信陽師範學院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

第4章 函数与数值积分

讲授人：牛言涛

日期：2020年2月24日

目录
CONTENTS

1. 一元函数极值函数fminbnd

- **一元函数的极小值：fminbnd 求得函数在给定区间内的局部极小值。**

只有求最小值的方法

如果相求最大值，直接修改原方程，根据最小值求最大值

[x, fval,exitflag,output] = fminbnd(fun,x1,x2,options)

- x为极值点，fval所求函数极值；
- exitflag > 0收敛到解， exitflag = 0已达最大迭代次数， exitflag < 0不收敛；
- fun 为函数句柄；
- x1 和 x2 分别用于指定区间的左右边界；
- options 用于指定程序的其他参数，其元素取值如下表所示。

1. 一元函数极值函数fminbnd

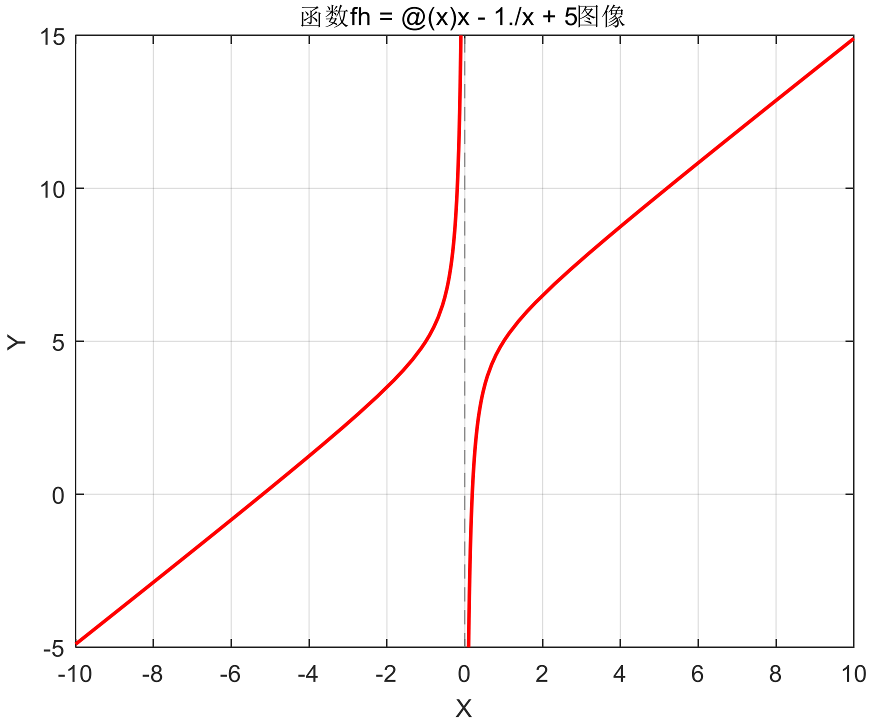
名称	options = optimset('Display','iter','PlotFcns',@optimplotfval);
Display	控制结果的输出，参数可以为“off”，不输出任何结果；“iter”，输出每个插值点的值；“final”，输出最后结果；“notify”为默认值，仅当函数不收敛时输出结果。
FunValCheck	检测目标函数值是否有效。选择 on 则当函数返回数据为复数或空数据时发出警告； off 则不发出警告。
MaxFunEvals	允许进行函数评价的最大次数。
MaxIter	最大迭代次数。
OutputFcn	指定每次迭代时调用的用户自定义的函数。
PlotFcns	@optimplotx plots the current point @optimplotfunccount plots the function count @optimplotfval plots the function value
TolX	返回的 x 的误差。

2. 一元函数极值案例

数学与统计学院

例：求函数 $f(x)=x-\dfrac{1}{x}+5$ 在区间(-10,1)和(1,10)上的最小值点。

```
>> fh = @(x)x - 1./x + 5;  %定义匿名函数
>> fplot(fh,[-10,10],'r-','LineWidth',1.5)
>> grid on
>> [x,fval,exitflag,output] = fminbnd(fh, -10,1)  %求(-10,1)区间最小值
x =
   -9.9999
fval =
   -4.8999
exitflag =
     1
output =
  包含以下字段的 struct:

    iterations: 24
     funcCount: 25
     algorithm: 'golden section search, parabolic interpolation'
       message: '优化已终止:…'
>> [x2,fval2,exitflag2,output2] = fminbnd(fh,1,10)  %求(1,10)区间最小值
x2 =
    1.0001
fval2 =
    5.0001
exitflag2 =
     1
```



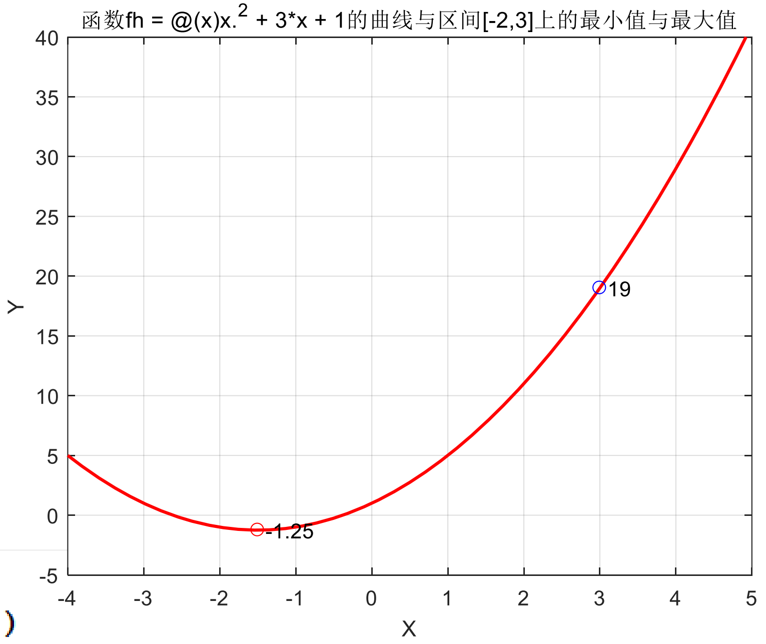
2. 一元函数极值案例

例：求函数$f(x) = x^2 + 3x + 1$在[-2, 3]内的最小值和最大值。

```
>> fh_min = @(x)x.^2 + 3*x + 1;  %定义匿名函数，最小值
>> [xmin, fmin,exitflag] = fminbnd(fh_min, -2,3,optimset('TolX',1e-12,'Display','iter'))

 Func-count     x          f(x)         Procedure
    1       -0.0901699    0.737621       initial
    2          1.09017     5.45898       golden
    3         -0.81966   -0.787138       golden
    4             -1.5       -1.25       parabolic
    5             -1.5       -1.25       parabolic
    6             -1.5       -1.25       parabolic

优化已终止:
 当前的 x 满足使用 1.000000e-12 的 OPTIONS.TolX 的终止条件

xmin =
   -1.5000
fmin =
   -1.2500
exitflag =
     1
```

```
>> fh_max = @(x)-(x.^2 + 3*x + 1);  %求最大值转化为求最小值的方法
>> [xmax, fmax,exitflag] = fminbnd(fh_max, -2,3,optimset('TolX',1e-12))
xmax =
    3.0000
fmax =
  -19.0000
exitflag =
     1
```

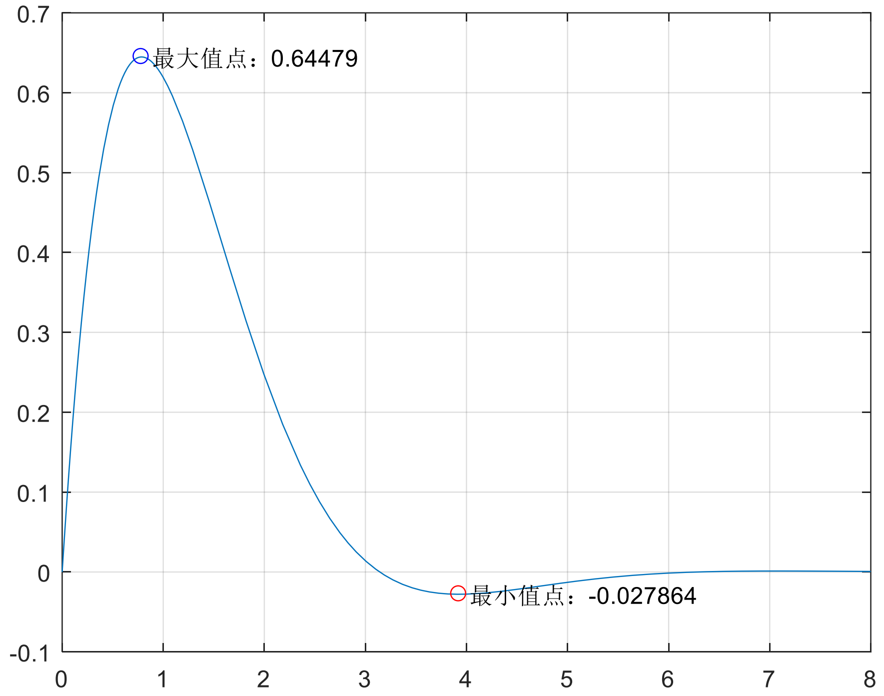
注意：fmax = -fmax，转化



2. 一元函数极值案例

- 例：求函数 $f = 2e^{-x}\sin x$ 在(0, 8)中的最大值和最小值。

```
>> fh_min = @(x)2*exp(-x).*sin(x); %最小值匿名函数
>> fh_max = @(x)-2*exp(-x).*sin(x); %最大值匿名函数，前加负号
>> [xmin, fmin,exitflag] = fminbnd(fh_min,0,8);
>> [xmax, fmax,exitflag] = fminbnd(fh_max,0,8);
>> fplot(fh_min, [0,8])
>> axis([0 8 -0.1 0.7]) %改变坐标轴的范围
>> text(xmin+0.1,fmin,['最小值点： ',num2str(fmin)])
>> text(xmax+0.1,-fmax,['最大值点： ',num2str(-fmax)])
>> hold on
>> plot(xmin,fmin,'ro', xmax,-fmax, 'bo')
>> grid on
```



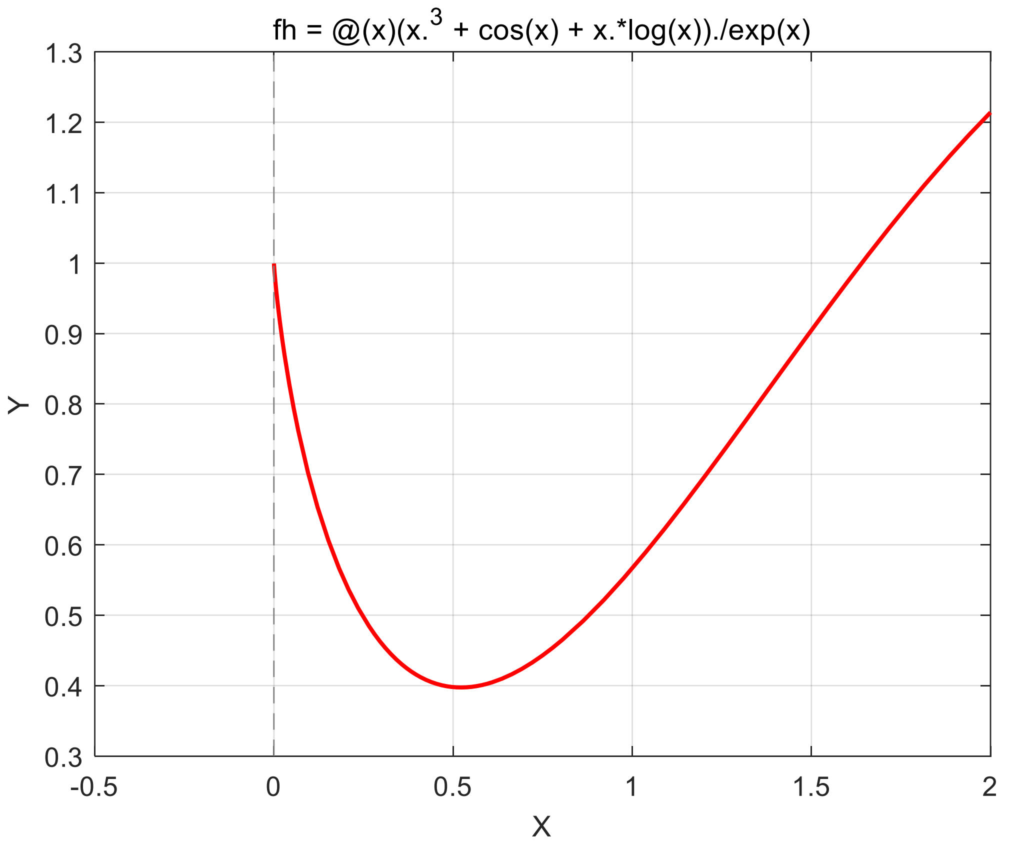

```
xmin =   3.9270     ymin = -0.0279
xmax = 0.7854       ymax = 0.6448
```

信阳师范学院 数学与统计学院 SCHOOL OF MATHEMATICS AND STATISTICS

2. 一元函数极值案例

- 例：求函数 $f(x)=\dfrac{x^3+\cos x+x\log x}{e^x}$ 在(0,1)内的最小值和最大值。

```
>> fh = @(x)(x.^3 + cos(x) + x.*log(x))./exp(x);
>> [xmin,ymin,exitflag] = fminbnd(fh,0,1,optimset('TolX',1e-12,'Display','iter'))

 Func-count     x          f(x)         Procedure
    1        0.381966   0.420464        initial
    2        0.618034   0.406237        golden
    3        0.763932    0.44824        golden
    4         0.53306   0.397484        parabolic
    5        0.527885   0.397396        parabolic
    6        0.521623   0.397364        parabolic
    7        0.522255   0.397363        parabolic
    8        0.522276   0.397363        parabolic
    9        0.522275   0.397363        parabolic
   10        0.522275   0.397363        parabolic
   11        0.522275   0.397363        parabolic

优化已终止:
 当前的 x 满足使用 1.000000e-12 的 OPTIONS.TolX 的终止条件

xmin =
    0.5223
ymin =
    0.3974
exitflag =
     1
```



3. 多元函数的极小值

用fminsearch、fminunc求多元函数的最小值问题

- [x,fval,exitflag,output] = fminsearch(@fh, x0, options)：此函数使用单纯型法搜索最值。
  - 其中fh为待求最值的向量函数，$x0$为搜索过程开始时自变量的初始值。
- [x,fval,exitflag,output,grad,hessian] = fminunc(@fh,x0 , options)：此函数使用牛顿法搜索最值，在效率上有所提高。

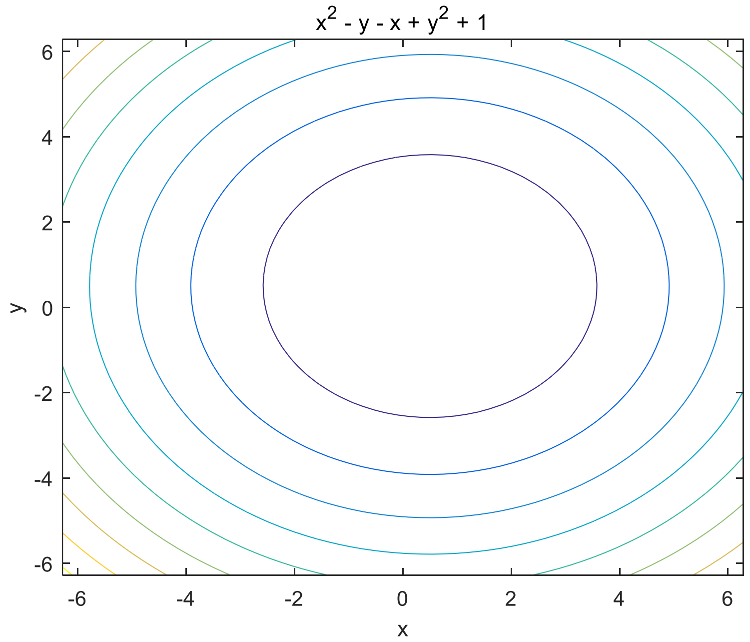
注意事项

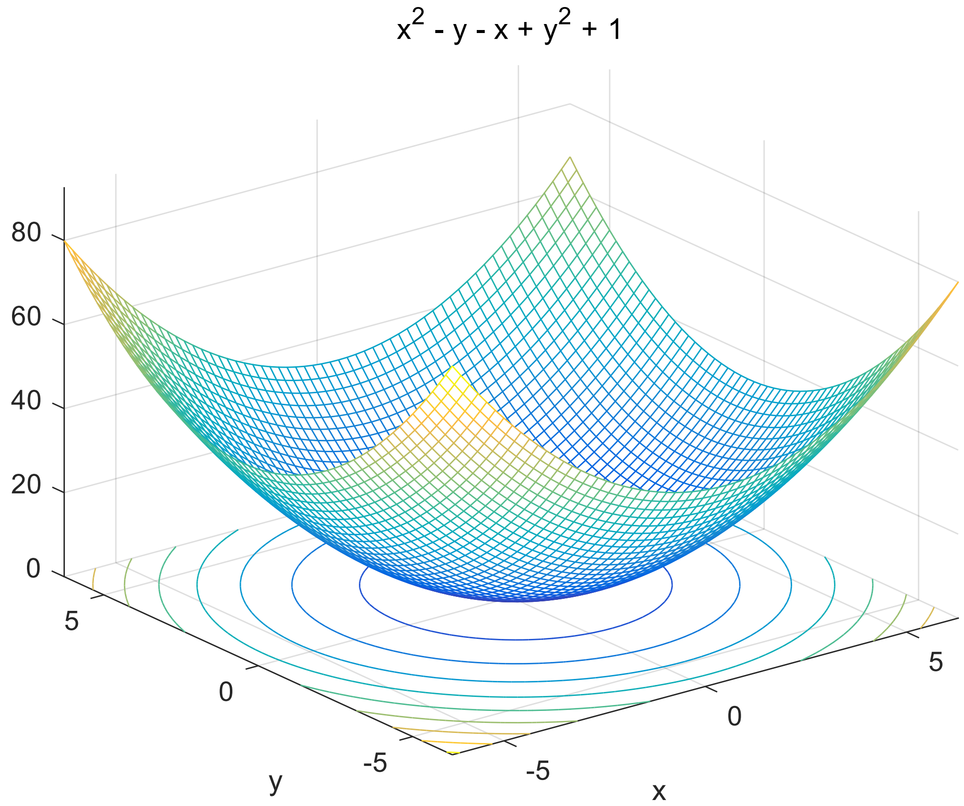
- 在使用这两个函数时，必须首先用M文件（非唯一形式）的形式存储待求最值的函数，并且需以向量函数的形式表达；注意参数的写法问题。
- 最大值问题需转化为最小值问题。

4. 多元函数的极小值案例

例：求函数 $f(x,y)=-(x+y)+(x^2+y^2+1)$ 在 $x=1, y=2$ 附近的最小值点。

```
>> syms x y;   %利用符号函数绘图方式
>> fxy = -(x+y) + (x^2 + y^2 + 1);
>> ezmeshc(fxy)  %绘制图形
>> ezcontour(fxy) %绘制等高线图
```





4. 多元函数的极小值案例

```
>> fh = @(x)-(x(1)+x(2)) + (x(1)^2+x(2)^2+1);
>> x0 = [1,2]; %通过三维图和等高线图得知在[1,2]附近有极值
>> [x,fval,exitflag] = fminsearch(fh,x,optimset('TolX',1e-12, 'Display','iter'))

 Iteration   Func-count     min f(x)         Procedure
     0            1            0.5
     1            3            0.5          initial simplex
     2            5            0.5          contract inside
     3            7            0.5          contract outside
     4            9            0.5          contract inside
     5           11            0.5          contract outside
     6           13            0.5          contract outside
     7           15            0.5          contract outside

    66          149           0.5          shrink
    67          153           0.5          shrink
    68          157           0.5          shrink

优化已终止:
 当前的 x 满足使用 1.000000e-12 的 OPTIONS.TolX 的终止条件,
F(X) 满足使用 1.000000e-04 的 OPTIONS.TolFun 的收敛条件

x =
    0.5000    0.5000
fval =
    0.5000
exitflag =
     1
```

1、建立M文件，保存函数fmin.m；

```
function fh = fmin(x)
   fh = -(x(1)+x(2)) + (x(1)^2+x(2)^2+1);
end
```

2、调用fminsearch函数求最值。在命令窗口中，输入：

```
x0 = [1,2];
[x,fval] = fminsearch(@f1,x0)
```

3、输出结果为：

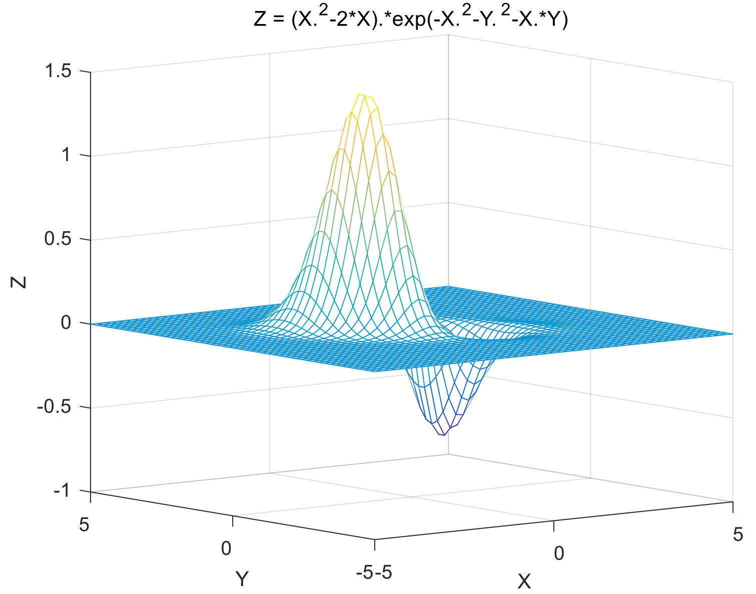
X = 0.5000   0.5000

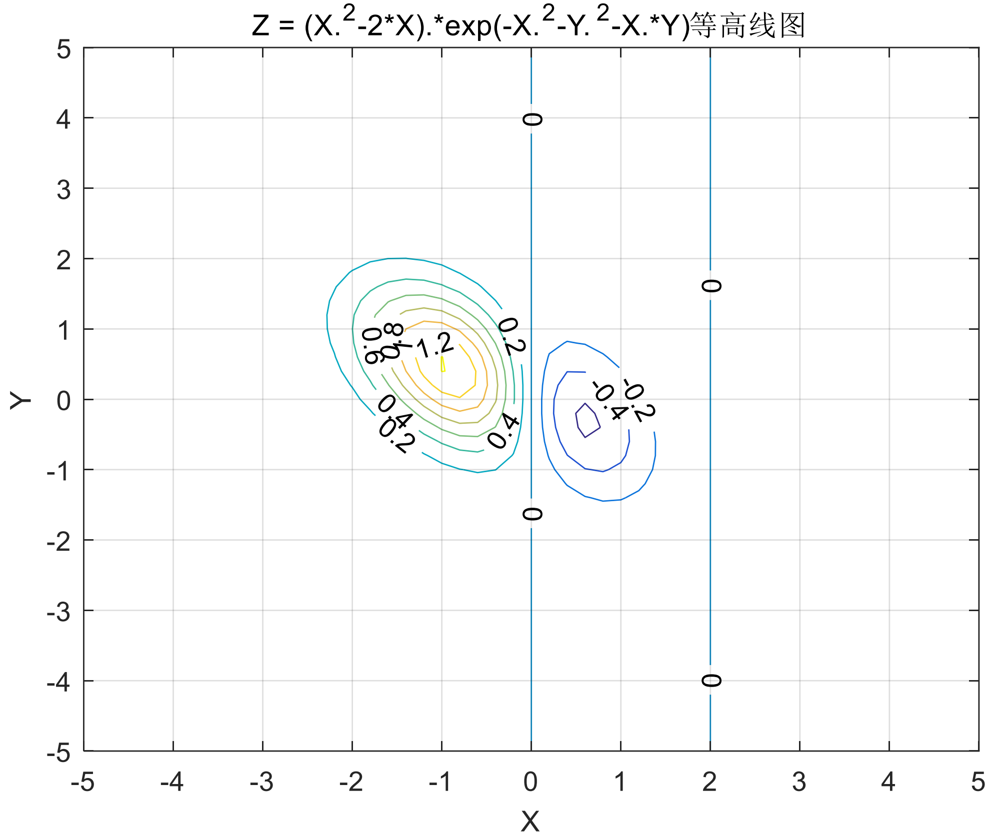
fval = 0.5000

4. 多元函数的极小值案例

例：求二元函数 $z=(x^2-2x)e^{-x^2-y^2-xy}$ 的（局部）最小值点（曲面的最低点）。

```
>> x = -5:0.2:5;
>> y = -5:0.2:5;
>> [X,Y] = meshgrid(x,y);
>> Z = (X.^2-2*X).*exp(-X.^2-Y.^2-X.*Y);
>> mesh(X,Y,Z)   %绘制网格面
>> figure(2)
>> [c,h] = contour(X,Y,Z);   %绘制等高线图
>> set(h,'ShowText','on')
>> grid on
```

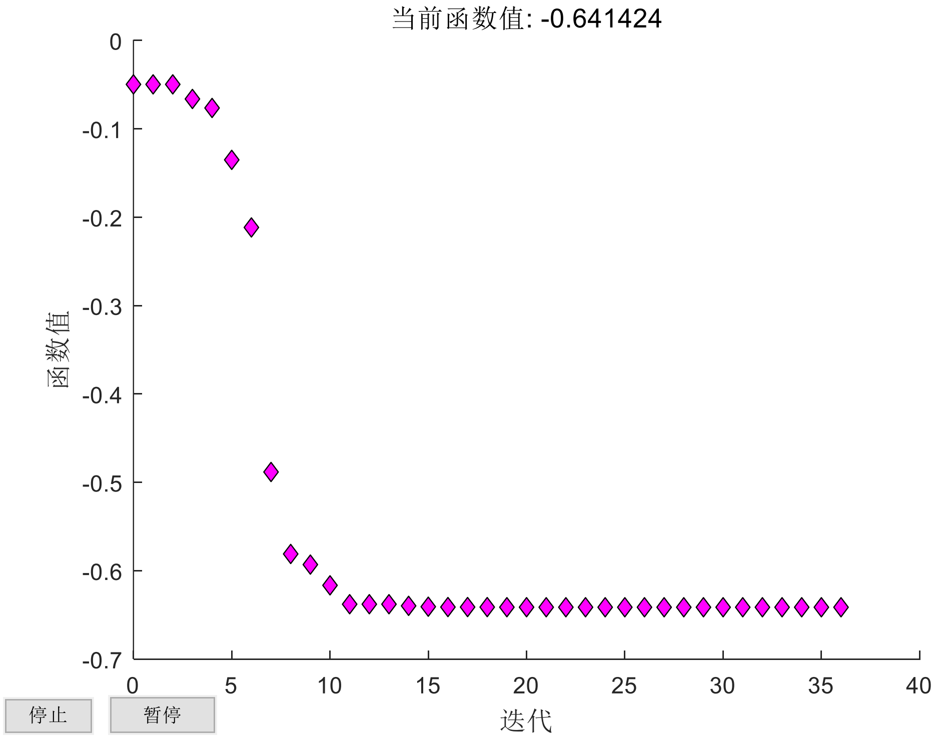




4. 多元函数的极小值案例

例：求二元函数 $z=(x^2-2x)e^{-x^2-y^2-xy}$ 的（局部）最小值点（曲面的最低点）。

```
>> fh = @(x)(x(1).^2-2*x(1)).*exp(-x(1).^2-x(2).^2-x(1).*x(2));
>> x0 = [1,1]; %通过图形以及等高线图观察得[1,1]附件有局部最小值点
>> options = optimset('Display','iter','PlotFcns',@optimplotfval);
>> [x,fval,exitflag,output] = fminsearch(fh,x0,options)
 Iteration   Func-count     min f(x)         Procedure
     …           …              …                …
     35          68         -0.641424        contract inside
     36          70         -0.641424        contract inside
优化已终止:
 当前的 x 满足使用 1.000000e-12 的 OPTIONS.TolX 的终止条件,
F(X) 满足使用 1.000000e-04 的 OPTIONS.TolFun 的收敛条件
xmin =
    0.6110   -0.3055
fval =
   -0.6414
```



4. 多元函数的极小值案例

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

例：求二元函数 $z=(x^2-2x)e^{-x^2-y^2-xy}$ 的（局部）最小值点（曲面的最低点）。

funwithgrad.m

```
function [f,g] = funwithgrad(x)
% Calculate objective f
f = (x(1).^2-2*x(1)).*exp(-x(1).^2-x(2).^2-x(1).*x(2));
if nargout > 1 % gradient required
    g = [exp(- x(1).^2 - x(1).*x(2) - x(2).^2)*(2*x(1) - 2) + ...
         exp(- x(1).^2 - x(1).*x(2) - x(2).^2)*(- x(1).^2 + 2*x(1)).*(2*x(1) + x(2));
         exp(- x(1).^2 - x(1).*x(2) - x(2).^2)*(- x(1).^2 + 2*x(1)).*(x(1) + 2*x(2))];
end
```

信赖域算法，使用函数定义中的梯度

```
>> fh = @funwithgrad;
>> options = optimoptions('fminunc','Algorithm','trust-region','SpecifyObjectiveGradient',true);
>> x0 = [1,1];  %通过图形以及等高线图观察得[1,1]附件有局部最小值点
>> [x,fval,exitflag,output] = fminunc(fh,x0,options)

Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.

<stopping criteria details>
```

```
x =
    0.6110   -0.3055
fval =
   -0.6414
exitflag =
     1
output =
  包含以下字段的 struct:

         iterations: 8
          funcCount: 9
           stepsize: 8.8014e-07
       cgiterations: 6
      firstorderopt: 1.8086e-12
          algorithm: 'trust-region'
            message: 'Local minimum found. …'
    constrviolation: []
```

4. 多元函数的极小值案例

例：求二元函数 $z=(x^2-2x)e^{-x^2-y^2-xy}$ 的（局部）最小值点（曲面的最低点）。

funwithgrad.m

```
function [f,g] = funwithgrad(x)
% Calculate objective f
f = (x(1).^2-2*x(1)).*exp(-x(1).^2-x(2).^2-x(1).*x(2));
if nargout > 1 % gradient required
    g = [exp(- x(1).^2 - x(1).*x(2) - x(2).^2)*(2*x(1) - 2) + ...
          exp(- x(1).^2 - x(1).*x(2) - x(2).^2)*(- x(1).^2 + 2*x(1)).*(2*x
          exp(- x(1).^2 - x(1).*x(2) - x(2).^2)*(- x(1).^2 + 2*x(1)).*(x(1)
end
```

输出迭代过程，使用拟牛顿法

```
>> options = optimoptions(@fminunc,'Display','iter','Algorithm','quasi-newton');
>> [x,fval,exitflag,output,gard,Hessian] = fminunc(fh,x0,options)
                                                        First-order
 Iteration  Func-count       f(x)        Step-size       optimality
     0           3         -0.0497871                      0.149
     1          21          -0.399184      4.15165         0.455
     2          30          -0.405878    0.0424109         0.561
     3          33          -0.501668            1         0.523
     4          39          -0.634422     0.400938         0.221
     5          42          -0.641403            1       0.00684
     6          45          -0.641423            1       0.00131
     7          48          -0.641424            1         7e-07
```

```
Computing finite-difference Hessian using objective function.
x =
    0.6110   -0.3055
fval =
   -0.6414
exitflag =
     1
output =
  包含以下字段的 struct:

       iterations: 7
        funcCount: 48
         stepsize: 0.0012
     lssteplength: 1
    firstorderopt: 7.0035e-07
        algorithm: 'quasi-newton'
          message: 'Local minimum found. …'
gard =
   1.0e-06 *
   -0.7004
   -0.3055
Hessian =
    3.3326    0.6414
    0.6414    1.2828
```

4. 多元函数的极小值案例

例：设 $f(x,y,z)=x+\dfrac{y^2}{4x}+\dfrac{z^2}{y}+\dfrac{2}{z}$ ，求函数在(0.5,0.5,0.5) 附近的最小值。

```
>> fh = @(x)x(1)+x(2).^2./(4.*x(1)) + x(3).^2./x(2)+2./x(3);
>> x0 = [0.5,0.5,0.5];
>> [xx,fval,exitflag] = fminsearch(fh,x0)
xx =
    0.5000    1.0000    1.0000
fval =
    4.0000
exitflag =
     1
```

```
%首先建立函数文件f2min.m：
function fh = f3min(u)
   x=u(1); y=u(2); z = u(3);
   fh = x+y.^2./x/4+z.^2./y+2./z ;
return
%在命令窗口输入：
[X,fval] = fminsearch('fh',[0.5,0.5,0.5])   %求函数的最小值点
X =
      0.5000      1.0000   1.0000
fval =
      4.0000
```


01
IDEA
02
LEARNING
03
CREATIVE
INSPIRATION
EDUCATION
BUSINESS
PHOTO BOOK

感谢聆听