无约束优化案例及求解

清华大学数学科学系　　张立平
Email: lipingzhang@tsinghua.edu.cn
Office: 理科楼A302
Tel:　　62798531

MATLAB优化工具箱

模型：$\underset{x}{Min}\ f(x), x \in R^n$

fun.m ~ f(x)的m文件名

x0 ~初始点；　x ~最优解

P1,P2, … ~ 传给fun的参数

中间输入项缺省用[ ]占位

基本用法：

x=fminunc(‘fun’,x0)

x=fminunc(‘fun’,x0,options,P1,P2,...)

例1: $\min \ \dfrac{x^2}{a} + \dfrac{y^2}{b}$

```
function y=uncopt1fun(x,a,b)
y=x(1)^2/a+x(2)^2/b;
```

其中$a = b = 2$

```
x0=[1,1];a=2;b=2;

x=fminunc('uncopt1fun',x0,[],a,b)
```

examUNCOPT1.m

X=(0,0)

控制精度，观察中间结果，控制迭代次数等

Options 控制参数设定: optimoptions; optimset

```
Optimset                  //显示控制参数
opt=optimset              //控制参数设为[](即缺省值)
opt=optimset(optfun)//optfun控制参数设缺省值
Opt=optimset('par1',val1,'par2',val2,...)
Opt=optimset(oldopts,'par1',val1,...)
opt=optimset(oldopts,newopts)
```

options=optimoptions(@fminunc,'Display',
'iter', 'Algorithm', 'quasi-newton');

主要控制参数

```
Diagnostics  ‘on’ | {‘off’} //是否显示诊断信息
Display      'off' | 'iter' |
            ‘final’ | ‘notify‵ //显示信息的级别
GradObj      ‘on’ | {‘off’}//是否采用分析梯度
Jacobian     ‘on’ | {‘off’}
               //采用分析Jacob阵（用于约束优化中）
MaxFunEvals   最大函数调用次数
MaxIter         最大迭代次数
TolCon        约束的控制精度（用于约束优化中）
TolFun        函数值的控制精度
TolX          解的控制精度
```

更多输出：最优值等

最一般的输出形式

`[x,f,exitflag,out,grad,hess]=fminunc(...)`

F 目标函数值

exitflag >0收敛,0达到函数或迭代次数，<0不收敛

Output

- **iterations** 实际迭代次数
- **funcCount** 实际函数调用次数
- **algorithm** 实际采用的算法
- **cgiterations** 实际PCG迭代次数（大规模算法用）
- **stepsize** 最后迭代步长（中等规模算法用）
- **firstorderopt** 一阶最优条件（梯度的范数）

grad 目标函数的梯度

hess 目标函数的Hessian矩阵

例 2 $\min \dfrac{x_1^2}{a}+\dfrac{x_2^2}{b}, \quad a=10, b=1$

examUNCOPT2.m

```
x0=[1,1];a=10;b=1; format short e
fopt1=optimset('Display','iter');
[x,f,exitf]=fminunc(‘uncopt1fun',x0,fopt1,a,b);
x,f,exitf
pause

fopt2=optimset('TolFun',1e-8,'TolX',1e-8);
[x,f,exitf,output,grad]=fminunc(‘uncopt1fun',x0,fopt2,
a,b);
x,f,exitf,output,grad
pause

fopt3=optimset('TolFun',1e-1,'TolX',1e-1);
[x,f,exitf,output,grad]=fminunc(‘uncopt1fun',x0,fopt3,
a,b);
x,f,exitf,output,grad
```

算法选择：options中参数控制

系统缺省采用拟牛顿算法；还可选择**trust-region**

搜索方向的算法选择

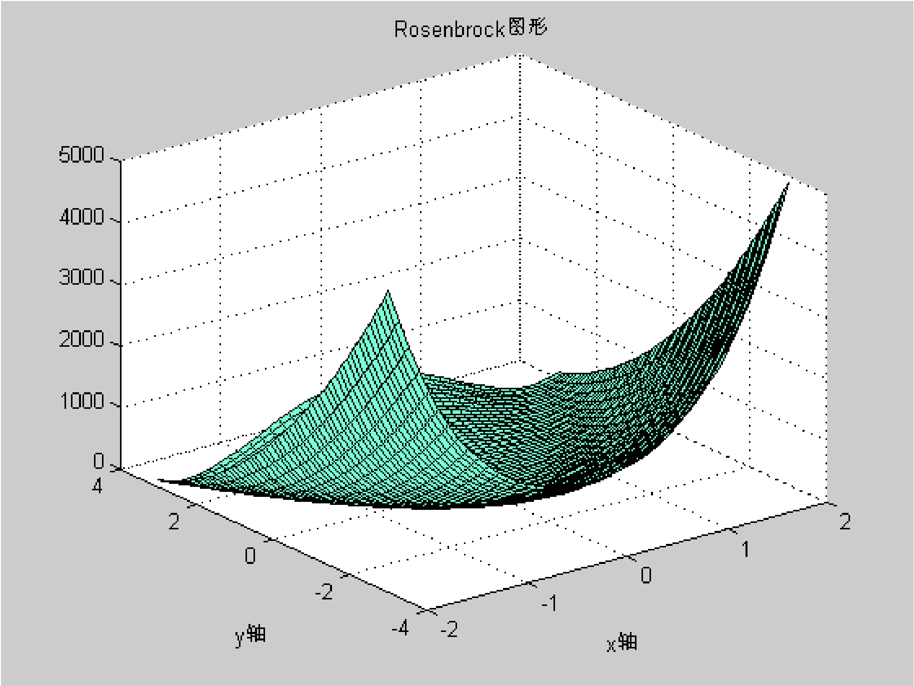
HessUpdate = ‘dfp’ （DFP算法）

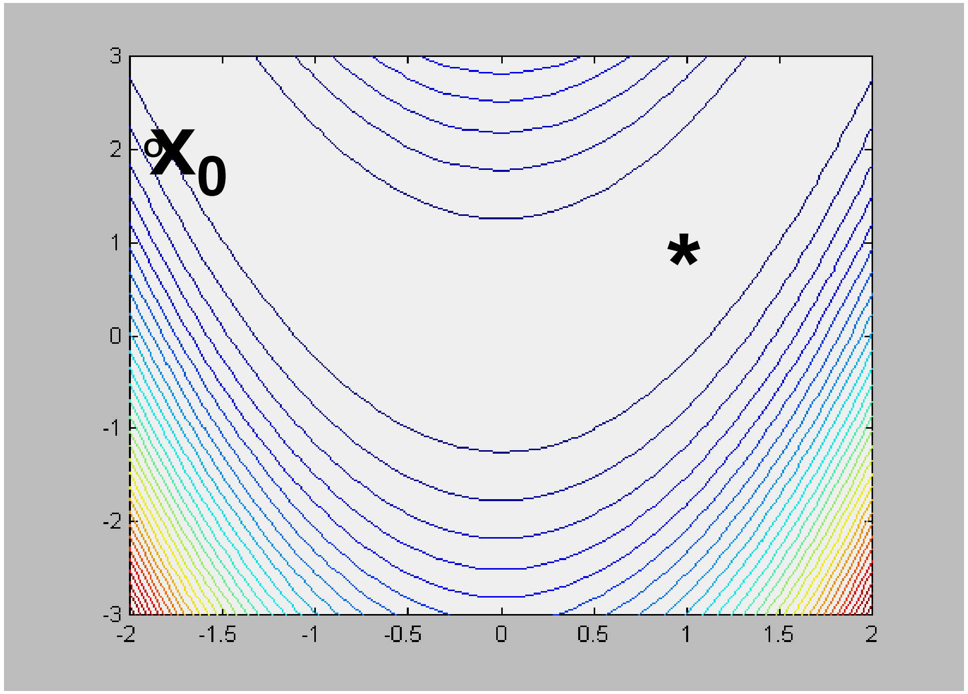
HessUpdate = ‘steepdesc’（最速下降算法）

```
options = optimoptions('fminunc', 'Algorithm', 'trust-region',
                        'SpecifyObjectiveGradient',true);
```

例3. $\min \quad f(x,y)=100(y-x^2)^2+(1-x)^2$

$f(x,y)=100(y-x^2)^2+(1-x)^2$ 的图形





uncopt2plot.m

$$\min\ f(x,y)=100(y-x^2)^2+(1-x)^2$$

精确解：$x=y=1$，$f(x,y)=0$

examUNCOPT2.m

计算结果

算法	最优解	最优值	迭代次数	函数调用次数	终止条件
BFGS	1.0000e+00 9.9999e-01	2.3633e-11	21	81	5.2179e-07
DFP	1.0000e+00 9.9999e-01	1.9959e-11	26	111	2.0153e-06
SDM	4.9862e-01 2.3635e-01	2.6644e-01	失败		2.4547e+00

无约束优化　模型: $\underset{x}{Min}\ f(x),\ x\in R^n$

采用分析梯度:

```
GradObj='on'
x=fminunc(@fun,x0,opt,…)
```

fun.m中还要有$\nabla f(x)$

一般形式 `function [f,g]=fun(x)`

$$\min\ f(x,y)=100(y-x^2)^2+(1-x)^2$$

$$\text{算出}\ \nabla f=\begin{bmatrix}-400x(y-x^2)-2(1-x)\\ 200(y-x^2)\end{bmatrix}$$

examUNCOPT2grad.m

计算结果

算法	最优解	最优值	迭代次数	函数调用次数	终止条件
BFGS	1.0000e+00 1.0000e+00	1.5385e-13	21	27	1.8832e-06
DFP	1.0000e+00 1.0000e+00	4.9792e-13	27	37	2.1336e-05
Trust-Region	1.0000e+00 1.0000e+00	1.9932e-11	13	14	1.6475e-04

与不用分析梯度的结果比较

算法	最优解	最优值	迭代次数	函数调用次数	终止条件
BFGS	1.0000e+00 9.9999e-01	2.3633e-11	21	81	5.2179e-07
DFP	1.0000e+00 9.9999e-01	1.9959e-11	26	111	2.0153e-06

几个值得注意的问题

梯度函数：利用分析梯度可能改进算法的性能

精度控制：对迭代次数有重大影响，应适当选择。

改变初始值 由一个初值出发通常得到局部最优解，如果函数存在多个局部最优，只有改变初值，对局部最优进行比较，才有可能得到全局最优解。

算法选择：**BFGS**，一般较好。

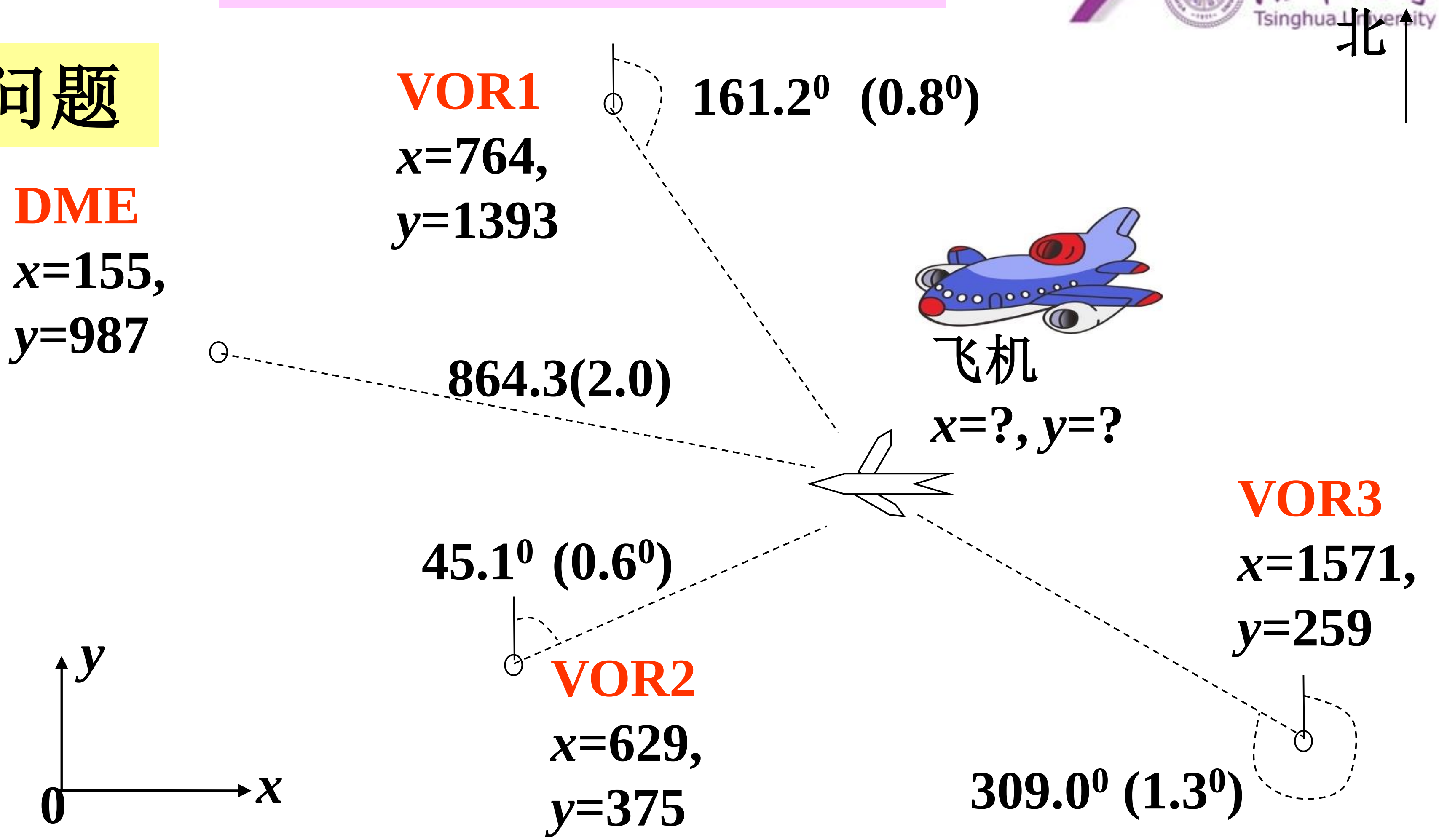
其他算法选择： （详细用法请查阅**help**文档）

高度非线性、不连续时可用程序 **fminsearch(@fun,x0)**

单变量时可用程序 **fminbnd(@fun,v1,v2)**

例 飞机的精确定位

问题



图中坐标和测量距离的单位是“公里”

模型准备

已知数据：设备位置坐标分别为$(x_i, y_i), i=1,\ldots 4;$

记测量角度为θ_i，角度误差限为$\sigma_i, i=1,\ldots,3;$

记测量距离为d_4，距离误差限为$\sigma_4.$

要求计算：飞机位置坐标（x, y）

	x_i	y_i	ϑ_i 或 d_4	σ_i
VOR1	746	1393	161.20 (2.813 rad)	0.80 (0.0140 rad)
VOR2	629	375	45.10 (0.787 rad)	0.60 (0.0105 rad)
VOR3	1571	259	309.00 (5.393 rad)	1.30 (0.0227 rad)
DME	155	987	864.3 (km)	2.0 (km)

模型

不考虑误差因素

量纲不符！

清華大学 Tsinghua University

$$\tan\theta_i = (x-x_i)/(y-y_i)$$

$$\sqrt{(x-x_4)^2+(y-y_4)^2} = d_4$$

？

超定方程组，

非线性最小二乘！

$$Min \quad J(x,y) = \sum_{i=1}^{3}[(x-x_i)/(y-y_i)-\tan\theta_i]^2$$

$$+[d_4-\sqrt{(x-x_4)^2+(y-y_4)^2}]^2$$

考虑误差因素

$$Min \quad E(x,y) = \sum_{i=1}^{3}\left(\frac{\alpha_i-\theta_i}{\sigma_i}\right)^2 + \left(\frac{d_4-\sqrt{(x-x_4)^2+(y-y_4)^2}}{\sigma_4}\right)^2$$

$$\tan\alpha_i = (x-x_i)/(y-y_i)$$

非线性规划模型(NLP)

例(飞机的精确定位)的求解

LINGO软件实现

不考虑误差因素

$$Min\quad J(x,y)=\sum_{i=1}^{3}[(x-x_i)/(y-y_i)-\tan\theta_i]^2$$
$$+[d_4-\sqrt{(x-x_4)^2+(y-y_4)^2}]^2$$

演示
examUNCOPT3_1

全局优化:

(980.7, 731.6)

目标值0.0007

```
MODEL:
SETS:
VOR/1..3/: x, y, cita, sigma;
ENDSETS
DATA:
x, y, cita, sigma =
    746      1393      2.81347       0.0140
    629       375      0.78714       0.0105
    1571      259      5.39307       0.0227;
x4 y4 d4 sigma4 = 155    987    864.3    2.0;
ENDDATA
! XX,YY表示飞机坐标;
min = @sum(VOR: @sqr((xx-x)/(yy-y) - @tan(cita)) )
      + @sqr(d4 - @sqrt(@sqr(xx-x4)+@sqr(yy-y4)) );
END
```

$$Min \quad J(x,y)=\sum_{i=1}^{3}[(x-x_i)/(y-y_i)-\tan\theta_i]^2$$
$$+[d_4-\sqrt{(x-x_4)^2+(y-y_4)^2}]^2$$

例(飞机的精确定位)的求解

考虑误差因素

$$Min \quad E(x,y) = \sum_{i=1}^{3}\left(\frac{\alpha_i - \theta_i}{\sigma_i}\right)^2 + \left(\frac{d_4 - \sqrt{(x-x_4)^2 + (y-y_4)^2}}{\sigma_4}\right)^2$$

$$\tan\alpha_i = (x - x_i)/(y - y_i)$$

约束非线性规划

LINGO软件实现

演示examUNCOPT3_2

注意初值: (980.7, 731.6)

全局优化:

(978.3, 724.0)

目标值0.6670

为什么目标值显著增大?

```
MODEL:
SETS:
VOR/1..3/: x, y, cita, sigma, alpha;
ENDSETS
DATA:
x, y, cita, sigma =
746     1393    2.81347     0.0140
629     375     0.78714     0.0105
1571    259     5.39307     0.0227;
x4 y4 d4 sigma4 = 155   987   864.3   2.0;
ENDDATA
INIT:
xx, yy = 980.69, 731.57;
ENDINIT
! XX,YY表示飞机坐标;
min = @sum(VOR: @sqr((alpha - cita)/sigma) )
      + @sqr((d4 - @sqrt(@sqr(xx-x4)+@sqr(yy-y4)))/sigma4 );
@for(VOR: @tan(alpha)=(xx-x)/(yy-y););
END
```

$$Min\ \ E(x,y)=\sum_{i=1}^{3}\left(\frac{\alpha_i-\theta_i}{\sigma_i}\right)^2+\left(\frac{d_4-\sqrt{(x-x_4)^2+(y-y_4)^2}}{\sigma_4}\right)^2$$

$$\text{s.t.}\quad \tan\alpha_i=(x-x_i)/(y-y_i)$$

使用LINGO的一些注意事项

1. LINGO解非线性规划与线性规划类似，采用什么算法全由LINGO自行决定。
2. LINGO缺省假设所有变量非负，如果有的变量可以取负，一定用@free函数去掉非负限制，否则可能得不到希望的结果。
3. 对于非线性规划，乘子非零仅表示对应的约束有效。
4. LINGO缺省设置只寻求局部最优解。如果做全局优化，运行菜单中的LINGO|Options命令，选择“Global Solver”选项卡，选中“Use Global Solver”，保存应用。