静态优化算法及软件实现之一

第3讲　无约束优化

李传江

2019. 7. 8

主要内容

无约束优化

1. 问题实例
2. 数学描述
3. 求解方法
4. **MATLAB** 软件求解
5. **MATHEMATICA** 软件求解

无约束优化问题实例

选址问题：

某投资商欲建一个超市，周围5个关注的居民区的坐标分别为$(10,0),(-10,0),(0,10),(0,-10),(5,5)$，试确定超市坐标使其到5个关注点距离之和最小。

解： 设超市坐标为(x,y)，容易写出本例的优化模型为：

$$\begin{aligned}\min f(x,y) = &\sqrt{(x-10)^2+y^2}+\sqrt{(x+10)^2+y^2} \\ &+\sqrt{x^2+(y-10)^2}+\sqrt{x^2+(y+10)^2} \\ &+\sqrt{(x-5)^2+(y-5)^2}\end{aligned}$$

数 学 描 述

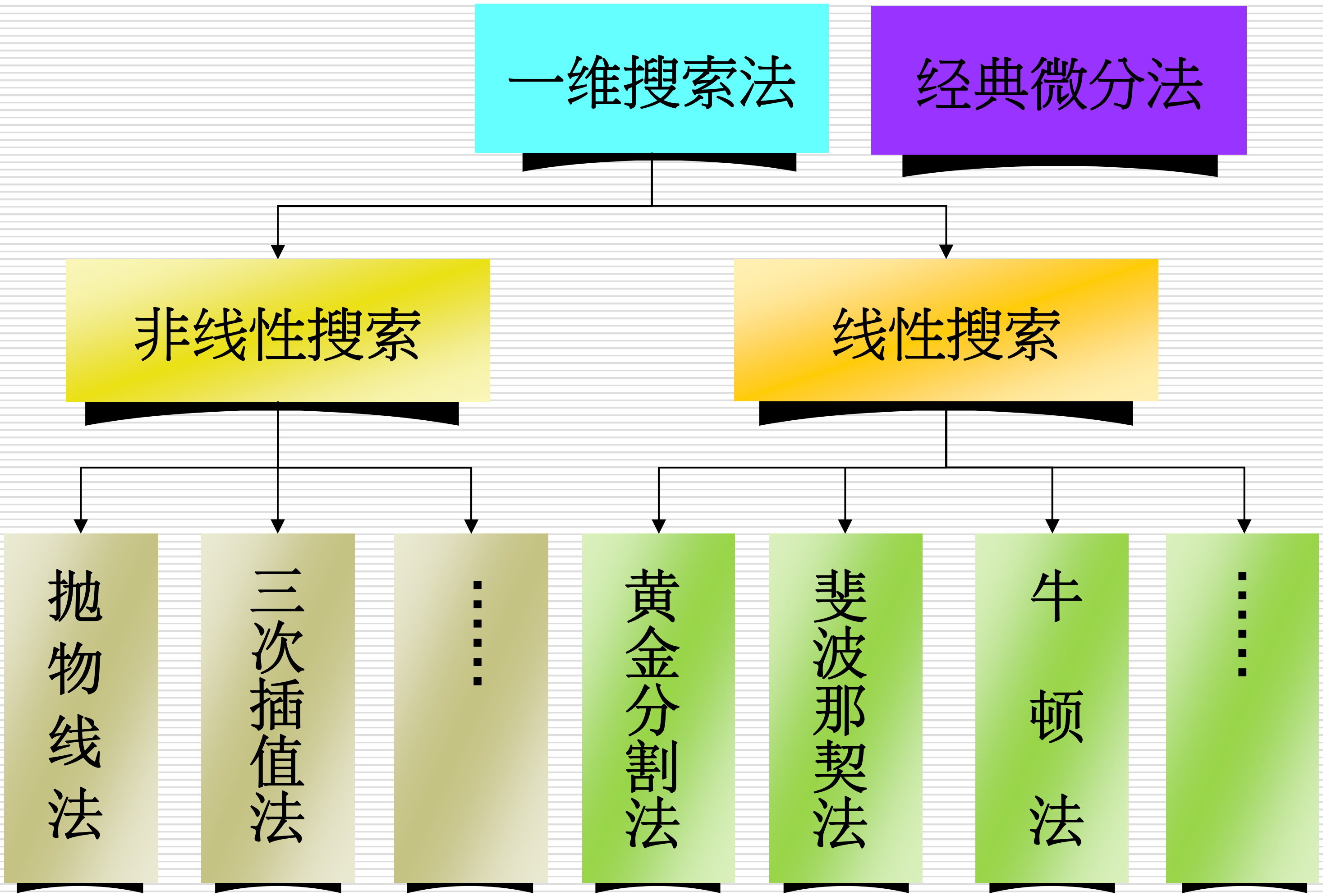
无约束一维优化问题可以描述为:

$$\min f(x), \qquad x \in \mathcal{R} \text{ or } [x_1, x_2]$$

无约束多维优化问题可以描述为:

$$\min f(x),\ x \in \mathcal{R}^n$$

一维优化问题求解方法



一维优化问题求解方法

经典微分法

函数$f(x)$在x^o处存在局部极小值的必要条件为：

$$f'(x^o) = \mathrm{d}f(x^o)/\mathrm{d}x = 0$$

或者

$$\begin{cases} f^l(x^o) = \mathrm{d}^l f(x^o)/\mathrm{d}x^l = 0, (l = 1, \cdots, 2k-1) \\ f^{2k}(x^o) \geq 0 \ (k = 1, \cdots) \end{cases}$$

一维优化问题求解方法

经典微分法

函数$f(x)$在x^o处存在局部极小值的充分条件为：

$$f'(x^o) = \mathrm{d}f(x^o)/\mathrm{d}x = 0,\ f''(x^o) > 0$$

或者

$$\begin{cases} f^l(x^o) = \mathrm{d}^l f(x^o)/\mathrm{d}x^l = 0, (l = 1, \cdots, 2k-1) \\ f^{2k}(x^o) > 0\ (k = 1, \cdots) \end{cases}$$

一维优化问题求解方法

例1： 已知$f(x)=(3-2x)^2x$，求函数在区间$(0,2)$上的极值。

[解] 计算$f(x)$的一阶和二阶导数得

$$f'(x)=3(3-2x)(1-2x),\ f''(x)=24(x-1)$$

当$x=1.5$或0.5时，

$$f'(x)=0,\ f''(1.5)=12>0,\ f''(0.5)=-12<0$$

结论：在区间$x\in(0,2)$中，$x=0.5$是局部极大值点，$x=1.5$是局部极小值点。

一维优化问题求解方法

牛顿法

基本思想是：用$f(x)$在已知点x_0处的二阶Taylor展开式来近似代替$f(x)$，即有$f(x) \approx g(x) = f(x_0) + f'(x_0)(x - x_0) + 0.5f''(x_0)(x - x_0)^2$，接着用$g(x)$的极小值点作为$f(x)$的近似极小值点。

$g(x)$的极小值点为

$$x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)}$$

一维优化问题求解方法

牛 顿 法

若已知x_k点，则有迭代公式

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}, \qquad k = 0,1,2,\cdots$$

可以计算出下一个点x_{k+1}，从而形成一个序列$\{x_k\}$，当$f'(x_k) < \varepsilon$时（其中ε为设定的计算精度），则迭代结束，函数$f(x)$的最优解$x^* \approx x_k$

一维优化问题求解方法

例2： 求函数$f(x)=x^4-4x^3-6x^2-16x+4$的极小值。

[解]由微分法易求$f(x)$在$x=4$时达到极小值$f_{\min}=-156$。

取$x_0=8,\varepsilon=3\times10^{-3}$，则有如下迭代

$$x_1=x_0-\frac{f'(x_0)}{f''(x_0)}=8-\frac{f'(8)}{f''(8)}=5.92908$$

$$x_2=x_1-\frac{f'(x_1)}{f''(x_1)}=4.71538, f'(x_1)=324.726$$

$$x_3=x_2-\frac{f'(x_2)}{f''(x_2)}=4.15074, f'(x_2)=79.9793$$

$$x_4=x_3-\frac{f'(x_3)}{f''(x_3)}=4.00889, f'(x_3)=13.4940$$

$$x_5=x_4-\frac{f'(x_4)}{f''(x_4)}=4.00003, f'(x_4)=0.749392$$

$$x_6=x_5-\frac{f'(x_5)}{f''(x_5)}=4.00000, f'(x_5)=0.00283<\varepsilon$$

一维优化—MATLAB软件求解

Matlab提供专有函数**fminbnd()**求解无约束一维优化问题：

$$\min f(x)$$
$$\text{s.t.}\ x \in (x_1, x_2)$$

函数fminbnd()多种不同调用和说明：

- **x = fminbnd (fun, x1, x2)**

返回标量函数fun在满足x1<x<x2条件下取极小值时的优化变量x的值。关于函数fun的定义，它可以是一个在M函数中定义的函数句柄，例如

x = fminbnd (@myfun, x1, x2)

一维优化—MATLAB软件求解

其中M函数文件myfun.m具有下面的形式：

```
function f = myfun(x)
f = …; %目标函数定义
```

除此，fun还可以定义成一个匿名函数句柄或字符串，例如

```
x = fminbnd (@(x)sin(x)*exp(-x), x1, x2)
x = fminbnd ('sin(x)*exp(-x)', x1, x2)
```

- **x = fminbnd (fun, x1, x2, options)**

按options指定的优化参数选项进行目标函数最优解的求取。可通过optimset函数来设置。

一维优化—MATLAB软件求解

参数名称	参数设置
Display	为off时，不显示任何输出信息；为iter时，显示每一步迭代输出信息；为final时，仅显示最终输出信息；为notify时，只有当求解不收敛时才显示结果
FunValCheck	检查目标函数值是否合法。为on时，若目标函数值为复数，Inf或者NaN，返回错误信息；为off时，不返回该错误信息
MaxFunEvals	目标函数的最大评价次数，默认值为500
MaxIter	算法运行中的最大迭代次数，默认值为500
PlotFcns	算法执行过程中按不同标准绘制优化进程，默认值为[]，若设置为'optimplotx'，则绘制迭代过程中的当前点；若设置为'optimplotfval'，则绘制目标函数值
TolX	最优解处的误差容限

一维优化—MATLAB软件求解

- **[x, fval] = fminbnd (…)**

优化完后返回最优解x及最优解处对应的目标函数值fval。

- **[x, fval, exitflag] = fminbnd (…)**

返回优化问题的状态指示exitflag，说明算法终止原因。

exitflag	物理意义
1	已经收敛到满足设定精度的最优解x
0	已经达到最大迭代次数限制options.MaxIter或者已经达到函数评价次数的最大允许值options.FunEvals，退出
-1	由输出函数引起的算法终止
-2	优化变量的边界是矛盾的

一维优化—MATLAB软件求解

例3：用fminbnd(·)求函数$f(x)=x^4-4x^3-6x^2-16x+4$的极小值。

[解] MATLAB计算程序如下：

```
fun = @(x)x^4-4*x^3-6*x^2-16*x+4;
options = optimset ('Display','iter');
[x, fmin, exitflag, output] = fminbnd (fun,-2,12,options)
```

一维优化—MATLAB软件求解

Func-count	x	f(x)	Procedure
1	3.34752	-141.272	initial
2	6.65248	412.938	golden
3	1.30495	-33.0856	golden
4	2.96804	-123.326	parabolic
5	4.6099	-137.516	golden
6	3.93012	-155.799	parabolic
7	3.94248	-155.863	parabolic
8	4.01478	-155.991	parabolic
9	4.2421	-153.365	golden
10	4.00031	-156	parabolic
11	3.99988	-156	parabolic
12	4	-156	parabolic
13	3.99997	-156	parabolic
14	4.00003	-156	parabolic

Optimization terminated:

the current x satisfies the termination criteria using OPTIONS.TolX of 1.0000e-04

x = 4.0000

fmin = -156.0000

exitflag = 1

output = iterations: 13

funcCount: 14

algorithm: 'golden section search, parabolic interpolation'

message: [1x111 char]

一维优化—MATLAB软件求解

fminbnd(·)函数在求取极值的迭代过程中无需用到目标函数的导数，而只需要函数值连续即可，故该函数还可以求取含有绝对值函数的极值。

例4： 用fminbnd(·)求函数$f(x)=e^{-|x|}\sin(2x)$在区间$(-2,2)$上的极值。

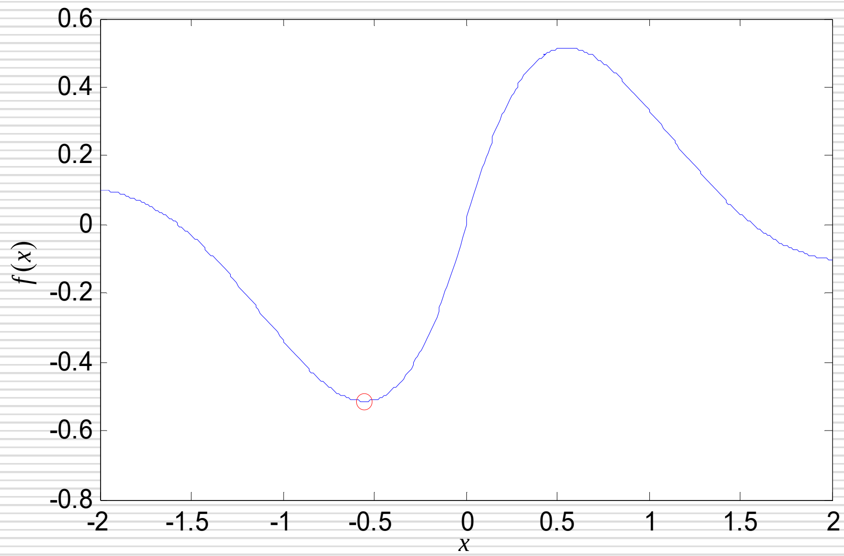
[解] MATLAB计算程序如下：

```
fun = @(x)exp(-abs(x))*sin(2*x);
options = optimset ('Display','iter');
[x, fmin, exitflag] = fminbnd (fun,-2,2,options)
```

一维优化—MATLAB软件求解

Func-count	x	f(x)	Procedure
1	-0.472136	-0.505215	initial
2	0.472136	0.505215	golden
3	-1.05573	-0.298312	golden
4	-0.573815	-0.513679	parabolic
5	-0.56879	-0.513904	parabolic
6	-0.554402	-0.514198	parabolic
7	-0.553459	-0.514198	parabolic
8	-0.553571	-0.514198	parabolic
9	-0.553604	-0.514198	parabolic
10	-0.553537	-0.514198	parabolic

Optimization terminated:
the current x satisfies the termination criteria
using OPTIONS.TolX of 1.000000e-04
x = -0.5536
fmin = -0.5142
exitflag = 1



函数$f(x)$在区间(-2,2)上存在两个极值点，一个极大值点，一个极小值点，fminbnd函数成功地求取了极小值点。

一维优化—MATHEMATICA软件求解

求解无约束一维优化问题还可以采用MATHEMATICA软件提供的FindMinimum函数、FindMaximum函数、Minimize函数、Maximize函数、NMinimize函数及NMaximize函数等。

FindMinimum函数几种常用调用格式及相关说明如下：

- **FindMinimum** $[f,\ x]$
  从一个自动选定的初始点开始，搜索f的局部极小值。
- **FindMinimum** $[f,\ \{x, x_0\}]$
  以$x = x_0$为初始值，搜索函数f的局部极小值。
- **FindMinimum** $[f,\ \{\{x, x_0\}, \{y, y_0\}, \cdots\}]$
  搜索多元函数$f(x, y, \cdots)$的局部极小值。
- **FindMinimum** $[\{f, Cons\},\ \{\{x, x_0\}, \{y, y_0\}, \cdots\}]$
  搜索多元函数$f(x, y, \cdots)$ 在约束条件Cons下的局部极小值。

一维优化—MATHEMATICA软件求解

1. FindMinimum返回$\{f_{\min}, \{x \to x_{\min}, y \to y_{\min}, \cdots\}\}$的列表
2. Cons可以包含方程/等式约束/不等式约束/逻辑组合约束
3. FindMinimum $[f, \{x, x_0, x_{\min}, x_{\max}\}]$ 表示如果x超出了从$x_{\min}$到$x_{\max}$的范围，则停止继续搜索
4. 除了当f和Cons都是线性的，FindMinimum的结果可能是局部最优的
5. 可以通过$x \in$ Integers或Element$[x,$ Integers$]$来指定整数变量

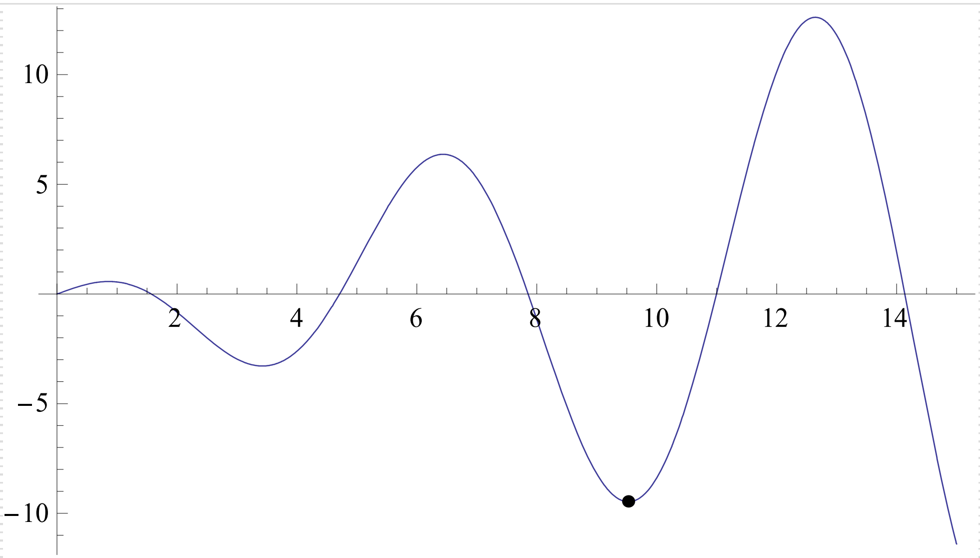
一维优化—MATHEMATICA软件求解

例5： 求函数 $f = x\cos x$ 在 $x = 10$ 附近的局部极小值。

$\text{In}[1] = \textbf{FindMinimum}[x\text{Cos}[x], \{x, 10\}]$

$\text{Out}[1] = \{-9.47729, \{x \to 9.52933\}\}$

$\text{In}[2] = \textbf{Plot}[x\text{Cos}[x], \{x, 0{,}15\}]$



一维优化—MATHEMATICA软件求解

例6： 求函数 $f(x)=x^4-2x^2+x+3$在区间$[-2,2]$上的局部极小值，初始值分别取为$x_0=1.5$和$x_0=-0.5$。

$\mathrm{In}[3]=\mathrm{FindMinimum}[\{x^4-2x^2+x+3,-2\leq x\leq 2\},\{x,1.5\}]$

$\mathrm{Out}[3]=\{2.92666,\{x\to 0.837566\}\}$

$\mathrm{In}[4]=\mathrm{FindMinimum}[\{x^4-2x^2+x+3,-2\leq x\leq 2\},\{x,-0.5\}]$

$\mathrm{Out}[4]=\{0.943827,\{x\to -1.10716\}\}$

$\mathrm{In}[5]=\mathrm{NMinimize}[x^4-2x^2+x+3,x]$

$\mathrm{Out}[5]=\{0.943827,\{x\to -1.10716\}\}$

一维优化—MATHEMATICA软件求解

例7： 求函数$f(x)=e^{-|x|}\sin(2x)$在区间$(-2,2)$上的极小值。

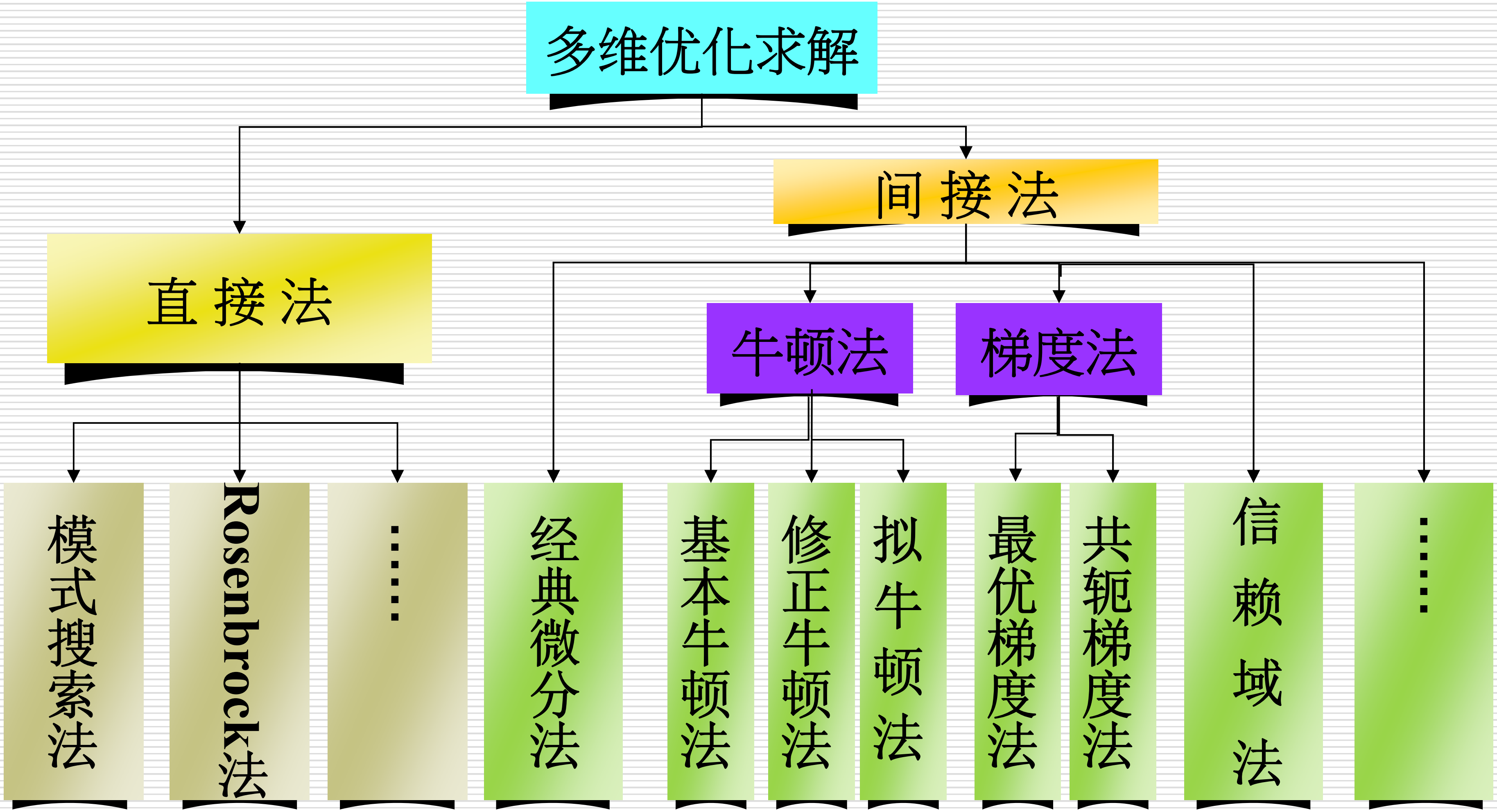
$\text{In[6]} = \text{FindMinimum}[\mathrm{e}^{-\text{Abs}[x]}\text{Sin}[2x],\{x,-2,2\}]$

$\text{Out[6]} = \{-0.514198,\{x\to -0.553574\}\}$

$\text{In[7]} = \text{NMinimize}[\mathrm{e}^{-\text{Abs}[x]}\text{Sin}[2x],x]$

$\text{Out[7]} = \{-0.514198,\{x\to -0.553574\}\}$

多维优化问题求解方法


多维优化求解
间接法
直接法
牛顿法
梯度法
模式搜索法
Rosenbrock法
……
经典微分法
基本牛顿法
修正牛顿法
拟牛顿法
最优梯度法
共轭梯度法
信赖域法
……

多维优化问题求解方法 经典微分法

函数$f(x)$在$x^* = [x_1^*, x_2^*, \cdots x_n^*]^T$处存在局部极小值的充分条件为：

$$\partial f(x^*)/\partial x = \nabla f(x^*) = \begin{bmatrix} \partial f(x^*)/\partial x_1 \\ \partial f(x^*)/\partial x_2 \\ \vdots \\ \partial f(x^*)/\partial x_n \end{bmatrix} = 0$$

$$\partial^2 f(x^*)/\partial x^2 = H(x^*) = \begin{bmatrix} \partial^2 f/\partial x_1^2 & \partial^2 f/\partial x_1 \partial x_2 & \cdots & \partial^2 f/\partial x_1 \partial x_n \\ \partial^2 f/\partial x_2 \partial x_1 & \partial^2 f/\partial x_2^2 & \cdots & \partial^2 f/\partial x_2 \partial x_n \\ \vdots & \vdots & \ddots & \vdots \\ \partial^2 f/\partial x_n \partial x_1 & \partial^2 f/\partial x_n \partial x_2 & \cdots & \partial^2 f/\partial x_n^2 \end{bmatrix} > 0$$

其中$\nabla f(x^*)$为函数$f(x)$在x^*点处的梯度向量；

H(x^*)为函数$f(x)$在x^*点处的汉森**(Hessian)**矩阵；

当汉森矩阵为负定时，$f(x)$在x^*点处取得局部极大值。

多维优化问题求解方法

经典微分法

例8： 求函数$f(x)=4x_1^2+2x_2^2+2x_1x_2-x_1-2x_2$的极小值。

[解]目标函数的一阶和二阶偏导数可以计算为

$$\nabla f(x)=\begin{bmatrix}8x_1+2x_2-1\\2x_1+4x_2-2\end{bmatrix},H(x)=\begin{bmatrix}8&2\\2&4\end{bmatrix}$$

令$\nabla f(x^*)=0$，得到$x^*=[0\ \ 0.5]^T$，再由$H(x^*)>0$可知，函数$f(x)$在点$x^*=[0\ \ 0.5]^T$处取局部极小值，且极小值为$f(x^*)=-0.5$。

多维优化问题求解方法

最优梯度法

19世纪中叶，法国科学家Cauchy指出：从任意初始搜索点$x^0 \in \mathcal{R}^n$出发，函数$f(x)$沿该点的负梯度方向，函数值下降最快。基于这一思想，Cauchy于1874年提出了最速下降法。该方法是众多最优化方法的最基本方法之一。

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} \cdot \nabla f(x^{(k)})$$

其中$-\nabla f(x^*)$为**搜索方向**；$\alpha^{(k)}$为**搜索步长**。

$$\alpha^{(k)} = \frac{\nabla f(x^{(k)})^T \cdot \nabla f(x^{(k)})}{\nabla f(x^{(k)})^T \cdot H(x^{(k)}) \cdot \nabla f(x^{(k)})}$$

多维优化问题求解方法

牛顿法

一维优化问题的迭代公式：

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}, \qquad k = 0,1,2,\cdots$$

多维优化问题时的迭代公式如下：

$$\begin{aligned} x_{k+1} &= x_k - \frac{\nabla f(x_k)}{\nabla^2 f(x_k)} \\ &= x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k), k = 0,1,2,\cdots \end{aligned}$$

可以看出，牛顿法是以$-[\nabla^2 f(x_k)]^{-1}\nabla f(x_k)$作为搜索方向的，且迭代步长在整个迭代过程中均取为1。

多维优化—MATLAB软件求解

Matlab提供专有函数**fminsearch()**求解无约束多维优化问题：

$$\min f(x),\ x \in \mathcal{R}^n$$

函数fminsearch()多种不同调用和说明：

- **x = fminsearch (fun, x0)**

  从初始搜索点x0出发，求出函数fun的一个局部极小值点；

- **x = fminsearch (fun, x0, options)**

  按options指定的优化参数选项进行目标函数优化，其参数指定方式同前述介绍的fminbnd函数；

多维优化—MATLAB软件求解

- **[x, fmin] = fminsearch (⋯)**

优化完后返回最优解x及最优解处对应的目标函数值fmin。

- **[x, fmin, exitflag] = fminsearch (⋯)**

返回优化问题的状态指示exitflag，说明算法终止原因。

exitflag	物理意义
1	已经收敛到满足设定精度的最优解x
0	已经达到最大迭代次数限制options.MaxIter或者已经达到函数评价次数的最大允许值options.FunEvals，退出
-1	由输出函数引起的算法终止

多维优化—MATLAB软件求解

例9： 用fminsearch(·)求函数$f(x) = u^2 + 2v^2 - 2uv - 4u$的极小值点，初始搜索点为$x_0 = [1\ \ 1]^T$。

[解] MATLAB计算程序如下：

```
x0 = [1;1];
fun = @(x)x(1)^2+2*x(2)^2-2*x(1)*x(2)-4*x(1);
[x, fmin, exitflag] = fminsearch (fun,x0)
```

返回迭代结果如下：

```
x = 4.0000
    2.0000
fmin = -8.0000
exitflag = 1
```

多维优化—MATLAB软件求解

例10： 用fminsearch(·)求函数$f(x,y)=x^2+2.5\sin y+2e^{x+y}$的最优解，初始搜索点为$x_0=[-1\quad -1]^T$。

[解] MATLAB计算程序如下：

```
x0 = [-1;-1];
fun = @(x)x(1)^2+2.5*sin(x(2))+2*exp(x(1)+x(2));
[x, fmin, exitflag] = fminsearch (fun,x0)
```

返回迭代结果如下：

```
x = -0.1567
      -1.6965
fmin = -2.1422
exitflag = 1
```

多维优化—MATLAB软件求解

Matlab还可采用函数**fminunc()**求解无约束多维优化问题，fminsearch(·)函数采用的算法是不基于梯度信息的单纯形算法，而fminunc(·)函数则是采用基于梯度信息或者Hessian矩阵信息的间接方法来完成求解的。

函数fminunc()多种不同调用和说明：

- **x = fminunc (fun, x0)**

  从初始搜索点x0出发，求出函数fun的一个局部极小值点；

- **x = fminunc (fun, x0, options)**

  按options指定的优化参数选项进行目标函数优化，其参数指定方式同前述介绍的fminbnd函数；

多维优化—MATLAB软件求解

- [x, fmin] = fminunc (…)

优化完后返回最优解x及最优解处对应的目标函数值fmin。

- [x, fmin, exitflag] = fminunc (…)

返回优化问题的状态指示exitflag，说明算法终止原因。

exitflag	物理意义
1	梯度的模小于函数计算终止的误差限TolFun
2	x的变化量小于x处的误差限TolX
3	目标函数值的变化小于函数计算终止的误差限TolFun
5	目标函数值的预计下降量小于TolFun
0	已经达到最大迭代次数限制或者达到函数评价次数的最大允许值
-1	由输出函数引起的算法终止
-3	目标函数值无界

多维优化—MATLAB软件求解

- **[x, fmin, exitflag, output] = fminunc (···)**

属性名称	属性含义
output.iterations	优化过程的实际迭代次数
output.algorithm	优化过程采用的具体算法
output.funcCount	目标函数的评价次数
output.firstorderopt	目标函数的一阶最优梯度
output.cgiterations	PCG迭代的次数（仅对大型规模算法有效）
output.stepsize	x的最终步长（仅对中型规模算法有效）
output.message	优化过程退出信息

- **[x, fmin, exitflag, output, grad, hess] = fminunc (···)**

  优化计算结束后返回最优解处对应的梯度和Hessian矩阵。

多维优化—MATLAB软件求解

例11：用fminunc(·)求$f(x)=-\frac{3}{(x_1+2)^2+3}-\frac{1}{(x_2-1)^2+0.5}$的最优解。

[解] MATLAB计算程序如下：

```
fun = @(x)-3/((x(1)+2)^2+3)-1/((x(2)-1)^2+0.5);
problem.objective=fun;
problem.x0=[2 -1];
problem.solver='fminunc';
problem.options=optimset('Display','iter');
[x,fmin,exitflag,output,grad,hessian] = fminunc (problem)
```

返回迭代过程和迭代结果如下：

多维优化—MATLAB软件求解

Iteration	Func-count	f(x)	Step-size	First-order optimality
0	3	-0.380117		0.198
1	18	-2.21332	10.1419	0.101
2	24	-2.22827	0.309599	0.561
3	33	-2.85148	0.110169	1.37
4	36	-2.94736	1	0.681
5	42	-2.99567	0.1	0.161
6	45	-2.99999	1	0.00297
7	48	-3	1	0.000755
8	51	-3	1	6.49e-05
9	54	-3	1	9.54e-07

多维优化—MATLAB软件求解

x = -2.0000   1.0000
fmin = -3.0000
exitflag = 1
output = iterations: 9
　　funcCount: 54
　　stepsize: 1
　　firstorderopt: 9.5367e-07
　　algorithm: 'medium-scale: Quasi-Newton line search'
　　message: [1x436 char]
grad =1.0e-06 *
　　-0.3725
　　0.9537
hessian = 0.6667   0
　　0   8.0000

目标函数在$x_1 = -2, x_2 = 1$处达到极小值$f_{\min} = -3$，且目标函数在最优解处的梯度接近于零，此外，Hessian矩阵的正定结果也表明了最优解是一个极小值点。

多维优化—MATHEMATICA软件求解

例12： 求函数$f(x) = 4x_1^2 + 2x_2^2 + 3x_1 + x_2$的极小值点。

$$\mathrm{In}[1] = \textbf{FindMinimum}\left[4x_1^2 + 2x_2^2 + 3x_1 + x_2, \{\{x_1, 1\}, \{x_2, 1\}\}\right]$$

$$\mathrm{Out}[1] = \{-0.6875, \{x_1 \to -0.375, x_2 \to -0.25\}\}$$

$$\mathrm{In}[2] = \textbf{NMinimize}\left[4x_1^2 + 2x_2^2 + 3x_1 + x_2, \{x_1, x_2\}\right]$$

$$\mathrm{Out}[2] = \{-0.6875, \{x_1 \to -0.375, x_2 \to -0.25\}\}$$

$$\mathrm{In}[3] = \textbf{Minimize}\left[4x_1^2 + 2x_2^2 + 3x_1 + x_2, \{x_1, x_2\}\right]$$

$$\mathrm{Out}[3] = \{-11/16, \{x_1 \to -3/8, x_2 \to -1/4\}\}$$