量	linprog,quadprog,fgoalattain, fmincon, fminimax
Aeq,beq	Aeq矩阵和beq向量分别为线性等式约束：$Aeq \cdot X = beq$ 中的系数矩阵和右端向量	linprog,quadprog,fgoalattain, fmincon, fminimax
vlb,vub	X的下限和上限向量：vlb≤X≤vub	linprog,quadprog,fgoalattain, fmincon,fminimax,lsqcurvefit, lsqnonlin
X_0	迭代初始点坐标	除fminbnd外所有优化函数
x_1, x_2	函数最小化的区间	fminbnd
options	优化选项参数结构，定义用于优化函数的参数	所有优化函数

3. 优化函数的输出变量下表:

变量	描　　述	调用函数
x	由优化函数求得的值.若exitflag>0,则x为解;否则,x不是最终解,它只是迭代制止时优化过程的值	所有优化函数
fval	解x处的目标函数值	linprog,quadprog,fgoalattain,fmincon,fminimax,lsqcurvefit,lsqnonlin, fminbnd
exitflag	描述退出条件: ● exitflag>0,表目标函数收敛于解x处 ● exitflag=0,表已达到函数评价或迭代的最大次数 ● exitflag<0,表目标函数不收敛	
output	包含优化结果信息的输出结构. ● Iterations:迭代次数 ● Algorithm:所采用的算法 ● FuncCount:函数评价次数	所有优化函数

4. 控制参数options的设置

Options中常用的几个参数的名称、含义、取值如下:

(1) Display: 显示水平.取值为’off’时,不显示输出;取值为’iter’时,显示每次迭代的信息;取值为’final’时,显示最终结果.默认值为’final’.

(2) MaxFunEvals: 允许进行函数评价的最大次数,取值为正整数.

(3) MaxIter: 允许进行迭代的最大次数,取值为正整数.

控制参数options可以通过函数optimset创建或修改。命令的格式如下：

(1) options=optimset('optimfun')

创建一个含有所有参数名,并与优化函数optimfun相关的默认值的选项结构options.

(2) options=optimset('param1',value1,'param2',value2,...)

创建一个名称为options的优化选项参数,其中指定的参数具有指定值,所有未指定的参数取默认值.

(3)options=optimset(oldops,'param1',value1,'param2',value2,...)

创建名称为oldops的参数的拷贝,用指定的参数值修改oldops中相应的参数.

例：opts=optimset('Display','iter','TolFun',1e-8)

该语句创建一个称为opts的优化选项结构,其中显示参数设为' iter', TolFun参数设为1e-8.

返回

用Matlab解无约束优化问题

1. 一元函数无约束优化问题：$min\ f(x)$　　$x_1 \le x \le x_2$

常用格式如下：

（1）x= fminbnd (*fun*, x_1, x_2)

（2）x= fminbnd (*fun*, x_1, x_2 , options)

（3）[x, fval]= fminbnd (...)

（4）[x, fval, exitflag]= fminbnd (...)

（5）[x, fval, exitflag, output]= fminbnd (...)

其中（3）、（4）、（5）的等式右边可选用（1）或（2）的等式右边。

函数fminbnd的算法基于黄金分割法和二次插值法，它要求目标函数必须是连续函数，并可能只给出局部最优解。

例 1　求　f = $2e^{-x}\sin x$ 在 0<x<8 中的最小值与最大值

主程序为wliti1.m:

```
f='2*exp(-x).*sin(x)';
fplot(f,[0,8]);                %作图语句
[xmin,ymin]=fminbnd (f, 0,8)
f1='-2*exp(-x).*sin(x)';
[xmax,ymax]=fminbnd (f1, 0,8)
```

To Matlab(wliti1)

运行结果:

```
xmin = 3.9270          ymin = -0.0279
xmax =  0.7854         ymax =  0.6448
```

例2 对边长为3米的正方形铁板，在四个角剪去相等的正方形以制成方形无盖水槽，问如何剪法使水槽的容积最大？

解 设剪去的正方形的边长为x，则水槽的容积为：$(3-2x^2)x$

建立无约束优化模型为：min y=$-(3-2x^2)x$， 0<x<1.5

先编写M文件fun0.m如下：

```
function f=fun0(x)
f=-(3-2*x).^2*x;
```

主程序为wliti2.m:

```
[x,fval]=fminbnd('fun0',0,1.5);
xmax=x
fmax=-fval
```

To Matlab(wliti2)

运算结果为： xmax = 0.5000, fmax =2.0000. 即剪掉的正方形的边长为0.5米时水槽的容积最大, 最大容积为2立方米.

2、多元函数无约束优化问题

标准型为：$min\ F(X)$

命令格式为:

(1) x= fminunc（fun, X_0 ）；或x=fminsearch（fun, X_0 ）

(2) x= fminunc（fun, X_0 , options）；
或x=fminsearch（fun, X_0 , options）

(3) [x, fval]= fminunc（...）；
或[x, fval]= fminsearch（...）

(4) [x, fval, exitflag]= fminunc（...）；
或[x, fval, exitflag]= fminsearch

(5) [x, fval, exitflag, output]= fminunc（...）；
或[x, fval, exitflag, output]= fminsearch（...）

说明:

- **fminsearch是用单纯形法寻优. fminunc的算法见以下几点说明:**

[1] fminunc为无约束优化提供了大型优化和中型优化算法。由options中的参数LargeScale控制:

LargeScale='on'(默认值),使用大型算法

LargeScale='off'(默认值),使用中型算法

[2] fminunc为中型优化算法的搜索方向提供了4种算法，由options中的参数HessUpdate控制:

HessUpdate='bfgs'（默认值），拟牛顿法的BFGS公式;

HessUpdate='dfp'，拟牛顿法的DFP公式;

HessUpdate='steepdesc'，最速下降法

[3] fminunc为中型优化算法的步长一维搜索提供了两种算法，由options中参数LineSearchType控制:

LineSearchType='quadcubic'(缺省值)，混合的二次和三次多项式插值;

LineSearchType='cubicpoly'，三次多项式插

- **使用fminunc和 fminsearch可能会得到局部最优解.**

例3 $\min f(x)=(4x_1^2+2x_2^2+4x_1x_2+2x_2+1)*\exp(x_1)$

1、编写M-文件 fun1.m:

```
function f = fun1 (x)
f = exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1)
```

2、输入M文件wliti3.m如下:

```
x0 = [-1, 1];
x=fminunc('fun1',x0);
y=fun1(x)
```

3、运行结果:

```
x=    0.5000      -1.0000
y =    1.3029e-10
```

To Matlab(wliti3)

例4　Rosenbrock 函数 f（x1，x2）=100（$x_2-x_1^2$）$^2+(1-x_1)^2$ 的最优解（极小）为 x*=（1，1），极小值为 f*=0. 试用不同算法（搜索方向和步长搜索）求数值最优解.

初值选为 x0=（-1.2 ， 2）.

1. **为获得直观认识，先画出 Rosenbrock　函数的三维图形,** 输入以下命令：

```
[x,y]=meshgrid(-2:0.1:2,-1:0.1:3);
z=100*(y-x.^2).^2+(1-x).^2;
mesh(x,y,z)
```

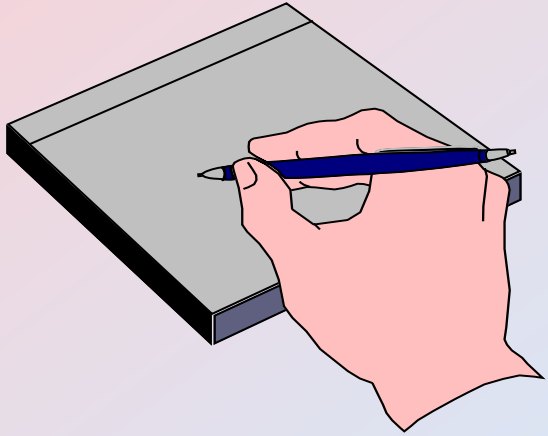
To Matlab (wliti31)

2. **画出 Rosenbrock　函数的等高线图,** 输入命令：

```
contour(x,y,z,20)
hold on
plot(-1.2,2,' o ');
text(-1.2,2,'start point')
plot(1,1,'o')
text(1,1,'solution')
```

To Matlab (wliti32)

3. 用fminsearch函数求解



输入命令:

```
f='100*(x(2)-x(1)^2)^2+(1-x(1))^2';
[x,fval,exitflag,output]=fminsearch(f, [-1.2 2])
```

To Matlab(wliti41)

运行结果:

```
      x =1.0000    1.0000
fval =1.9151e-010
exitflag = 1
output =
           iterations: 108
           funcCount: 202
          algorithm: 'Nelder-Mead simplex direct search'
```

4. 用fminunc 函数

(1)建立M-文件fun2.m

```
function f=fun2(x)
f=100*(x(2)-x(1)^2)^2+(1-x(1))^2
```

(2)主程序wliti44.m

To Matlab(wliti44)

Rosenbrock函数不同算法的计算结果

搜索方向	步长搜索	最优解	最优值	迭代次数
BFGS	混合二、三次插值	(0.9996，0.9992)	2.3109×10^{-7}	155
	三次插值	(1.0001，1.0002)	2.3943×10^{-8}	132
DFP	混合二、三次插值	(0.9995，0.9990)	2.6223×10^{-7}	151
	三次插值	(0.8994，0.7995)	0.0192	204
最速下降	混合二、三次插值	(-1.1634，1.3610)	4.6859	204
		(0.9446，0.8920)	0.0031	8002
		（0.9959，0.9916	1.8543×10^{-5}	9002
单纯形法		(1.0000,1.0000)	1.9151×10^{-10}	202

可以看出，最速下降法的结果最差.因为最速下降法特别不适合于从一狭长通道到达最优解的情况.

例5　产销量的最佳安排

某厂生产一种产品有甲、乙两个牌号，讨论在产销平衡的情况下如何确定各自的产量，使总利润最大. 所谓产销平衡指工厂的产量等于市场上的销量.

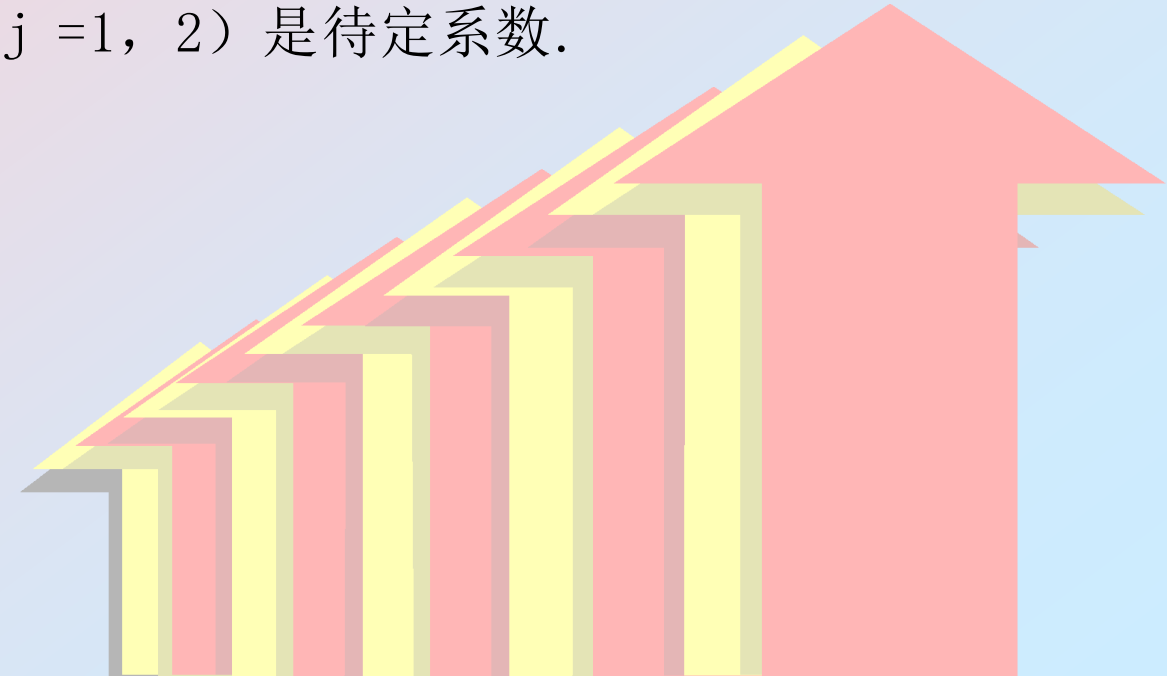
符号说明

$z(x_1, x_2)$表示总利润；

p_1，q_1，x_1分别表示甲的价格、成本、销量；

p_2，q_2，x_2分别表示乙的价格、成本、销量；

a_{ij}，b_i，λ_i, c_i（i，j =1，2）是待定系数.



基本假设

1．价格与销量成线性关系

利润既取决于销量和价格，也依赖于产量和成本。按照市场规律，甲的价格 p_1 会随其销量 x_1 的增长而降低，同时乙的销量 x_2 的增长也会使甲的价格有稍微的下降，可以简单地假设价格与销量成线性关系，

即：　$p_1 = b_1 - a_{11} x_1 - a_{12} x_2$，$b_1$，$a_{11}$，$a_{12} > 0$，且 $a_{11} > a_{12}$;

同理，　$p_2 = b_2 - a_{21} x_1 - a_{22} x_2$，$b_2$，$a_{21}$，$a_{22} > 0$，且 $a_{22} > a_{21}$.

2．成本与产量成负指数关系

甲的成本随其产量的增长而降低,且有一个渐进值,可以假设为负指数关系,即:

$$q_1 = r_1 e^{-\lambda_1 x_1} + c_1, \qquad r_1, \lambda_1, c_1 > 0$$

同理，

$$q_2 = r_2 e^{-\lambda_2 x_2} + c_2, \qquad r_2, \lambda_2, c_2 > 0$$

模型建立

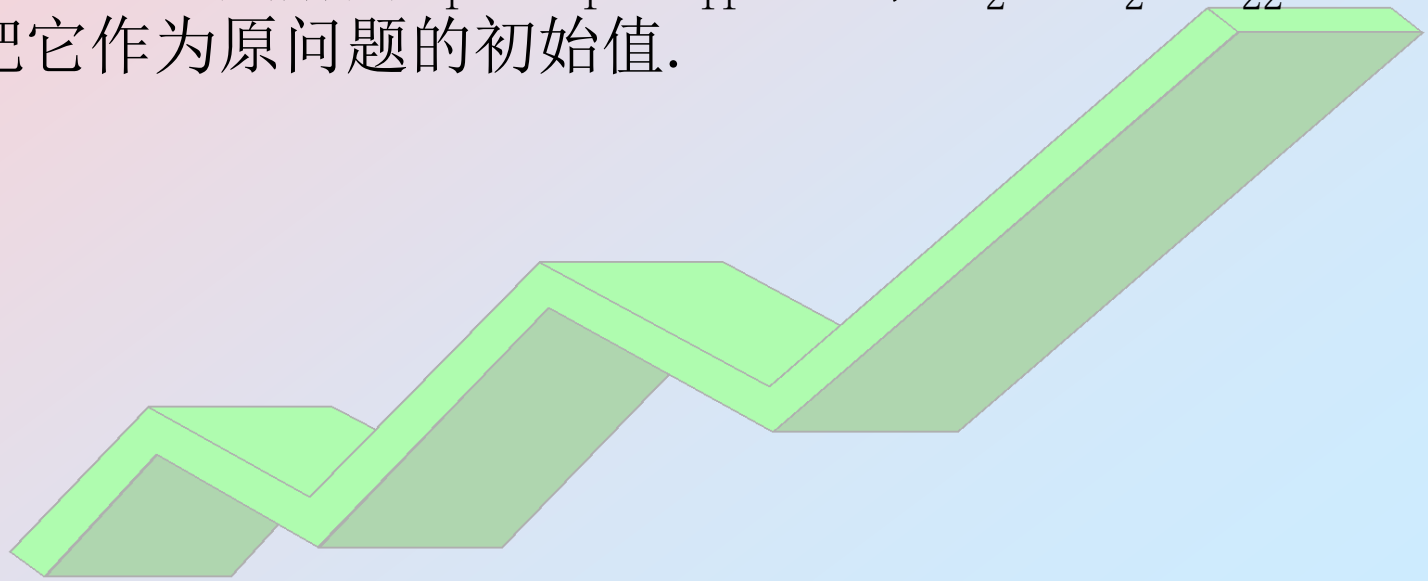
总利润为： $z(x_1, x_2)=(p_1-q_1)x_1+(p_2-q_2)x_2$

若根据大量的统计数据，求出系数$b_1=100$, $a_{11}=1$, $a_{12}=0.1$, $b_2=280$, $a_{21}=0.2$, $a_{22}=2$, $r_1=30$, $\lambda_1=0.015$, $c_1=20$, $r_2=100$, $\lambda_2=0.02$, $c_2=30$, 则问题转化为无约束优化问题：求甲，乙两个牌号的产量x_1，x_2，使总利润z最大.

为简化模型，先忽略成本，并令$a_{12}=0$, $a_{21}=0$, 问题转化为求：

$$z_1 = (b_1 - a_{11}x_1)x_1 + (b_2 - a_{22}x_2)x_2$$

的极值. 显然其解为$x_1 = b_1/2a_{11} = 50$, $x_2 = b_2/2a_{22} = 70$, 我们把它作为原问题的初始值.



模型求解

1. 建立M-文件fun.m:

```
function f = fun(x)
y1=((100-x(1)- 0.1*x(2))-(30*exp(-0.015*x(1))+20))*x(1);
y2=((280-0.2*x(1)- 2*x(2))-(100*exp(-0.02*x(2))+30))*x(2);
f=-y1-y2;
```

2. 输入命令:

```
x0=[50,70];
x=fminunc('fun',x0),
z=fun(x)
```

To Matlab(wliti5)

3. 计算结果:

x=23.9025, 62.4977, z=6.4135e+003

即甲的产量为23.9025,乙的产量为62.4977,最大利润为6413.5.

返回

实验作业

1. 求下列函数的极小点：

1) $f(X)=x_1^2+4x_2^2+9x_3^2-2x_1+18x_2$；

2) $f(X)=x_1^2+\frac{3}{2}x_2^2-2x_1x_2+x_1-2x_2$；

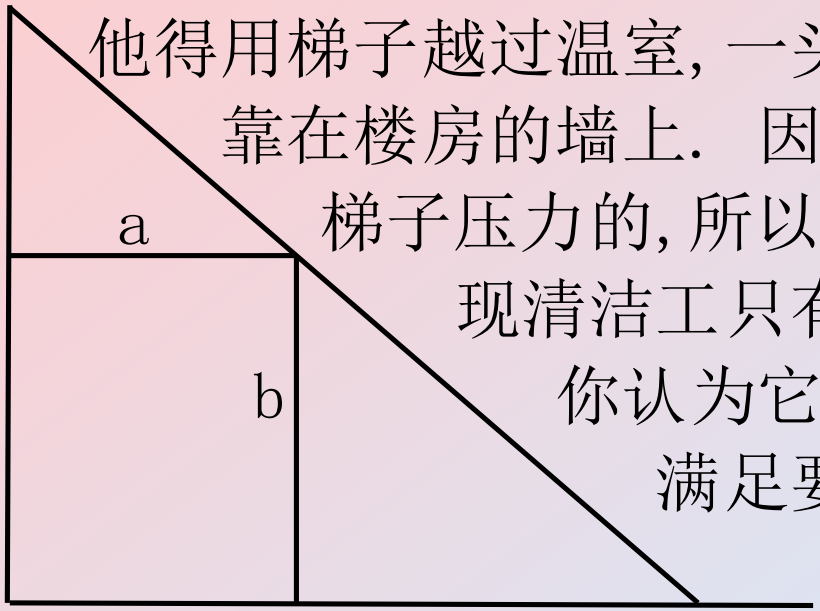
3) $f(X)=(x_1-1)^4+2_2^2$.

第 1），2）题的初始点可任意选取，

第 3）题的初始点取为$X^0=(0,1)^T$.

2. 梯子长度问题

一楼房的后面是一个很大的花园. 在花园中紧靠着楼房有一个温室,温室伸入花园 2m,高 3m,温室正上方是楼房的窗台. 清洁工打扫窗台周围,他得用梯子越过温室,一头放在花园中,一头靠在楼房的墙上. 因为温室是不能承受梯子压力的,所以梯子太短是不行的. 现清洁工只有一架 7m 长的梯子,你认为它能达到要求吗? 能满足要求的梯子的最小长度为多少?



3. 陈酒出售的最佳时机问题

某酒厂有批新酿的好酒,如果现在就出售,可得总收入 R_0=50 万元(人民币),如果窖藏起来待来日(第 n 年)按陈酒价格出售,第 n 年末可得总收入 $R = R_0 e^{\sqrt{n}/6}$ (万元),而银行利率为 r=0.05,试分析这批好酒窖藏多少年后出售可使总收入的现值最大. (假设现有资金 X 万元,将其存入银行,到第 n 年时增值为 R(n)万元,则称 X 为 R(n)的现值.)并填下表:

第一种方案：将酒现在出售，所获 50 万元本金存入银行；

第二种方案：将酒窖藏起来，待第 n 年出售。

（1）计算 15 年内采用两种方案，50 万元增值的数目并填入表 1，2 中；

（2）计算 15 年内陈酒出售后总收入 R（n）的现值填入表 3 中。

表 1　第一种方案

第 1 年	第 2 年	第 3 年	第 4 年	第 5 年
第 6 年	第 7 年	第 8 年	第 9 年	第 10 年
第 11 年	第 12 年	第 13 年	第 14 年	第 15 年

表 2　第二种方案

第 1 年	第 2 年	第 3 年	第 4 年	第 5 年
第 6 年	第 7 年	第 8 年	第 9 年	第 10 年
第 11 年	第 12 年	第 13 年	第 14 年	第 15 年

表 3　陈酒出售后的现值

第 1 年	第 2 年	第 3 年	第 4 年	第 5 年
第 6 年	第 7 年	第 8 年	第 9 年	第 10 年
第 11 年	第 12 年	第 13 年	第 14 年	第 15 年