回顾

第8章 数值优化

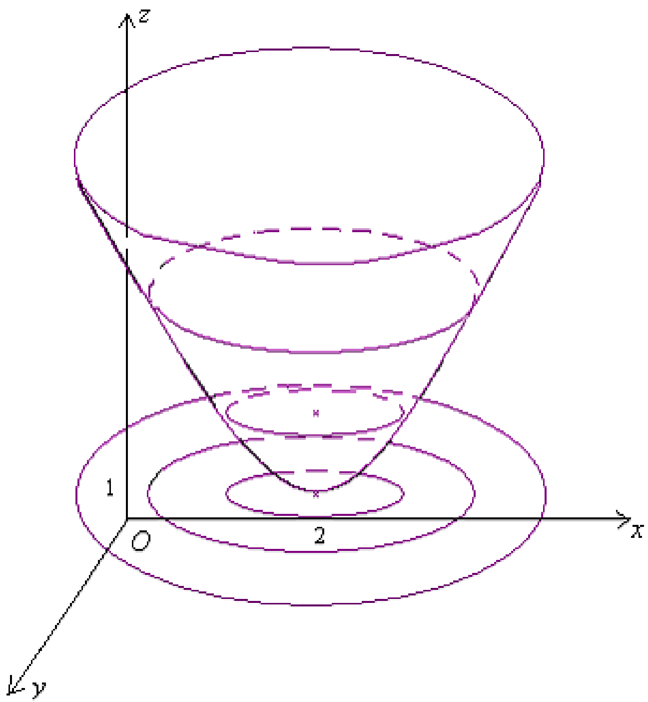
什么是最优化

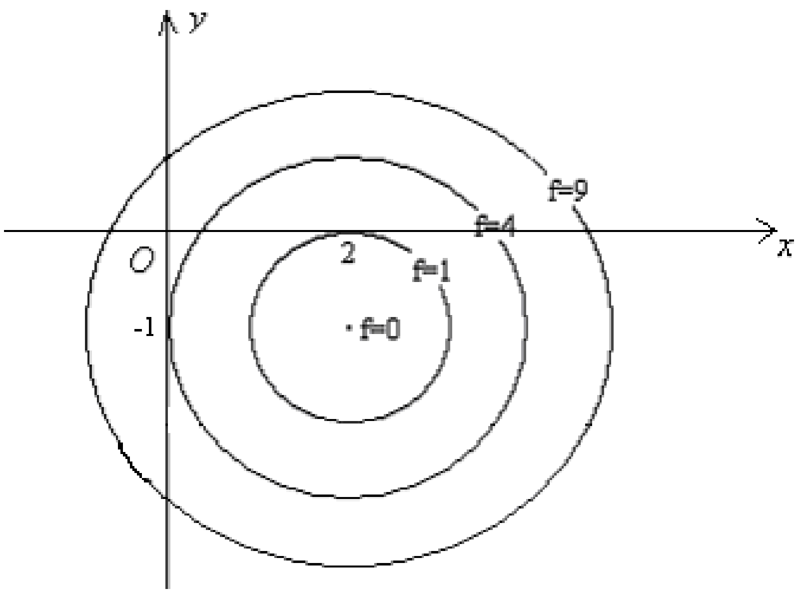
回顾

例8.1 二维问题图解法

二维极值问题有时可以用图解的方式进行求解，有明显的几何解释。

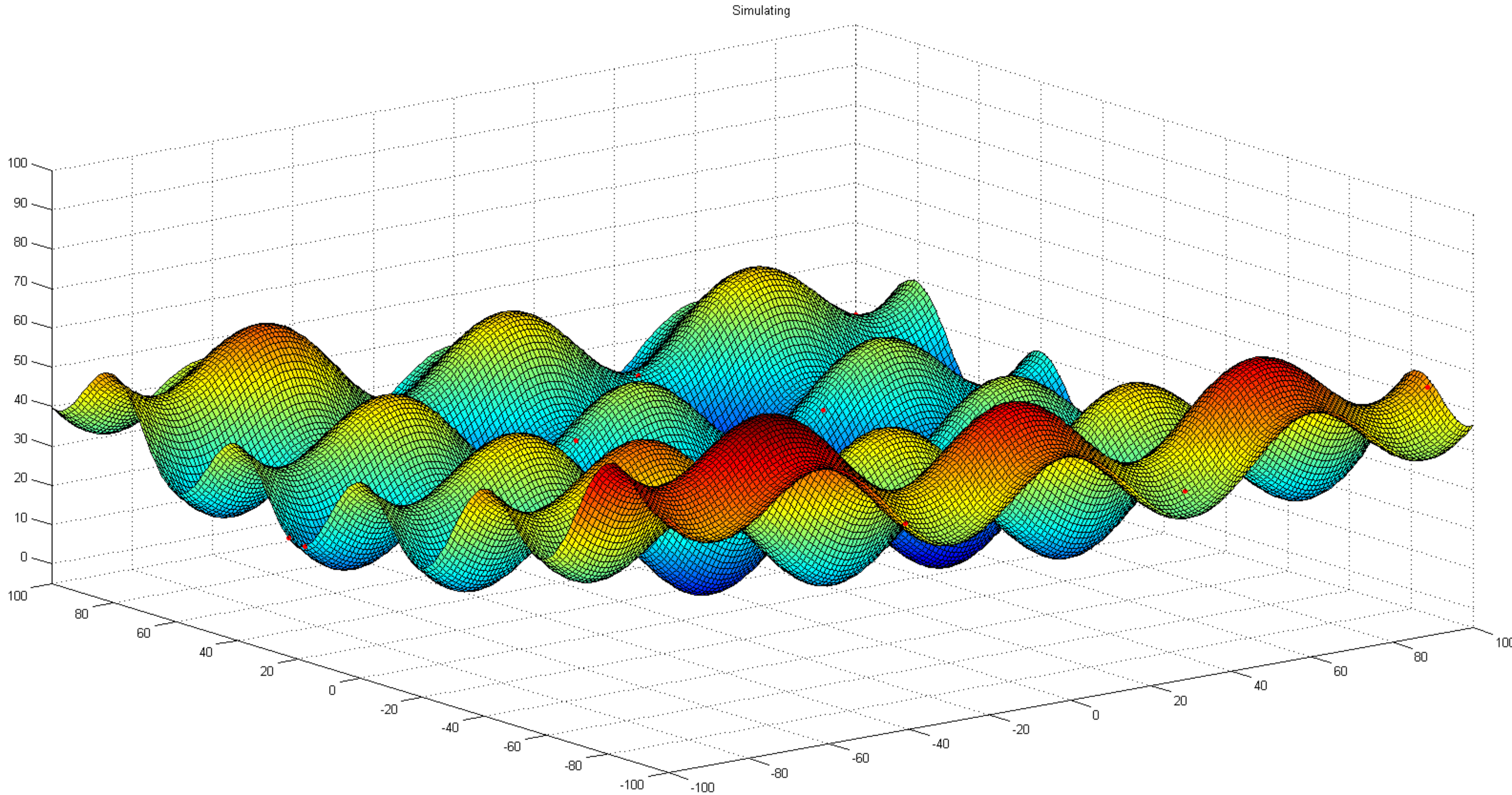
例 求解 $\min f(x,y)=(x-2)^2+(y+1)^2$





例8.2

回顾



8.2 单变量函数的极小值

回顾

例8.3 利用二阶导数测试，对函数 $f(x)=x^3+x^2-x+1$ 在区间[-2,2]上的局部极值进行分类。

解：一阶导数为 $f'(x)=3x^2+2x-1=(3x-1)(x+1)$，二阶导数为 $f''(x)=6x+2$。有两个点满足 $f'(x)=0$，即 $x=1/3, x=-1$。

情况(i)：在 $x=1/3$ 处，$f'(1/3)=0$，而 $f''(1/3)=4>0$，因此在 $x=1/3$ 处 $f(x)$ 有一个局部极小值。

情况(ii)：在 $x=-1$ 处，$f'(-1)=0$，而 $f''(-1)=-4<0$，因此在 $x=-1$ 处 $f(x)$ 有一个局部极大值。

■

上述求解极值的方法，称为解析法。

在经典极值问题中，解析法虽然具有概念简明，计算精确等优点，但因只能适用于简单或特殊问题的寻优，对于复杂的工程实际问题由于目标函数不可导，或其导数的求解过程非常复杂，此时，解析法就会无能为力，所以极少使用。

8.2.2 分类搜索方法

回顾

- 另外一种求解极小值的方法，称为迭代法，它是一种数值方法。
- 其基本思想是：从某一选定的初始点出发，根据目标函数、约束函数在该点的某些信息，确定本次迭代的一个搜索方向和适当的步长，并通过迭代产生一个点序列$\{X^{(k)}\}$，使之逐步接近最优点。

优点：它只用到目标函数，通过对函数多次求值来求函数$f(x)$在给定区间上的一个局部极小值,可用于$f(x)$不可微的情况。

要求:要尽量减少函数求值的次数，确定在哪里求$f(x)$值的好策略非常重要。如黄金分割搜索法、Fibonacci搜索法

使用这些方法来求$f(x)$的极小值必须满足特定的条件，以保证在给定的区间内有合适的极小值

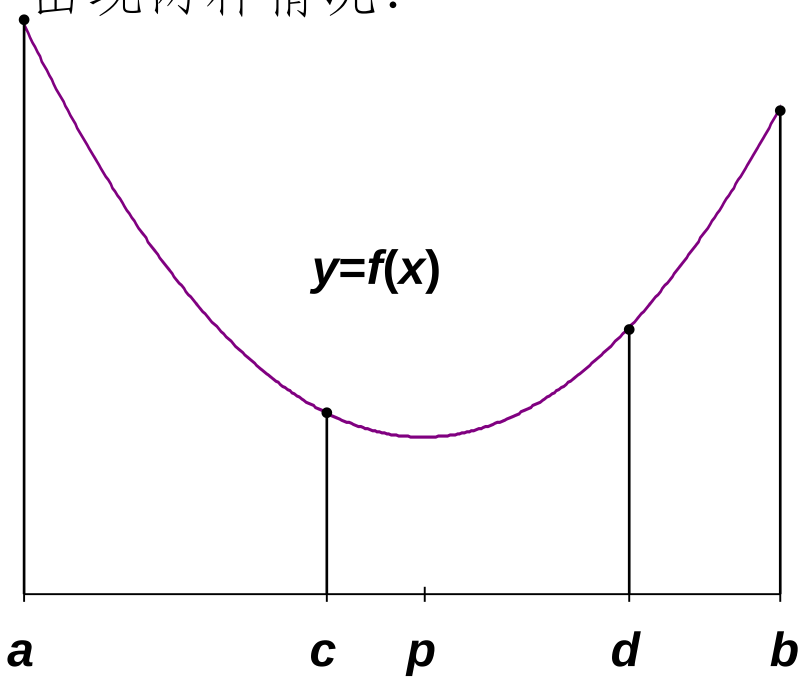
（1）黄金分割搜索法（0.618法）

回顾

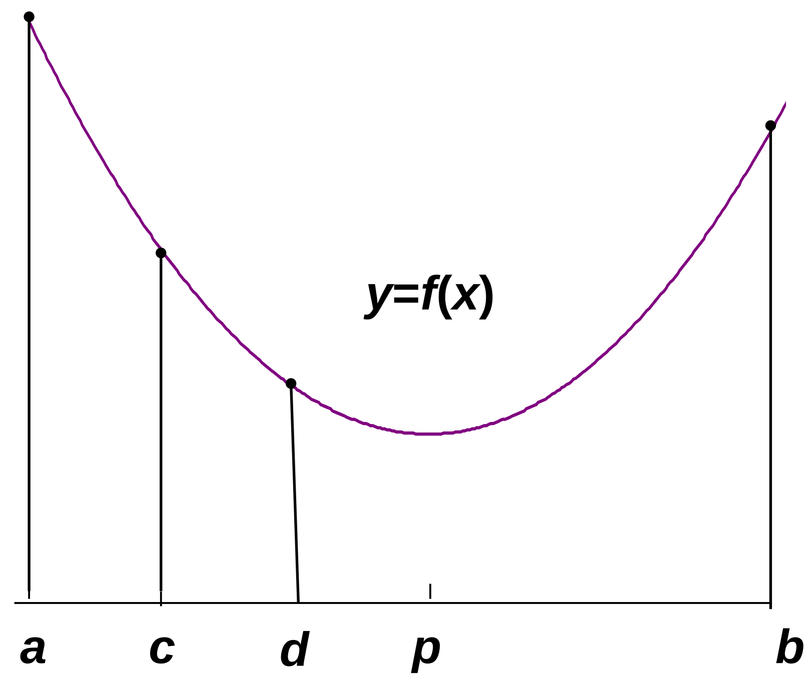
- 如果已知$f(x)$在$[a, b]$上是（下）单峰的，则该方法的目的是找到该区间的一个子区间，使得$f(x)$在该子区间上取得极小值。
- 选择两个内点$c<d$，这样就有$a<c<d<b$。

$f(x)$的单峰特性保证了函数值$f(c)$和$f(d)$小于$\max\{f(a), f(b)\}$

- 出现两种情况：



$f(c)\le f(d)$



$f(c)>f(d)$

- 选择内点c和d，**使得区间[a, c]与[d, b]对称，即b-d=c-a，**其中

$$c=a+(1-r)(b-a)=ra+(1-r)b$$

$$d=b-(1-r)(b-a)=(1-r)a+rb$$

并且$1/2<r<1$（保证$c<d$）

- 希望r在每个子区间上保持为常数，且旧的内点中有一个成为新子区间的一个内点，而另一个则成为新子区间的一个端点(如下图8.3所示)

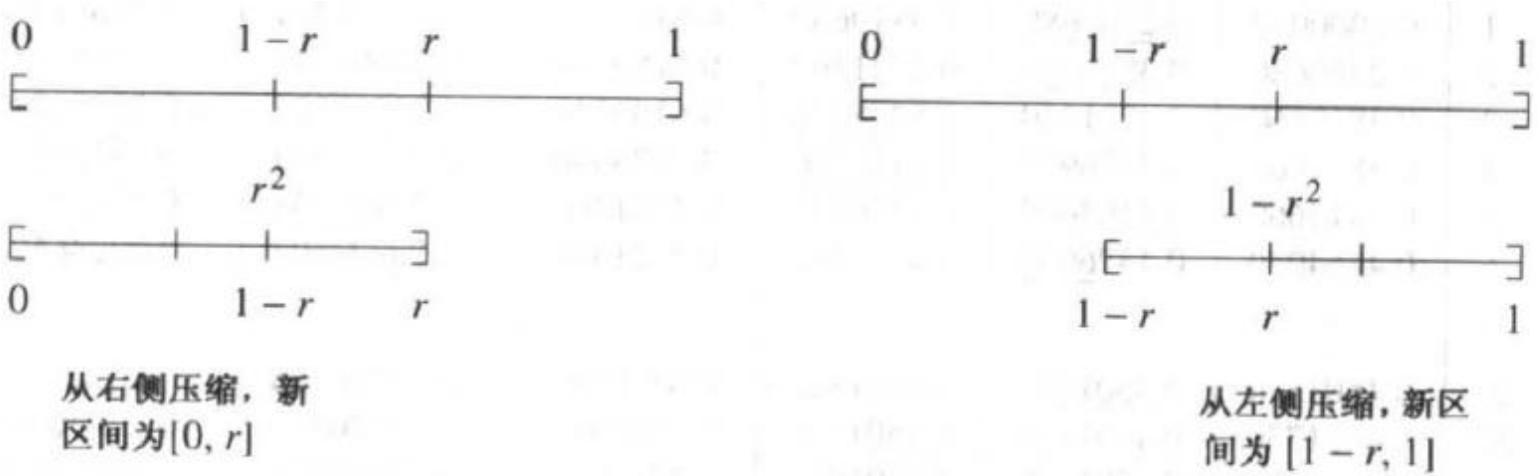


图 8.3　黄金分割搜索方法中的区间

在每次迭代中只需要找一个新的点，则只需要一次新的函数求值计算

比例因子的选择

$c=a+(1-r)(b-a)=ra+(1-r)b$

$d=b-(1-r)(b-a)=(1-r)a+rb$

回顾

假设成立

$$\frac{d-a}{b-a}=\frac{c-a}{d-a}$$

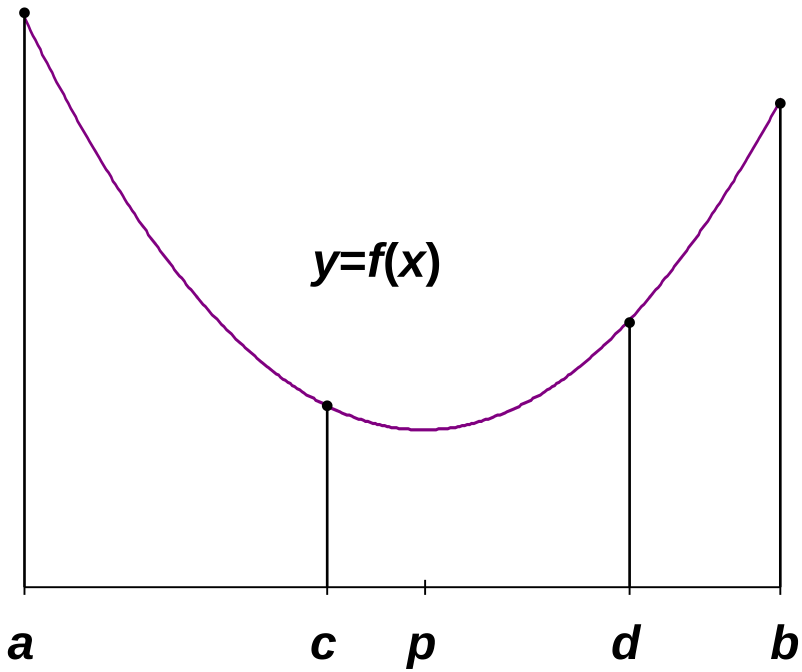
则有

$$\frac{r(b-a)}{b-a}=\frac{(1-r)(b-a)}{r(b-a)}$$

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

$$r^2+r-1=0$$

$$r=\frac{-1\pm\sqrt{5}}{2}$$



如果$f(c)\le f(d)$，则从右侧压缩，使用$[a,d]$

因为$1/2<r<1$（保证$c<d$），故取 $r=\frac{\sqrt{5}-1}{2}$.

给定下单峰区间 $[a, b]$ 及控制误差 $\varepsilon>0$;
黄金分割法(0.618法)的迭代步骤

回顾

①取 $d= a + 0.618\,(b- a)$, $f_2 = f(d)$, 转向②.

②取 $c= a + 0.382\,(b- a)$, $f_1 = f(c)$, 转向③.

③若 $|b - a|<\varepsilon$, 则取 $p= (a + b)/2$, 停. 否则转向④.

④若 $f_1< f_2$, 则取 $b = d$, $d= c$, $f_2= f_1$, 转向②;

若 $f_1= f_2$, 则取 $a = c$, $b = d$, 转向①;

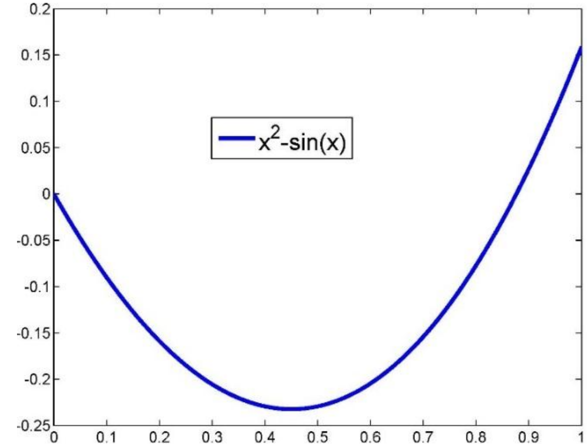
若 $f_1>f_2$, 则取 $a = c$, $c= d$, $f_1 = f_2$, 转向⑤.

⑤取 $d= a + 0.618\,(b- a)$, $f_2= f(d)$, 转向③.

- 第一次迭代中进行了两次函数求值，而在后续的每次迭代中则只进行一次函数求值
- r的值对每个子区间相同，当$|b_k-a_k|<\varepsilon$或$|f(b_k)-f(a_k)|<\varepsilon$时，迭代结束，取$[a_k,b_k]$的中点为所求最小值点。其中$\varepsilon$是预定义的容差.

例8.4 比较求根方法和黄金分割搜索方法

求单峰函数 $f(x) = x^2 - \sin(x)$
在区间[0,1] 上的极小值。



解:通过解 $f'(x)=0$ 求解。可以通过求根方法来确定导数 $f'(x)=2x-\cos(x)$ 在何处为 0。

通过黄金分割搜索方法求解。令 $a_0=0$ 而 $b_0=1$。根据公式(3)和公式(4)分别得到

$$c_0 = 0 + \left(1 - \frac{-1+\sqrt{5}}{2}\right)(1-0) = \frac{3-\sqrt{5}}{2} \approx 0.38919660$$

$$d_0 = 1 - \left(1 - \frac{-1+\sqrt{5}}{2}\right)(1-0) = \frac{-1+\sqrt{5}}{2} \approx 0.6180340$$

计算得 $f(c_0) = -0.22684748$ 和 $f(d_0) = -0.19746793$。由于 $f(c_0) < f(d_0)$，新的子区间为 $[a_0, d_0] = [0.00000000, 0.6180340]$。令 $a_1 = a_0, b_1 = d_0, d_1 = c_0$，并由公式(3)求得 c_1:

$$\begin{aligned} c_1 &= a_1 + (1-r)(b_1 - a_1) \\ &= 0 + \left(1 - \frac{-1+\sqrt{5}}{2}\right)(0.6180340 - 0) \\ &\approx 0.2360680 \end{aligned}$$

计算并比较 $f(c_1)$ 和 $f(d_1)$，以确定新的子空间，并继续迭代过程。表 8.2 列出了部分计算结果。

表 8.2 求 $f(x)=x^2-\sin(x)$ 的极小值的黄金分割搜索法

k	a_k	c_k	d_k	b_k	$f(c_k)$	$f(d_k)$
0	0.0000000	0.3819660	0.6180340	1	−0.22684748	−0.19746793
1	0.0000000	0.2360680	0.3819660	0.6180340	−0.17815339	−0.22684748
2	0.2360680	0.3819660	0.4721360	0.6180340	−0.22684748	−0.23187724
3	0.3819660	0.4721360	0.5278640	0.6180340	−0.23187724	−0.22504882
4	0.3819660	0.4376941	0.4721360	0.5278640	−0.23227594	−0.23187724
5	0.3819660	0.4164079	0.4376941	0.4721360	−0.23108238	−0.23227594
6	0.4164079	0.4376941	0.4508497	0.4721360	−0.23227594	−0.23246503
⋮	⋮	⋮	⋮	⋮	⋮	⋮
21	0.4501574	0.4501730	0.4501827	0.4501983	−0.23246558	−0.23246558
22	0.4501730	0.4501827	0.4501886	0.4501983	−0.23246558	−0.23246558
23	0.4501827	0.4501886	0.4501923	0.4501983	−0.23246558	−0.23246558

在第 23 次迭代时，区间收缩为$[a_{23},b_{23}]=[0.4501827,0.4501983]$。该区间的宽度为 0.0000156。而在该区间的两个端点处求得的函数值在小数点后有 8 位相同，即 $f(a_{23})\approx -0.23246558\approx f(b_{23})$，因此算法结束。搜索法的一个问题是，函数在极小值附近可能比较平缓，从而限制了精度。割线方法能够求得更精确的解 $p_5=0.4501836$。

尽管本例中黄金分割搜索法的速度较慢，但它的优点是可用于 $f(x)$ 不可微的情况。■

（2）斐波那契（Fibonacci）搜索法

在黄金分割搜索法中，第一次迭代中进行了两次函数求值，而在后续的每次迭代中则只进行一次函数求值。r的值对每个子区间相同，当$|b_k-a_k|<\varepsilon$或$|f(b_k)-f(a_k)|<\varepsilon$时，迭代结束.

在此基础上，假设

- **r**的值不是常数，而且子区间数（迭代数）是由指定的容差决定的。

- 斐波那契（Fibonacci）搜索基于公式：

$$F_0=0,\ F_1=1,$$

$$F_n=F_{n-1}+F_{n-2},\qquad n>=2$$

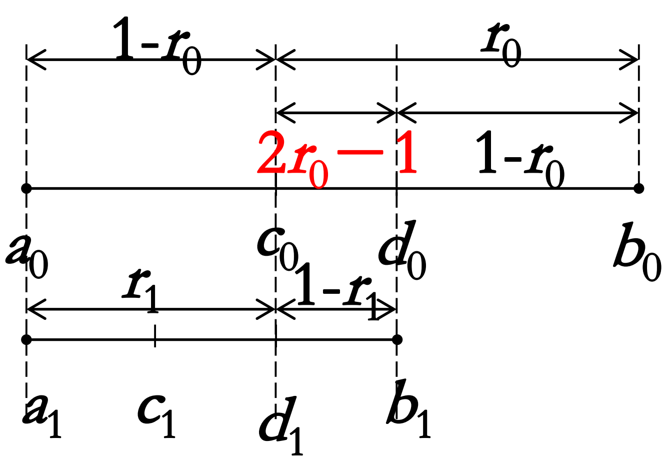
因此，我们有$\{F_k\}=\{0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots\}$

比例因子的确定

设函数 $f(x)$ 在闭区间 $[a_0, b_0]$ 上是单峰函数，选择$1/2<r_0<1$，使内点c_0和d_0可在下一个子区间上使用，从而只需一次新的函数求值计算

设$f(c_0)\leqslant f(d_0)$，则从右侧压缩，使用$[a, d]$，即取$a_1=a_0$，$b_1=d_0$和$d_1=c_0$，则需要再求一个新的点c_1

为子区间$[a_1,b_1]$选择r_1($1/2<r_1<1$)，使得



$$d_0-c_0=b_1-d_1$$

则有

$$(2r_0-1)(b_0-a_0)=(1-r_1)(b_1-a_1)$$

$$(2r_0-1)(b_0-a_0)=(1-r_1)(r_0(b_0-a_0))$$

$$2r_0-1=(1-r_1)r_0$$

$$r_1=\frac{1-r_0}{r_0}$$

由Fibonacci数列，将$r_0=F_{n-1}/F_n$，$n\geqslant 4$代入r_1，得

$$r_1=\frac{1-r_0}{r_0}=\frac{1-\dfrac{F_{n-1}}{F_n}}{\dfrac{F_{n-1}}{F_n}}=\frac{F_n-F_{n-1}}{F_{n-1}}=\frac{F_{n-2}}{F_{n-1}}$$

因为有$F_n=F_{n-1}+F_{n-2}$，Fibonacci搜索从$r_0=F_{n-1}/F_n$开始，对$k=1,2,\cdots,n-3$，用$r_k=F_{n-1-k}/F_{n-k}$。

注意$r_{n-3}=F_2/F_3=1/2$，因此这一步无需增加新的点。整个过程总共需要$(n-3)+1=n-2$步。将第k个子区间的长度按因子$r_k=F_{n-1-k}/F_{n-k}$缩减，得到第$(k+1)$个子区间。最后一个子区间的长度为

$$\frac{F_{n-1}F_{n-2}\cdots F_2}{F_nF_{n-1}\cdots F_3}(b_0-a_0)=\frac{F_2}{F_n}(b_0-a_0)=\frac{1}{F_n}(b_0-a_0)=\frac{b_0-a_0}{F_n}$$

如果极小值横坐标的容差为ε，则需要找到最小的n，使得

$$\frac{b_0 - a_0}{F_n} < \varepsilon \text{ 或 } F_n > \frac{b_0 - a_0}{\varepsilon} \qquad (8.1)$$

按要求，由如下公式可找到第k个子区间$[a_k, b_k]$的内点c_k和d_k

$$c_k = a_k + \left(1 - \frac{F_{n-k-1}}{F_{n-k}}\right)(b_k - a_k) \qquad (8.2)$$

$$d_k = a_k + \frac{F_{n-k-1}}{F_{n-k}}(b_k - a_k) \qquad (8.3)$$

其中的n由上面不等式（8.1）求得。

每次迭代需要确定两个新的内点，一个来自前一次的迭代，另一个根据公式（8.2）或（8.3）重新计算。

特别的，当$r_0=F_2/F_3=1/2$时，两个内点将在区间中点重合。为区分它们，引入一个小的区别常数e。当求c_k和d_k的时候系数分别是$r_k=1/2+e$，$(1-r_k)=1/2-e$

例8.5 用斐波那契搜索方法求函数 $f(x)=x^2-\sin(x)$ 在区间 $[0,1]$ 上的极小值。容差为 $\epsilon=10^{-4}$，区别常数为 $e=0.01$。

解：满足

$$F_n > \frac{b_0-a_0}{\epsilon}=\frac{1-0}{10^{-4}}=10000$$

的最小整数为 $F_{21}=10946$。因此 $n=21$。令 $a_0=0$ 且 $b_0=1$，由公式(8)和公式(9)得

$$c_0=0+\left(1-\frac{F_{20}}{F_{21}}\right)(1-0)\approx 0.3819660$$

$$d_0=0+\frac{F_{20}}{F_{21}}(1-0)\approx 0.6180340$$

令 $a_1=a_0, b_1=d_0$ 和 $d_1=c_0$，由于 $f(0.3819660)=-0.2268475$ 和 $f(0.6180340)=-0.1974679$，其中 $f(d_0)\geqslant f(c_0)$。新的包含极小值 f 的子区间是 $[a_1,b_1]=[0,0.6180340]$。利用公式(8)来计算内点 c_1：

$$\begin{aligned} c_1 &= a_1+\left(1-\frac{F_{21-1-1}}{F_{21-1}}\right)(b_1-a_1) \\ &= 0+\left(1-\frac{F_{19}}{F_{20}}\right)(0.6180340-0) \\ &\approx 0.2360680 \end{aligned}$$

计算并比较 $f(c_1)$ 和 $f(d_1)$，以确定新的子区间 $[a_2,b_2]$，并继续进行迭代过程。表8.3给出了部分计算结果。

表 8.3 使用斐波那契搜索法求 $f(x)=x^2-\sin(x)$ 的极小值

k	a_k	c_k	d_k	b_k
0	0.0000000	0.3819660	0.6180340	1.0000000
1	0.0000000	0.2360680	0.3819660	0.6180340
2	0.2360680	0.3819660	0.4721359	0.6180340
3	0.3819660	0.4721359	0.5278641	0.6180340
4	0.3819660	0.4376941	0.4721359	0.5278641
⋮	⋮	⋮	⋮	⋮
16	0.4499360	0.4501188	0.4502102	0.4503928
17	0.4501188	0.4502101	0.4503015	0.4503928
18	0.4501188	0.4502083	0.4502101	0.4503015

在第 17 次迭代中，区间宽度缩小为 $[a_{17},b_{17}]=[0.4501188,0.4503928]$，其中 $c_{17}=0.4502101$，$d_{17}=0.4503105$，而 $f(d_{17})\geqslant f(c_{17})$，因此 $[a_{18},b_{18}]=[0.4501188,0.4503015]$，$d_{18}=0.4502101$。在这一步区间收缩 $r_{18}=1-F_2/F_3=1-1/2=1/2$，用区别常数为 $e=0.01$ 计算 c_{18}：

$$\begin{aligned} c_{18} &= a_{18}+(0.5-0.01)(b_{18}-a_{18}) \\ &= 0.4501188-0.49(0.450315-0.4501188) \\ &\approx 0.4502083 \end{aligned}$$

由于 $f(d_{18})\geqslant f(c_{18})$，最后的子区间为 $[a_{19},b_{19}]=[0.4501188,0.4502101]$，其宽度为 0.0000913。选择该区间的中点为极小值点，因此极小值为 $f(0.4501645)=-0.2324656$。■

8.2.2 分类搜索法

- 黄金分割搜索法与斐波那契搜索法都可用于$f(x)$不可微的情况。
- 搜索法存在的问题：函数在极小值附近可能比较平缓，从而限制了精度，而且速度也较慢
- 对于小的n值，斐波那契搜索法比黄金分割搜索法更为有效；对于大的n值，两者几乎相同

程序实现

黄金分割搜索法：goldenfenge.m

斐波那契搜索法：fibonacci.m

```
function [S,E,G]=goldenfenge(f,a,b,delta,epsilon)
%Input    - f is the object function
%         - a and b are the endpoints of the interval
%   - delta is the tolerance for the abscissas %
% - epsilon is the tolerance for the ordinates
%Output - S=(p,yp) contains the abscissa p and
%   the ordinate yp of the minimum
% - E=(dp,dy) contains the error bounds for p and yp
%  - G is an n x 4 matrix: the kth row contains [ak ck dk
bk];
%   the values of a, c, d, and b at the kth iteration
% f=@(x) sin(x)+exp(x);
% a=0; b=1; delta=0.001; epsilon=0.001;
% [S,E,G]=goldenfenge(f,a,b,delta,epsilon)

r1=(sqrt(5)-1)/2;         r2=r1^2;
h=b-a;
ya=f(a);     yb=f(b);
c=a+r2*h;    d=a+r1*h;
yc=f(c);   yd=f(d);
k=1;
A(k)=a; B(k)=b; C(k)=c; D(k)=d;
while (abs(yb-ya)>epsilon)|(h>delta)
   k=k+1;
   if (yc<yd)
      b=d;   yb=yd;
      d=c;    yd=yc;
h=b-a;
      c=a+r2*h;
      yc=f(c);
   else
      a=c;
      ya=yc;
      c=d;
      yc=yd;
      h=b-a;
      d=a+r1*h;
      yd=f(d);
   end
   A(k)=a; B(k)=b; C(k)=c; D(k)=d;
end

dp=abs(b-a);
dy=abs(yb-ya);
p=a;
yp=ya;

if (yb<ya)
   p=b;
   yp=yb;
end

G=[A' C' D' B'];
S=[p yp];
E=[dp dy];
```

```
function X=fibonacci(f,a,b,tol,e)
%Input    - f, the object function
%         - a, the left endpoint of the interval
%         - b, the right endpoint of the interval
%         - tol, length of uncertainty
%         - e, distinguishability constant
%Output - X, x and y coordinates of minimum
%Note this function calls the m-file fib.m
% f=@(x) sin(x)+exp(x);
% a=0; b=1; tol=0.01; e=2/3;
% X=fibonacci(f,a,b,tol,e)
i=1;
F=1;
while F<=(b-a)/tol
    F=fib(i);
    i=i+1;
end

%Initialize values
n=i-1;
A=zeros(1,n-2);B=zeros(1,n-2);
A(1)=a;     B(1)=b;
c=A(1)+(fib(n-2)/fib(n))*(B(1)-A(1));
d=A(1)+(fib(n-1)/fib(n))*(B(1)-A(1));
k=1;
%Compute Iterates
while k<=n-3
    if f(c)>f(d)
        A(k+1)=c;    B(k+1)=B(k);   c=d;
        d=A(k+1)+(fib(n-k-1)/fib(n-k))*(B(k+1)-A(k+1));
    else
A(k+1)=A(k);
        B(k+1)=d;
        d=c;
        c=A(k+1)+(fib(n-k-2)/fib(n-k))*(B(k+1)-A(k+1));
    end
    k=k+1;
end

%Last iteration using distinguishability constant e
if f(c)>f(d)
    A(n-2)=c;
    B(n-2)=B(n-3);
    c=d;
    d=A(n-2)+(0.5+e)*(B(n-2)-A(n-2));
else
    A(n-2)=A(n-3);
    B(n-2)=d;
    d=c;
    c=A(n-2)+(0.5-e)*(B(n-2)-A(n-2));
end

%Output: Use midpoint of last interval for abscissa
if f(c)>f(d)
    a=c;b=B(n-2);
else
    a=A(n-2);b=d;
end
X=[(a+b)/2 f((a+b)/2)];
```

斐波那契搜索法fibonacci.m

作业 8.1

7. 用黄金分割搜索法，对以下函数求$[a_k, b_k]$，$k=0,1,2$。四舍五入取 5 位有效数字。注：这些函数在给定区间上都是单峰的。

(a) $f(x)=e^x+2x+\frac{x^2}{2}$; $[-2.4, -1.6]$

(b) $f(x)=-\sin(x)-x+\frac{x^2}{2}$; $[0.8, 1.6]$

8. 用斐波那契搜索法，对习题 7 中的函数求$[a_k, b_k]$，$k=0,1,2$。四舍五入取 5 位有效数字。设 F_{10}是满足某个给定容差ϵ的最小斐波那契数。

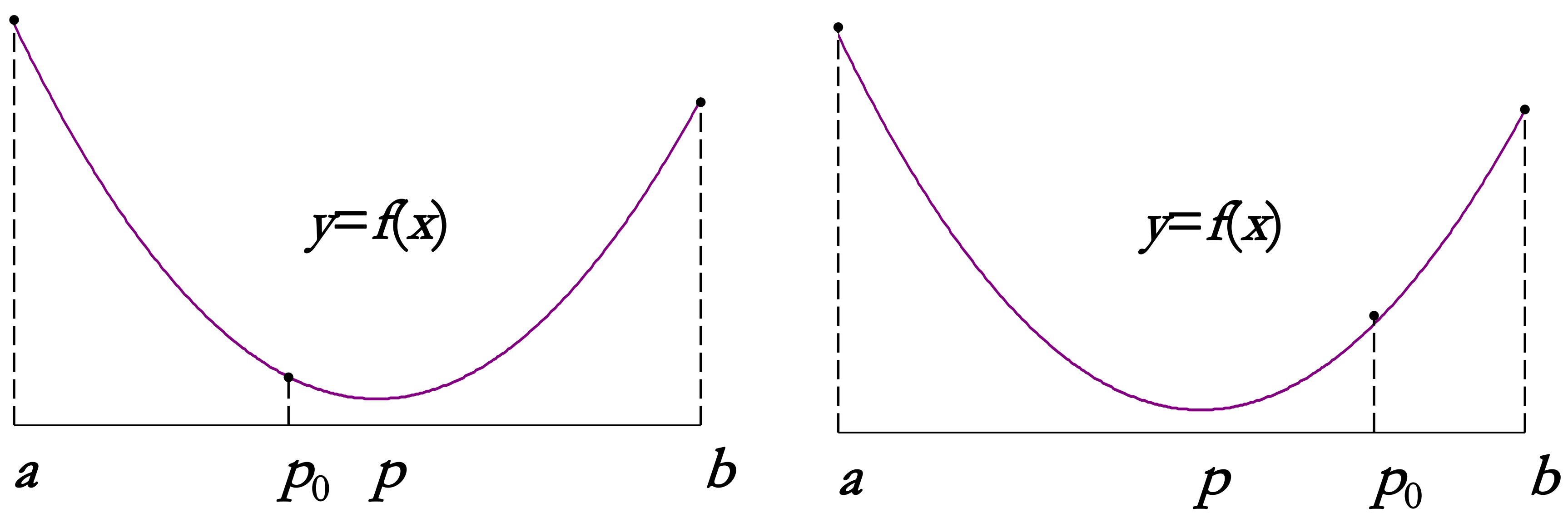
算法与程序

1. 用程序 8.1 求习题 7 中各函数的局部极小值，精确到小数点后 6 位。
2. 用程序 8.2 求习题 7 中各函数的局部极小值，精确到小数点后 6 位。

注： 程序8.1是黄金分割搜索法goldenfenge.m；程序8.2 是斐波那契搜索法fibonacci.m

8.2.3 利用导数求极小值

- 设$f(x)$在区间$[a,b]$上是（下）单峰的，并在$x=p$处有唯一极小值。并设$f'(x)$在(a,b)上所有的点处有定义。令初始点p_0在(a, b)内。若$f'(p_0)<0$，则极小值点p在p_0右侧；若$f'(p_0)>0$，则极小值点p在p_0左侧。



对极小值分类

- 首先求出三个测试值p_0，$p_1=p_0+h$，$p_2=p_0+2h$，使得$f(p_0) > f(p_1)$，$f(p_1) < f(p_2)$成立
- 若$f'(p_0)<0$, 则$p_0<p$，且应该选择步长$h>0$
- 若$f'(p_0)>0$, 则$p_0>p$，且应该选择步长$h<0$
- 容易找到h，使三点p_0, $p_1=p_0+h$, $p_2=p_0+2h$满足要求。如有$a+1<b$，则令$h=1$，否则令$h=1/2$，依此类推。

1. 若满足$f(p_0)>f(p_1)$, $f(p_1)<f(p_2)$则结束
2. 若$f(p_0)>f(p_1)$且$f(p_1)>f(p_2)$，则说明$p_2<p$。则需检测更靠右的点。步长加倍，并重复检测过程
3. 若$f(p_0) \leq f(p_1)$，表明h太大，p_1已经跳过了p。则需检测更靠近p_0的点。步长减半，并重复检测过程

求极小值p的二次逼近方法

- 由p_0, $p_1=p_0+h$, $p_2=p_0+2h$，可用二次插值来求p的近似值p_{min}。基于三点的拉格朗日多项式为

$$Q(x)=\frac{y_0(x-p_1)(x-p_2)}{2h^2}-\frac{y_1(x-p_0)(x-p_2)}{h^2}+\frac{y_2(x-p_0)(x-p_1)}{2h^2}$$

其中$y_i=f(p_i)$, $i=0,1,2$. $Q(x)$的导数为

$$Q'(x)=\frac{y_0(2x-p_1-p_2)}{2h^2}-\frac{y_1(2x-p_0-p_2)}{h^2}+\frac{y_2(2x-p_0-p_1)}{2h^2}$$

以$Q'(p_0+h_{min})$的形式求解$Q'(x)=0$，得

$$0=\frac{y_0(2(p_0+h_{\min})-p_1-p_2)}{2h^2}-\frac{y_1(4(p_0+h_{\min})-2p_0-2p_2)}{2h^2}$$
$$+\frac{y_2(2(p_0+h_{\min})-p_0-p_1)}{2h^2}$$

合并化简可得

$$-h_{\min}(2y_0-4y_1+2y_2)=y_0(2p_0-p_1-p_2)-y_1(4p_0-2p_0-2p_2)+y_2(2p_0-p_0-p_1)$$
$$=y_0(-3h)-y_1(-4h)+y_2(-h)$$

$$-h_{\min}(2y_0-4y_1+2y_2)=y_0(2p_0-p_1-p_2)-y_1(4p_0-2p_0-2p_2)+y_2(2p_0-p_0-p_1)$$
$$=y_0(-3h)-y_1(-4h)+y_2(-h)$$

由上式解得

$$h_{\min}=\frac{h(4y_1-3y_0-y_2)}{4y_1-2y_0-2y_2}$$

值$\boldsymbol{p_{\min}=p_0+h_{\min}}$比$\boldsymbol{p_0}$更逼近$\boldsymbol{p}$，因此可用$\boldsymbol{p_{\min}}$代替$\boldsymbol{p_0}$，并重复上述计算过程，求出新的$\boldsymbol{h}$和新的$\boldsymbol{h_{\min}}$。重复这一迭代过程，直到得到所需的精度。

8.3 多元函数求极值的方法

设函数$f(x_1,x_2,\dots,x_N)$定义在区域

$$R=\left\{(x_1,x_2,\dots,x_N):\sum_{k=1}^{N}(x_k-p_k)^2<r^2\right\}$$

上。如果$f(p_1,p_2,\dots,p_N)\le f(x_1,x_2,\dots,x_N)$对所有的点$(x_1,x_2,\dots,x_N)\in R$都成立，则函数$f(x_1,x_2,\dots,x_N)$在点$(p_1,p_2,\dots,p_N)$处有局部极小值；

如果$f(p_1,p_2,\dots,p_N)\ge f(x_1,x_2,\dots,x_N)$对所有的点$(x_1,x_2,\dots,x_N)\in R$都成立，则函数$f(x_1,x_2,\dots,x_N)$在点$(p_1,p_2,\dots,p_N)$ 处有局部极大值。

二元函数的极小值问题---理论结果

- 二元函数的图形是一个几何表面
- 定理8.5（二阶偏导数测试）设$f(x, y)$及其一阶和二阶偏导数在区域R上连续。设点$(p, q) \in R$是一个临界点，即$f_x(p, q)=0$且$f_y(p, q)=0$。可用高阶偏导数来确定临界点的属性。

I. 若 $f_{xx}(p,q)f_{yy}(p,q)-f_{xy}^2(p,q)>0$ 且 $f_{xx}(p, q)>0$，则 $f(p, q)$是 f 的**局部极小值**。

II. 若$f_{xx}(p,q)f_{yy}(p,q)-f_{xy}^2(p,q)>0$ 且 $f_{xx}(p, q)<0$，则 $f(p, q)$是 f 的**局部极大值**。

III. 若 $f_{xx}(p,q)f_{yy}(p,q)-f_{xy}^2(p,q)<0$，则$f(x, y)$在$(p, q)$没有局部极值。

IV. 若$f_{xx}(p,q)f_{yy}(p,q)-f_{xy}^2(p,q)=0$，则结果不确定。

§ 3.3.1基础知识回顾

函数的方向导数与梯度

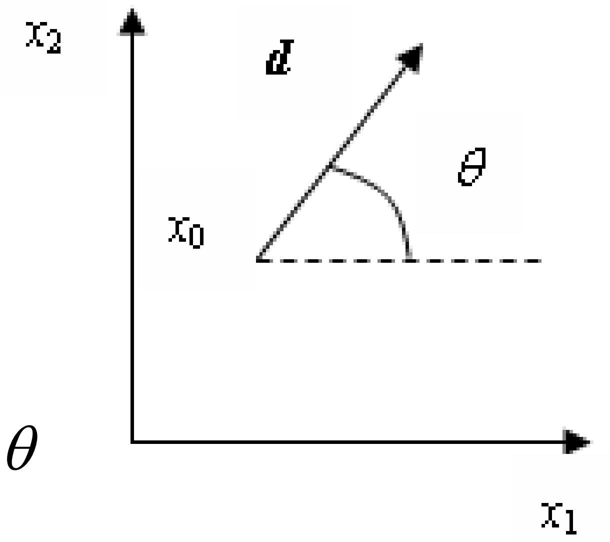
一、梯度

定义8.5：设 $f(x_i)$ $i=1, 2, \cdots n$，则其梯度为：

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \\ \vdots \\ \dfrac{\partial f}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} & \dfrac{\partial f}{\partial x_2} & \cdots & \dfrac{\partial f}{\partial x_n} \end{bmatrix}^T$$

§ 3.3.1基础知识回顾

二、方向导数



定义8.4：函数在某个方向d上的变化率

对于二元函数 $f(x_1,x_2)$（在空间表示一个曲面）有：

$$\frac{\partial f}{\partial \boldsymbol{d}}=\frac{\partial f}{\partial x_1}\frac{\mathrm{d}x_1}{\mathrm{d}\boldsymbol{d}}+\frac{\partial f}{\partial x_2}\frac{\mathrm{d}x_2}{\mathrm{d}\boldsymbol{d}}=\frac{\partial f}{\partial x_1}\cos\theta+\frac{\partial f}{\partial x_2}\sin\theta$$

一般公式(多元函数):

$$\left.\frac{\partial f}{\partial \boldsymbol{d}}\right|_{X_0}=\sum_{i=1}^{n}\left.\frac{\partial f}{\partial x_i}\right|_{X_0}\cos\theta_i$$

θ_i ——***d***与***x*i**轴的夹角

特例： $\frac{\partial f}{\partial x_1}$ $\frac{\partial f}{\partial x_2}$ —— $f(x_1,x_2)$ 分别沿 ***x*1**、***x*2**轴的方向导数

注：教材中，方向用 ***S*** 表示

梯度的几何意义：

$$\frac{\partial f}{\partial \boldsymbol{d}}=\sum_{i=1}^{n}\frac{\partial f}{\partial x_i}\bullet\cos\theta_i=\begin{bmatrix}\frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \cdots & \frac{\partial f}{\partial x_n}\end{bmatrix}\bullet\begin{bmatrix}\cos\theta_1\\ \cos\theta_2\\ \vdots\\ \cos\theta_n\end{bmatrix}$$

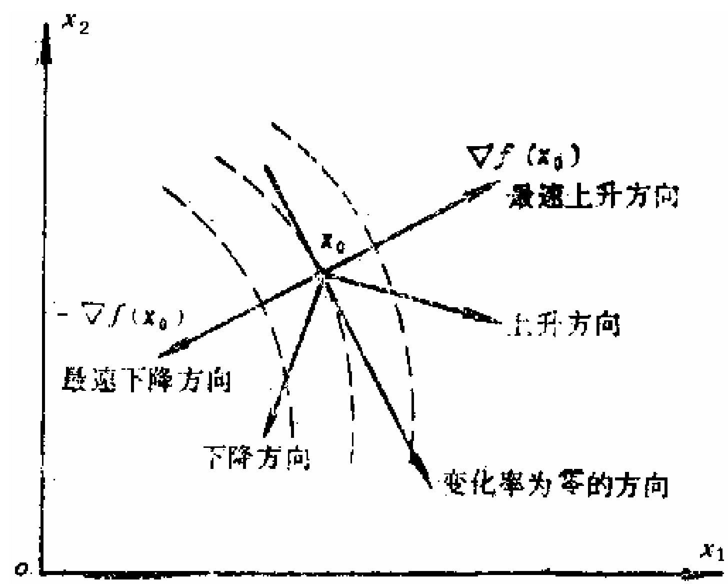
$$=\nabla f(\boldsymbol{x})^T\cdot\boldsymbol{d}=\left\|\nabla f(\boldsymbol{x})\right\|\cos(\nabla f,\boldsymbol{d})$$

当 $\cos(\nabla f,\boldsymbol{d})=1$ 时，即$\boldsymbol{d}$与 $\nabla f(\boldsymbol{x})$ 同向时， $\frac{\partial f}{\partial \boldsymbol{d}}=\nabla f(x)\Rightarrow \max(\frac{\partial f}{\partial \boldsymbol{d}})$

当 $\cos(\nabla f,\boldsymbol{d})=-1$ 时，即$\boldsymbol{d}$与 $\nabla f(\boldsymbol{x})$ 反向时， $\frac{\partial f}{\partial \boldsymbol{d}}=-\nabla f(x)\Rightarrow \min(\frac{\partial f}{\partial \boldsymbol{d}})$

因此，梯度方向是函数值变化最快

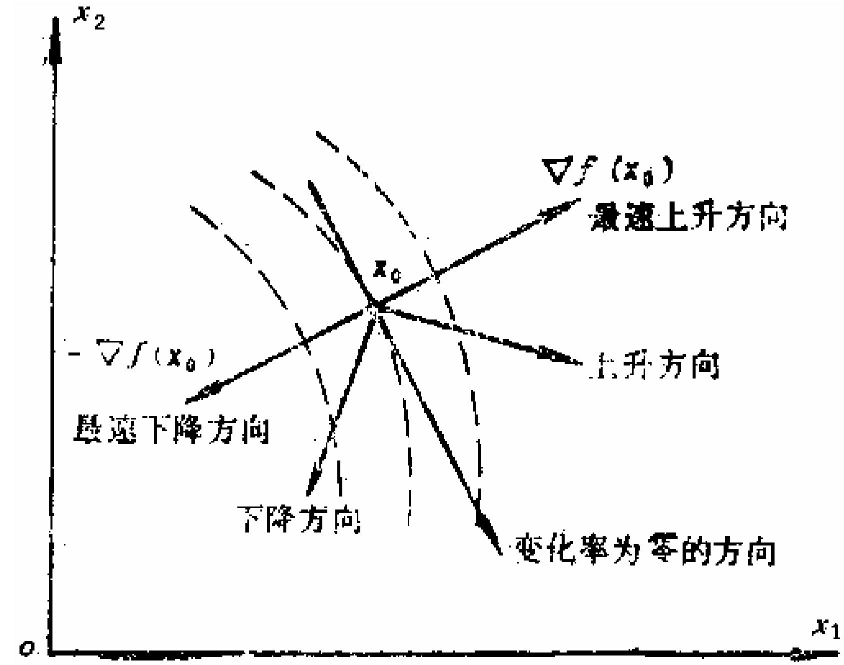
（函数的变化率最大）的方向：



梯度方向与等值线的关系

梯度的性质：

1）梯度是在函数定义域空间里的一个矢量，$\nabla f(x)$ 的方向指向函数最速上升方向；$-\nabla f(x)$ 的方向指向函数最速下降方向。即梯度能定量表明函数在某一点的变化性态。



梯度方向与等值线的关系

2) 函数在某点的梯度只是指出了在该点极小邻域内函数的最速上升方向——并非全局最速上升方向

3) $\nabla f(x)$ 方向是 $f(x)$ 等值线的法线方向

无约束优化 —— 目标函数取极小值

目标函数 $f(\boldsymbol{x}) = f(x_1, x_2, \cdots, x_n)$ 在 $\boldsymbol{x}^*$ 处取极小值的充要条件是：

必要条件：
$$\nabla f(\boldsymbol{x}^*) = \begin{bmatrix} \dfrac{\partial f}{x_1} & \dfrac{\partial f}{x_2} & \cdots & \dfrac{\partial f}{x_n} \end{bmatrix}^T_{x=x^*} = 0$$

充分条件：
$$H(x^*) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n} \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2 \partial x_n} \\ \cdots & & & \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1} & \dfrac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$ 正定

正定的条件：各阶主子式均大于零

主子式:

对于矩阵 $A=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$

一阶主子式 a_{11}

二阶主子式 $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$

三阶主子式 $\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$

正定：各阶主子式均大于零

二元函数的极小值问题---理论结果

- 二元函数的图形是一个几何表面
- 定理8.5（二阶偏导数测试）设$f(x,y)$及其一阶和二阶偏导数在区域R上连续。设点$(p,q)\in R$是一个临界点，即$f_x(p,q)=0$且$f_y(p,q)=0$。可用高阶偏导数来确定临界点的属性。

I. 若 $f_{xx}(p,q)f_{yy}(p,q)-f_{xy}^2(p,q)>0$ 且 $f_{xx}(p,q)>0$，则 $f(p,q)$是 f 的局部极小值。

II. 若$f_{xx}(p,q)f_{yy}(p,q)-f_{xy}^2(p,q)>0$ 且 $f_{xx}(p,q)<0$，则 $f(p,q)$是 f的局部极大值。

III. 若 $f_{xx}(p,q)f_{yy}(p,q)-f_{xy}^2(p,q)<0$，则$f(x,y)$在$(p,q)$没有局部极值。

IV. 若$f_{xx}(p,q)f_{yy}(p,q)-f_{xy}^2(p,q)=0$，则结果不确定。

例8.7 试证明函数 $f(z)=x_1^4-2x_1^2x_2+x_1^2+x_2^2-4x_1+5$

在点(2，4)处具有极小值。

解： $\frac{\partial f}{\partial x_1}=4x_1^3-4x_1x_2+2x_1-4$ $\frac{\partial f}{\partial x_2}=-2x_1^2+2x_2$

$\frac{\partial^2 f}{\partial x_1^2}=12x_1^2-4x_2+2$ $\frac{\partial^2 f}{\partial x_2^2}=2$ 将 *x*1=2，*x*2=4代入

$$\frac{\partial^2 f}{\partial x_1\partial x_2}=-4x_1=\frac{\partial^2 f}{\partial x_2\partial x_1}$$

$\frac{\partial f}{\partial x_1}=\frac{\partial f}{\partial x_2}=0$——存在极值的必要条件

*Hessian*矩阵 $H=\begin{bmatrix}-12x_1^2-4x_2+2 & -4x_1\\ -4x_1 & 2\end{bmatrix}_{(2,4)}=\begin{bmatrix}34 & -8\\ -8 & 2\end{bmatrix}$ 正定

34＞0 34×2-[（-8）×（-8）=4] ＞0 存在充分条件

故函数在点(2，4)处有极小值 *f*(2，4)=1 (极小值为1)

8.3.2 内德-米德方法和鲍威尔方法(选讲)

上一小节介绍的黄金分割搜索法和斐波那契搜索法都不直接使用单变量目标函数的导数，它们只进行函数值的比较。多元函数的直接搜索法也具有这个性质：

- 多变量目标函数$f(x_1,x_2,...,x_N)$的极值直接搜索法对函数的可微性不作显式或隐性的假设
- 对非光滑（不可微）目标函数而言，直接方法特别有用

✓ 内德—米德方法 $\Longrightarrow$ 单纯形法

✓ 鲍威尔方法

单纯形的概念

- 单纯形是指0维中的点，一维中的线段，二维中的三角形，三维中的四面体，n维空间中的有$n+1$个顶点的多面体。例如在三维空间中的四面体，其顶点分别为(0,0,0)，(1,0,0)，(0,1,0)，(0,0,1)。具有单位截距的单纯形的方程是$\sum x_i \le 1$，并且$x_i \ge 0$，$i=1,2,\ldots,n$

二元函数的内德—米德方法（单纯形方法）

- 在二维平面空间中，单纯形就是三角形
- 搜索过程：比较三角形3个顶点处的函数值，

$f(x, y)$值最大的顶点为最差顶点(W)，用一个新的顶点代替最差顶点，形成新的三角形式

- 继续这一过程，生成一系列三角形（它们可能具有不同的形状），函数在其顶点处的值越来越小
- 随着三角形的减小就可以找到极小值点的坐标

算法：

1. 初始三角形 BGW

设要求函数 $f(x,y)$ 的极小值。首先，给定三角形的 3 个顶点 $V_k=(x_k,y_k)$，$k=1,2,3$。在这 3 个点上对函数 $f(x,y)$ 求值：$z_k=f(x_k,y_k)$，$k=1,2,3$。下标编号满足 $z_1\leqslant z_2\leqslant z_3$。用

$$B=(x_1,y_1),\qquad G=(x_2,y_2),\qquad W=(x_3,y_3) \tag{5}$$

来帮助记忆：B 是最佳顶点，G 是次最佳顶点，而 W 是最差顶点。

2. 良边的中点

构造过程使用了连接 $\boldsymbol{B}$ 和 $\boldsymbol{G}$ 的线段的中点。它可由平均坐标得到：

$$\boldsymbol{M}=\frac{\boldsymbol{B}+\boldsymbol{G}}{2}=\left(\frac{x_1+x_2}{2},\frac{y_1+y_2}{2}\right)$$

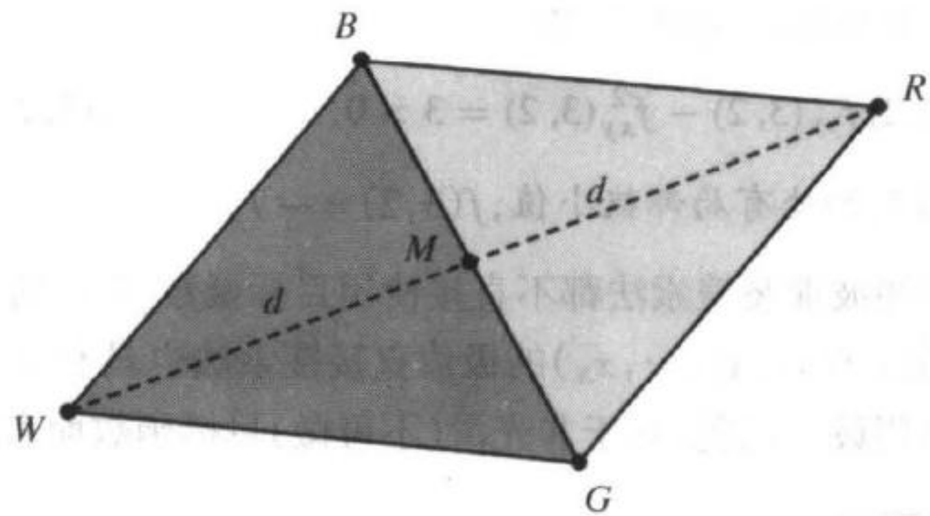


图 8.6　内德－米德方法的三角形△BGW，中点 $\boldsymbol{M}$ 和反射点 $\boldsymbol{R}$

3. 反射点 R

沿着三角形的边由 $\boldsymbol{W}$ 向 $\boldsymbol{B}$ 方向和由 $\boldsymbol{W}$ 向 $\boldsymbol{G}$ 方向，函数的值递减，因此以点 $\boldsymbol{B}$ 和 $\boldsymbol{G}$ 连线为分界线，与 $\boldsymbol{W}$ 相对的点的函数值 $f(x,y)$ 会较小。选择测试点 $\boldsymbol{R}$ 为关于边 $\overline{BG}$ 对三角形进行的“反射”。要确定点 $\boldsymbol{R}$，首先找到边 $\overline{BG}$ 的中点 $\boldsymbol{M}$，然后从点 $\boldsymbol{W}$ 到 $\boldsymbol{M}$ 画线段，称其长度为 d。从 $\boldsymbol{M}$ 点做该线段的延长线，长度为 d，得到点 $\boldsymbol{R}$（见图 8.6）。$\boldsymbol{R}$ 的向量公式为

$$\boldsymbol{R}=\boldsymbol{M}+(\boldsymbol{M}-\boldsymbol{W})=2\boldsymbol{M}-\boldsymbol{W} \tag{7}$$

情形1

4. 开拓点 E

如果 R 处的函数值比 W 处的函数值小，则求解的方向是正确的。可能极小值点的位置只比点 R 略远一点。因此，将过 M 和 R 的线段延长到点 E，构成一个新的开拓三角形 BGE。点 E 沿着连结 M 和 R 的线段的方向延长距离 d（见图 8.7）。如果 E 处的函数值比 R 处的小，则该顶点比 R 好。点 E 的向量公式为

$$E = R + (R - M) = 2R - M$$

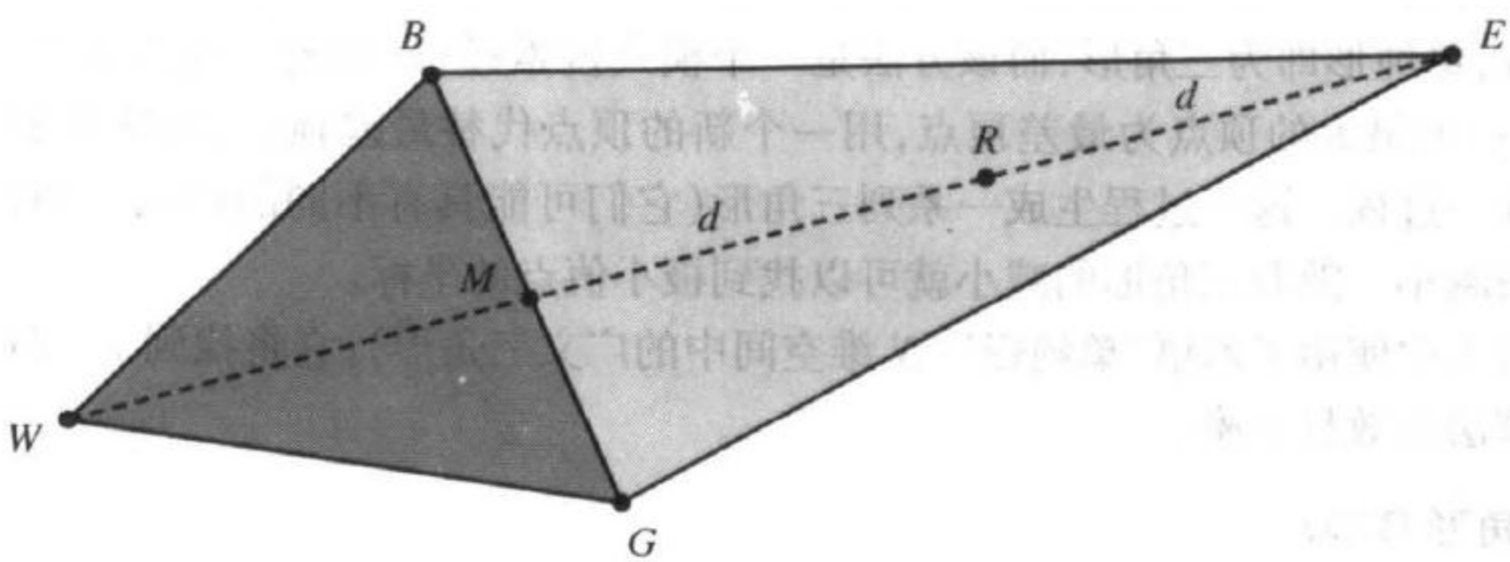


图 8.7　三角形△BGW，点 R 和开拓点 E

情形2

5. 收缩点 C

如果点 $\boldsymbol{R}$ 和 $\boldsymbol{W}$ 的函数值相等，则需要测试另一个点。或许点 $\boldsymbol{M}$ 处的函数值较小，但是为了保证构成三角形，不能用点 $\boldsymbol{M}$ 替代 $\boldsymbol{W}$。分别考虑两个线段 $\overline{\boldsymbol{WM}}$ 和 $\overline{\boldsymbol{MR}}$ 上的点 $\boldsymbol{C}_1$ 和 $\boldsymbol{C}_2$（见图 8.8）。具有较小函数值的点为 $\boldsymbol{C}$，并由此构成三角形 $\boldsymbol{BGC}$。注意，在二维情况下，在点 $\boldsymbol{C}_1$ 和 $\boldsymbol{C}_2$ 之间选择看起来不合适，但在高维情况下这一点非常重要。

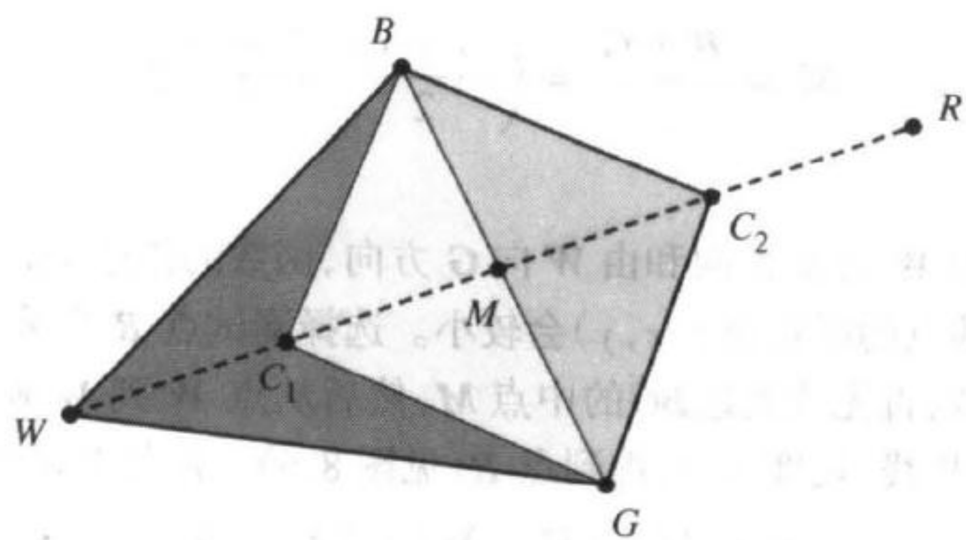


图 8.8　内德－米德方法的收缩点 $\boldsymbol{C}_1$ 和 $\boldsymbol{C}_2$

6. 向 B 方向收缩

如果点 $\boldsymbol{C}$ 处的函数值不小于 $\boldsymbol{W}$ 处的值，则点 $\boldsymbol{G}$ 和 $\boldsymbol{W}$ 必将向 $\boldsymbol{B}$ 的方向收缩(见图 8.9)。点 $\boldsymbol{G}$ 置换为 $\boldsymbol{M}$，点 $\boldsymbol{W}$ 置换为 $\boldsymbol{S}$，后者是连接 $\boldsymbol{B}$ 和 $\boldsymbol{W}$ 的线段中点。

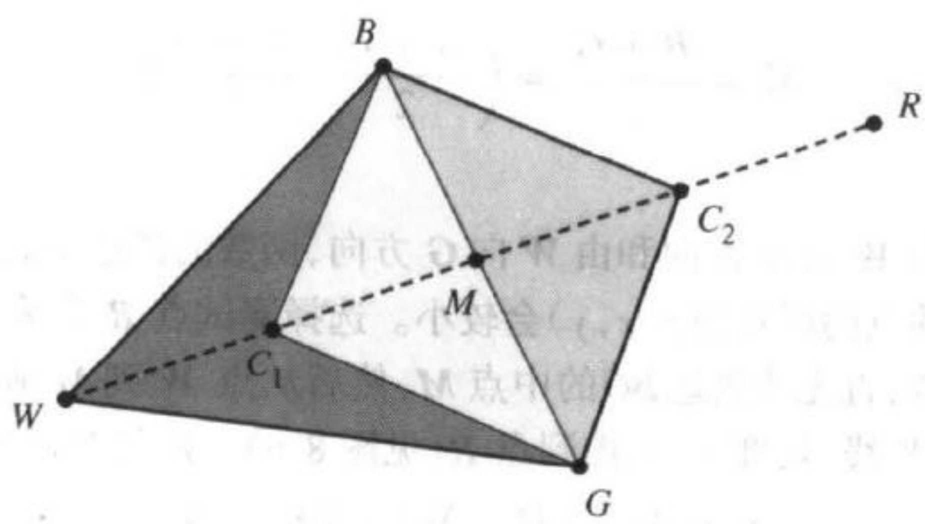


图 8.8　内德－米德方法的收缩点 $\boldsymbol{C}_1$ 和 $\boldsymbol{C}_2$

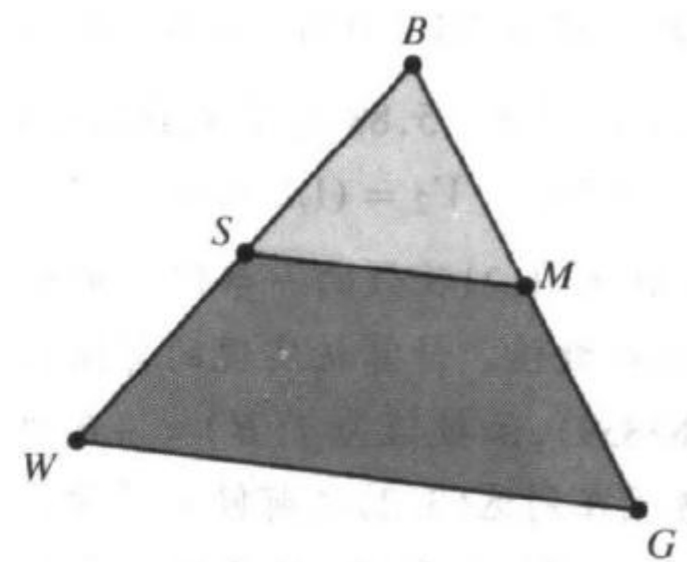


图 8.9　向 $\boldsymbol{B}$ 收缩三角形

7. 每一步的逻辑判断

高效的算法应当只在必要的时候进行函数求值。在每一步中找到一个新的点替代W. 一旦找到这个点，这一步的迭代就完成了。

若$f(R)<f(G)$，则转①（反射点或开拓点）

否则转②（压缩点或收缩点）

①若$f(B)<f(R)$，则以R代替W

否则计算E和$f(E)$

若$f(E)<f(B)$，则以E代替W

否则以R代替W

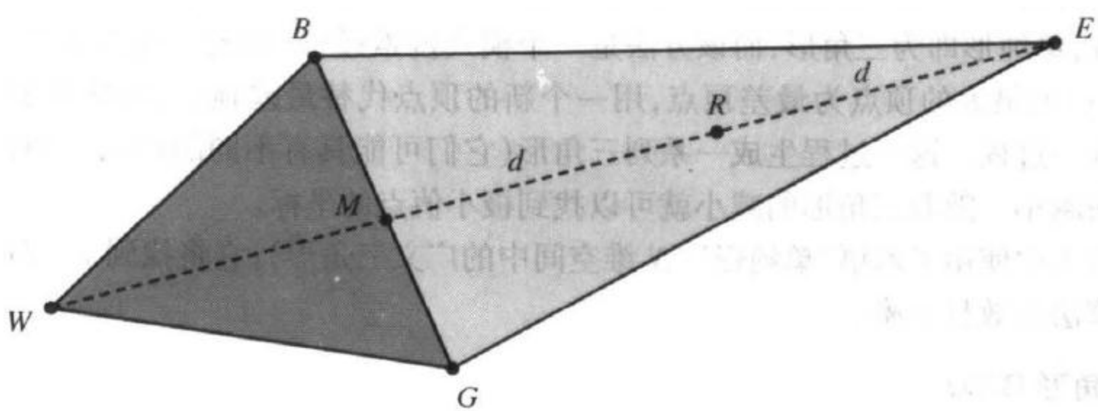


图 8.7　三角形△BGW，点 R 和开拓点 E

②若$f(R)<f(W)$，则以R代替W

否则计算$C_1=(M+R)/2$和$f(C_1)$

计算$C_2=(W+M)/2$和$f(C_2)$

取两者中函数值较小者为C

若$f(C)<f(W)$，则以C代替W

否则计算$S=(W+B)/2$和$f(S)$

以S代替W

以M代替G

内德一米德方法（单纯形方法）的基本思想

- 内德和米德提出了单纯形法，可用于求解多变量函数的局部极小值
- 从可行域中的一个基本可行解出发，判断它是否已是最优解，若不是，寻找下一个基本可行解，并使目标函数得到改进，如此迭代下去，直到找出最优解或判定问题无解为止。
- 从另一个角度说，就是从可行域的某一个极点出发，迭代到另一个极点，并使目标函数的值有所改善，直到找出有无最优解时为止。

例8.9

用内德－米德方法求函数 $f(x,y)=x^2-4x+y^2-y-xy$ 的极小值。从以下 3 个顶点开始：

$$\boldsymbol{V}_1=(0,0),\qquad \boldsymbol{V}_2=(1.2,0.0),\qquad \boldsymbol{V}_3=(0.0,0.8)$$

解：函数 $f(x,y)$ 在顶点处的值为

$$f(0,0)=0.0,\quad f(1.2,0.0)=-3.36,\quad f(0.0,0.8)=-0.16$$

比较它们的值，确定 $\boldsymbol{B}$，$\boldsymbol{G}$ 和 $\boldsymbol{W}$：

$$\boldsymbol{B}=(1.2,0.0),\qquad \boldsymbol{G}=(0.0,0.8),\qquad \boldsymbol{W}=(0,0)$$

顶点 $\boldsymbol{W}=(0,0)$ 将被替代，点 $\boldsymbol{M}$ 和 $\boldsymbol{R}$ 为

$$\boldsymbol{M}=\frac{\boldsymbol{B}+\boldsymbol{G}}{2}=(0.6,0.4),\qquad \boldsymbol{R}=2\boldsymbol{M}-\boldsymbol{W}=(1.2,0.8)$$

函数值 $f(\boldsymbol{R})=f(1.2,0.8)=-4.48$ 小于 $f(\boldsymbol{G})$，因此这是情况(i)。由于 $f(\boldsymbol{R})\leqslant f(\boldsymbol{B})$，移动的方向是正确的，顶点 $\boldsymbol{E}$ 由

$$\boldsymbol{E}=2\boldsymbol{R}-\boldsymbol{M}=2(1.2,0.8)-(0.6,0.4)=(1.8,1.2)$$

构造。函数值 $f(\boldsymbol{E})=f(1.8,1.2)=-5.88$ 小于 $f(\boldsymbol{B})$，新的三角形顶点为

$$\boldsymbol{V}_1=(1.8,1.2),\qquad \boldsymbol{V}_2=(1.2,0.0),\qquad \boldsymbol{V}_3=(0.0,0.8)$$

继续该过程，得到朝着解点(3,2)收敛的一系列三角形(见图 8.10)。表 8.6 给出了迭代中各步的三角形顶点处的函数值。计算机实现的算法执行到第 33 步，得到的最佳顶点为 $\boldsymbol{B}=(2.99996456,1.99983839)$，函数值为 $f(\boldsymbol{B})=-6.99999998$。它是例 8.5 中求得的 $f(3,2)=-7$ 的近似值。迭代在到达(3,2)之前停止的原因是，函数在极小值点附近平缓。经过检查(见表 8.6)，函数值 $f(\boldsymbol{B})$，$f(\boldsymbol{G})$ 和 $f(\boldsymbol{W})$(见表 8.6)相同(这是舍入误差的一个例子)，算法终止。 ■

表 8.6　例 8.6 中各三角形顶点处的函数值

k	最佳点	较好点	最差点
1	$f(1.2, 0.0)=-3.36$	$f(0.0, 0.8)=-0.16$	$f(0.0, 0.0)=0.00$
2	$f(1.8, 1.2)=-5.88$	$f(1.2, 0.0)=-3.36$	$f(0.0, 0.8)=-0.16$
3	$f(1.8, 1.2)=-5.88$	$f(3.0, 0.4)=-4.44$	$f(1.2, 0.0)=-3.36$
4	$f(3.6, 1.6)=-6.24$	$f(1.8, 1.2)=-5.88$	$f(3.0, 0.4)=-4.44$
5	$f(3.6, 1.6)=-6.24$	$f(2.4, 2.4)=-6.24$	$f(1.8, 1.2)=-5.88$
6	$f(2.4, 1.6)=-6.72$	$f(3.6, 1.6)=-6.24$	$f(2.4, 2.4)=-6.24$
7	$f(3.0, 1.8)=-6.96$	$f(2.4, 1.6)=-6.72$	$f(2.4, 2.4)=-6.24$
8	$f(3.0, 1.8)=-6.96$	$f(2.55, 2.05)=-6.7725$	$f(2.4, 1.6)=-6.72$
9	$f(3.0, 1.8)=-6.96$	$f(3.15, 2.25)=-6.9525$	$f(2.55, 2.05)=-6.7725$
10	$f(3.0, 1.8)=-6.96$	$f(2.8125, 2.0375)=-6.95640625$	$f(3.15, 2.25)=-6.9525$

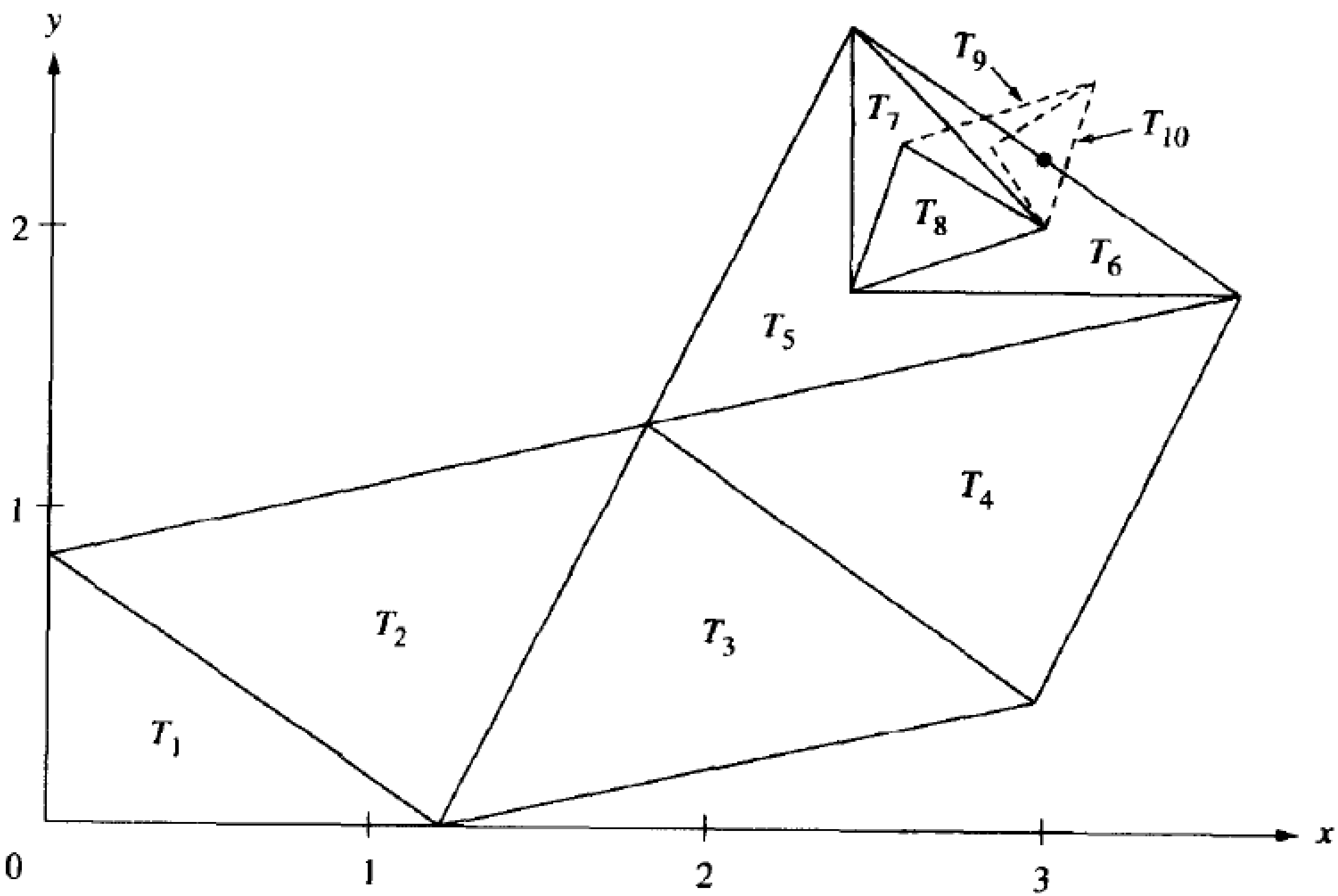


图 8.10　内德 - 米德方法中的三角形系列 $\{T_k\}$，收敛到点(3,2)

实现程序： nelder.m

```
function [V0,y0,dV,dy]=nelder(F,V,min1,max1,epsilon,show)

%Input    - F is input as an M-file function
%         - V is a 3xn matrix containing starting simplex
%         - min1 & max1 are minimum and maximum number of iterations
%         - epsilon is the tolerance
%         - show == 1 displays iterations (P and Q)
%Output  - V0 is the vertex forthe minimum
%         - y0 is the function value F(V0)
%         - dV is the size of the final simplex
%         - dy is the error bound for the minimum
%         - P is a matrix containing the vertex iterations
%         - Q is an array containing the iterations for F(P)

if nargin==5,
  show=0;
end

[mm n]=size(V);

% Order the vertices
for j=1:n+1
  Z=V(j,1:n);
  Y(j)=F(Z);
end
[mm lo]=min(Y);
[mm hi]=max(Y);
li=hi;
ho=lo;

for j=1:n+1
  if(j~=lo&j~=hi&Y(j)<=Y(li))
    li=j;
  end
  if (j~=hi&j~=lo&Y(j)>=Y(ho))
    ho=j;
  end
end

cnt=0;

%  Start of Nelder-Mead algorithm
while (Y(hi)>Y(lo)+epsilon&cnt<max1)|cnt<min1
  S=zeros(1:n);
  for j=1:n+1
    S=S+V(j,1:n);
  end
  M=(S-V(hi,1:n))/n;
  R=2*M-V(hi,1:n);
  yR=F(R);
  if (yR<Y(ho))
    if (Y(li)<yR)
      V(hi,1:n)=R;
      Y(hi)=yR;
    else
      E=2*R-M;
      yE=F(E);
      if (yE<Y(li))
        V(hi,1:n)=E;
        Y(hi)=yE;
      else
        V(hi,1:n)=R;
        Y(hi)=yR;
      end
    end
  else
    if (yR<Y(hi))
      V(hi,1:n)=R;
      Y(hi)=yR;
    end
```

```
C=(V(hi,1:n)+M)/2;
     yC=F(C);
     C2=(M+R)/2;
     yC2=F(C2);
     if (yC2<yC)
        C=C2;
        yC=yC2;
     end
     if (yC<Y(hi))
        V(hi,1:n)=C;    Y(hi)=yC;
     else
        for j=1:n+1
           if (j~=lo)
              V(j,1:n)=(V(j,1:n)+V(lo,1:n))/2;
              Z=V(j,1:n);
              Y(j)=F(Z);
           end
        end
     end
  end
  [mm lo]=min(Y);
  [mm hi]=max(Y);
  li=hi;
  ho=lo;
  for j=1:n+1
     if (j~=lo&j~=hi&Y(j)<=Y(li))
        li=j;
     end
     if (j~=hi&j~=lo&Y(j)>=Y(ho))
        ho=j;
     end
  end
```

```
 cnt=cnt+1;
   P(cnt,:)=V(lo,:);
   Q(cnt)=Y(lo);
end
% End of Nelder-Mead algorithm

%Determine size of simplex
snorm=0;
for j=1:n+1
   s=norm(V(j)-V(lo));
   if(s>=snorm)
      snorm=s;
   end
end

Q=Q';
V0=V(lo,1:n);
y0=Y(lo);
dV=snorm;
dy=abs(Y(hi)-Y(lo));

if (show==1)
    disp([P Q])
end
```

鲍威尔方法

设X_0是函数$z=f(x_1,x_2,\cdots,x_N)$的极小值点位置的初始估计，假设函数的偏导数不可得。一种直观上的很有吸引力的方法是，通过连续地沿着每个标准基向量方向找极小值，来求下一个近似.

$\{E_k=[0\ 0\ \ldots\ 0\ 1_k\ 0\ \ldots\ 0]: k=1,2,\cdots,N\}$**为标准基向量，**

$$\boldsymbol{U}=[\boldsymbol{U}_1^T\ \boldsymbol{U}_2^T\ \cdots \boldsymbol{U}_N^T]=[\boldsymbol{E}_1^T\ \boldsymbol{E}_2^T\ \cdots \boldsymbol{E}_N^T]$$

且i=0

I. 令$P_0=x_i$

II. 对k=1,2,⋯,N，求值γ_k，使得$f(P_{k\text{-}1}+\gamma_k U_k)$极小，并令$P_k=P_{k\text{-}1}+\gamma_k U_k$

III. 令$i=i+1$

IV. 对j=1,2,⋯,N-1，令$U_j=U_{j+1}$。令$U_N=P_N-P_0$，若$||U_N||\leqslant\varepsilon$，则得到解$P_N$，迭代停止

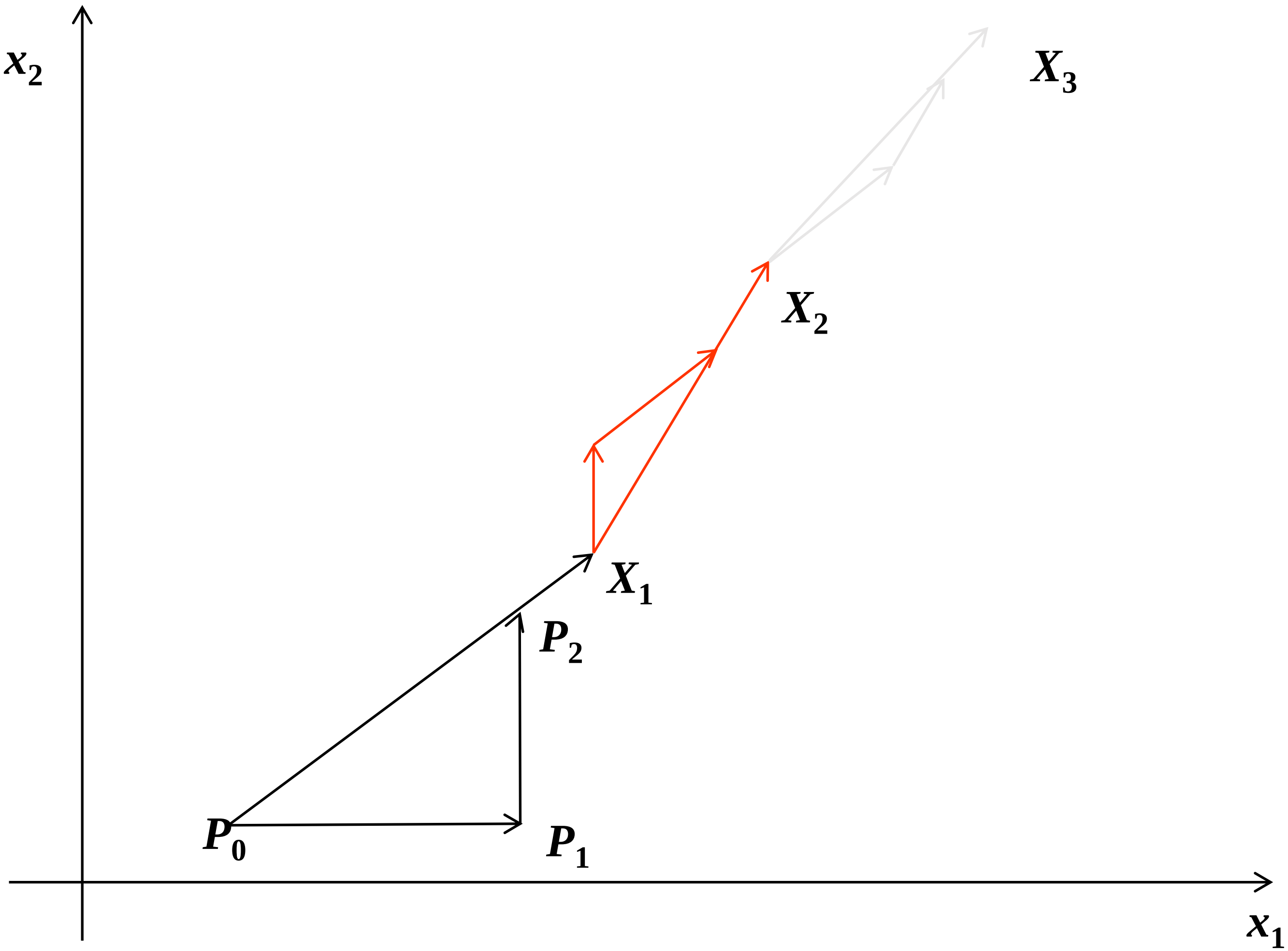
V. 求值γ，使得$f(P_0+\gamma U_N)$极小。令$X_i=P_0+\gamma U_N$

VI. 重复第(I)步到第(V)步

沿着函数f的每个标准基向量是一个单变量函数，这样可用黄金分割搜索法或斐波那契所搜法求解

鲍威尔基本方法示意图

（二维情形）



鲍威尔基本算法要点

- 在每一轮迭代中总有一个始点（第一轮的始点是任取的初始点）和***n***个线性独立的搜索方向。从始点出发顺次沿***n***个方向作一维搜索得一终点，由始点和终点决定了一个新的搜索方向
- 用这个方向替换原来***n***个方向中的一个，于是形成新的搜索方向组。替换的原则是去掉原方向组的第一个方向而将新方向排在原方向的最后。此外规定，从这一轮的搜索终点出发沿新的搜索方向作一维搜索而得到的极小点，作为下一轮迭代的始点。这样就形成算法的循环

上述基本算法仅具有理论意义

鲍威尔方法核心

（1）在某种程度上，向量$\boldsymbol{P}_N$ -$\boldsymbol{P}_0$代表着每次迭代过程移动的平均方向。因此确定点$\boldsymbol{X}_1$为沿向量$\boldsymbol{P}_N$-$\boldsymbol{P}_0$方向的函数f 取得极小值的点。沿着$\boldsymbol{P}_N$ -$\boldsymbol{P}_0$方向的f 是单变量函数，因此可用黄金分割搜索法或斐波那契搜索法

（2）用向量$\boldsymbol{P}_N$ -$\boldsymbol{P}_0$代替下一迭代过程中的某个方向向量，对新的方向向量集开始迭代，并产生点序列$\{\boldsymbol{X}_k\}_{k=0}^{\infty}$

鲍威尔方法的进一步讨论

- 在鲍维尔基本算法中，每一轮迭代都用连结始点和终点所产生出的搜索方向P_N -P_0去替换原来向量组中的第一个向量U_1 ，而不管它的“好坏”。
- 上述步骤中假设了平均方向是继续搜索的良好方向，但实际情况有可能并不是这样
- 随着迭代次数的增加，方向向量集的n个搜索方向有时会趋向于变成线性相关，它将会丢掉一个或多个方向，从而不能组成n维空间，因此可能出现点集 $\{X_k\}_{k=0}^{\infty}$ 并不收敛于局部极小值点的情况

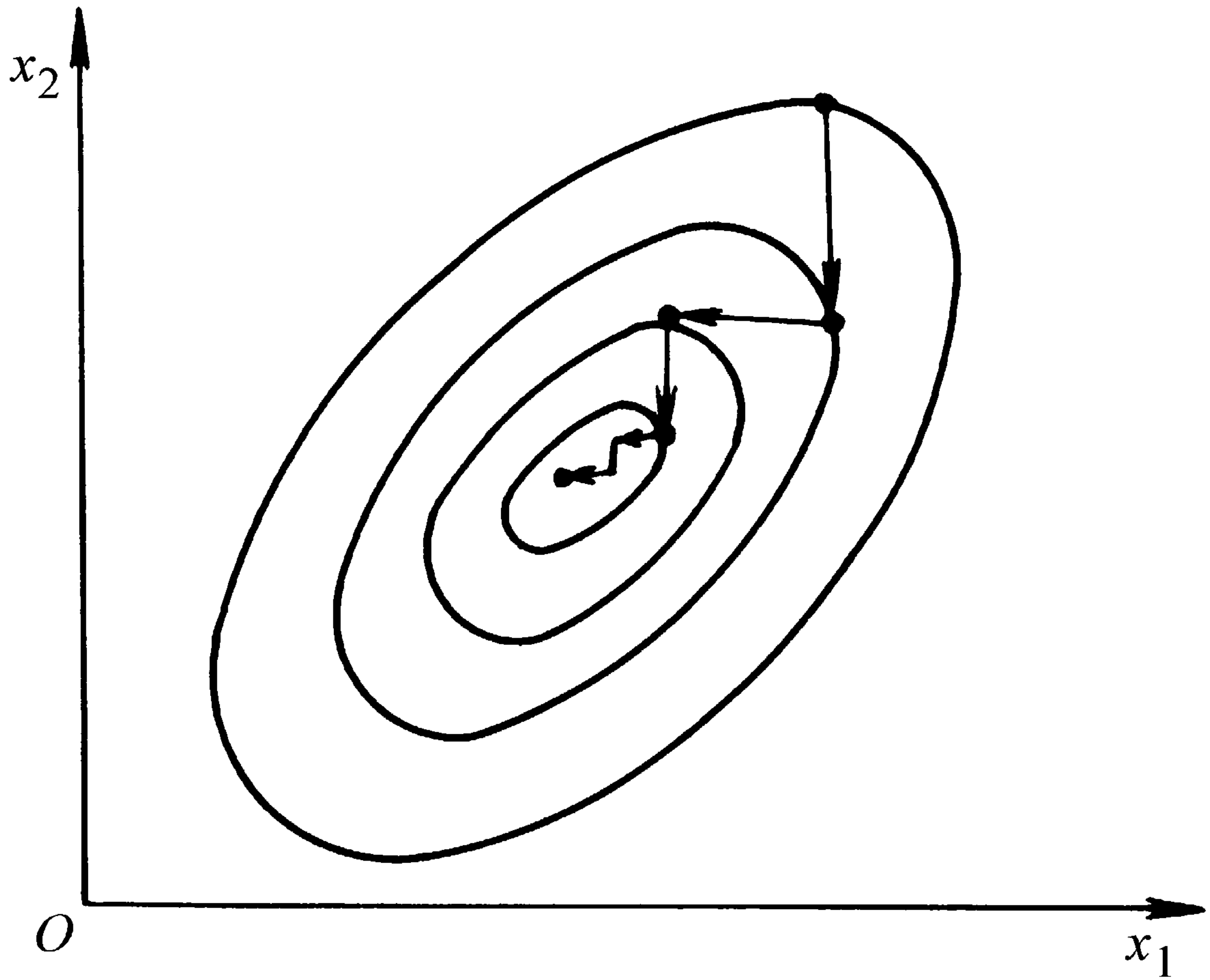
8.3.3 梯度和牛顿方法

- 8.3.2节介绍的内德一米德方法和鲍威尔方法是针对多元函数偏导数不可求的方法，是直接搜索法
- 8.3.3节介绍的梯度方法和牛顿方法是针对函数$f(\boldsymbol{X})$的偏导数可得的求极小值的方法，其中，$\boldsymbol{X}=(x_1, x_2, \cdots, x_N)^T$

(1) 梯度法的定义

- 梯度法又称最速下降法（Steepest descent method）
- 由法国数学家Cauchy于1847年首先提出。在每次迭代中，沿最速下降方向（负梯度方向）进行搜索，每步沿负梯度方向取最优步长，因此这种方法称为最优梯度法。

梯度法搜索过程示意图



梯度方法概要

设$\boldsymbol{P}_k$已知

1. 求梯度向量$\nabla f(\boldsymbol{P}_k)$
2. 计算搜索方向$\boldsymbol{S}_k=-\nabla f(\boldsymbol{P}_k)/||-\nabla f(\boldsymbol{P}_k)||$
3. 在区间$[0,b]$上对$\Phi(\gamma)=f(\boldsymbol{P}_k+\gamma\boldsymbol{S}_k)$进行单参数极小化，$b$为一个较大值。这一过程将产生值$\gamma=h_{min}$，它是$\Phi(\gamma)$的一个局部极小值点。关系式$\Phi(h_{min})=f(\boldsymbol{P}_k+h_{min}\boldsymbol{S}_k)$表明，它是$f(\boldsymbol{X})$沿搜索线$\boldsymbol{X}=\boldsymbol{P}_k+h_{min}\boldsymbol{S}_k$的一个极小值
4. 构造下一个点$\boldsymbol{P}_{k+1}=\boldsymbol{P}_k+h_{min}\boldsymbol{S}_k$
5. 进行极小化过程的终止判断，若函数值满足$|f(\boldsymbol{P}_{k+1})-f(\boldsymbol{P}_k)|<\varepsilon$或两点距离满足$||\boldsymbol{P}_{k+1}-\boldsymbol{P}_k||<\varepsilon$，则迭代终止，否则转第1步

梯度法小结

- 梯度法是从梯度的几何含义自然延伸得到的，所以几何上比较直观
- 梯度法的基本思想是从当前点x_k出发寻找使得目标函数下降最快的方向，即负梯度方向。
- 优点：迭代点列总是收敛的，而且计算过程简单

本章教学要求及重点难点

- 理解数值优化的基本思想、方法和理论
- 熟练掌握分类搜索方法的计算方法，包括黄金分割搜索法和斐波那契搜索法
- 重点：黄金分割搜索法和斐波那契搜索法
- 掌握利用导数求极小值的方法
- 熟练：多元函数求极值的方法，包括：内德-米德方法、鲍威尔方法、梯度和牛顿方法
- 难点：多元函数求极值的方法

其他多变量优化算法

随机森林回归