组合优化与凸优化

第4章 一维搜索(One-dimensional Search)

刘绍辉
计算机科学与技术学院 哈尔滨工业大学
shliu@hit.edu.cn

2022年春季

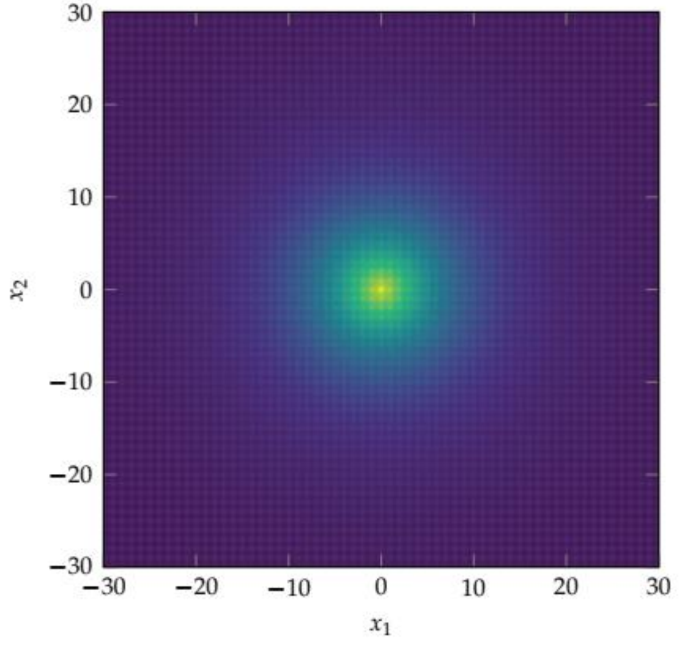
第4章 一维搜索(One-dimensional Search)

- **精确求解**
  - 缩小区间(Narrowing interval)的精确一维搜索
- **近似求解**

几个常见的测试函数

◆Ackley函数

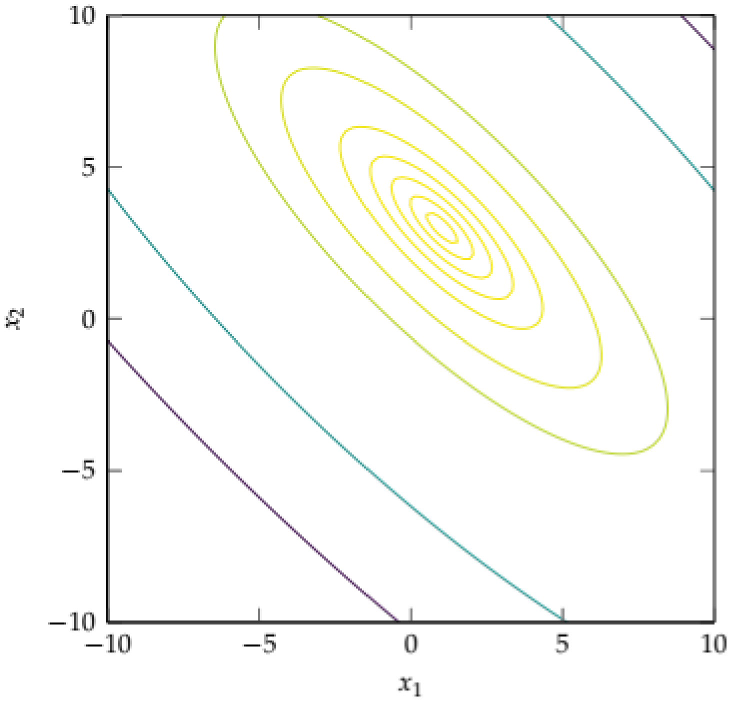
- $f(x) = -ae^{\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right)} - e^{\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)} + a + e$
- 全局极小值在原点
- 如$a = 20, b = 0.2, c = 2\pi$



几个常见的测试函数

◆Booth函数

- ➢$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
- ➢极小点在(1,3),最优值为0



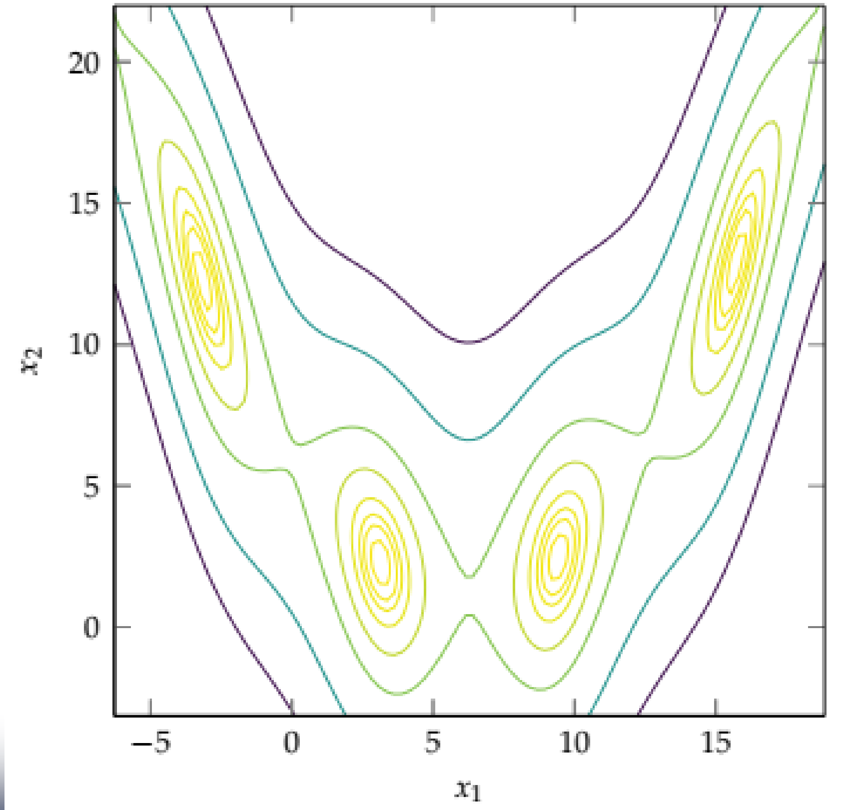
几个常见的测试函数

◆Branin函数

- $f(x) = a\left(x_2 - bx_1^2 + cx_1 - r\right)^2 + s(1-t)cos(x_1) + s$
- 推荐参数：$a = 1, b = \frac{5.1}{4\pi^2}, c = 5\pi\ , r = 6, s = 10, t = \frac{1}{8\pi}$

四个全局极小点
$(-\pi, 12.275), (\pi, 2.275),$
$(3\pi, 2.475), (5\pi, 12.875)$

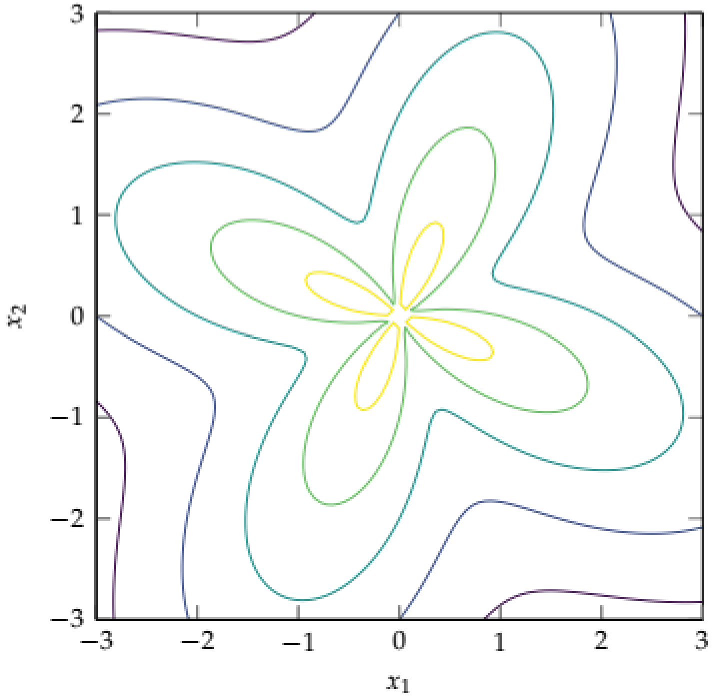
若$x_1 = \pi + 2\pi m, m \in N$,则无全局极小点



几个常见的测试函数

◆Flower函数

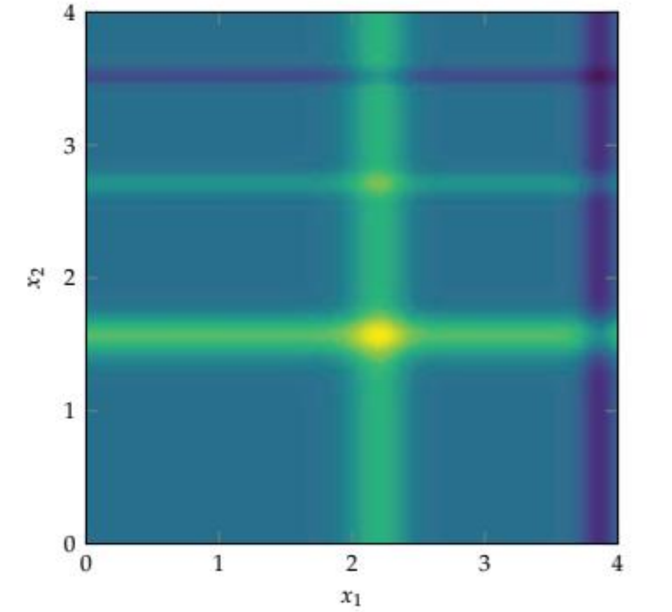
- $f(x) = a||x|| + bsin(c \cdot tan^{-1}(x_2, x_1))$
- 典型参数：$a = 1, b = 1, c = 4$
- 其最小值靠近原点，但不是原点，反正切无定义



几个常见的测试函数

◆Michalewicz函数

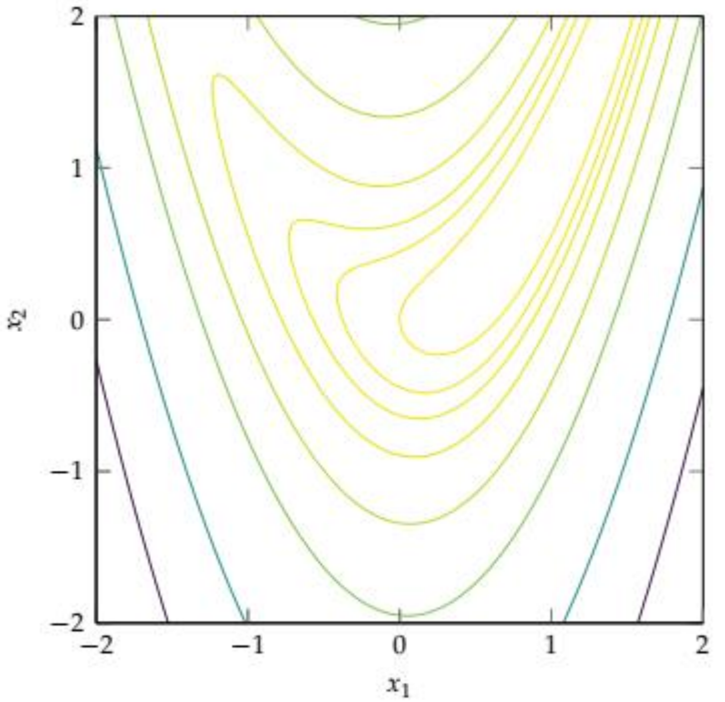
- $f(x) = -\sum_{i=1}^{d} sin(x_i) sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$
- 参数m典型的选择是10，控制陡峭度
- 全局最小值与维数相关
- 二维约在(2.20,1.57),$f(x^*) = -1.8011$



几个常见的测试函数

◆Rosenbrock Banana函数

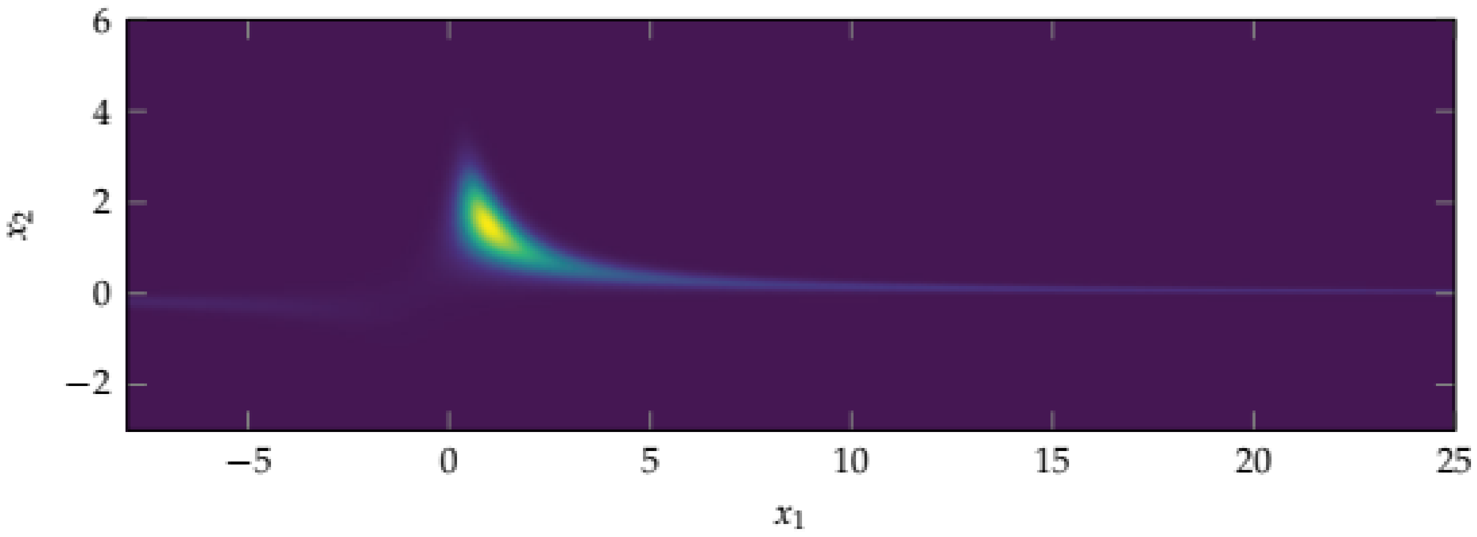
- $f(x) = (a - x_1)^2 - b\left(x_2 - x_1^2\right)^2$
- 全局极小点$\left(a, a^2\right), f(x^*) = 0$
- $a = 1, b = 5$,极小点为(1,1)

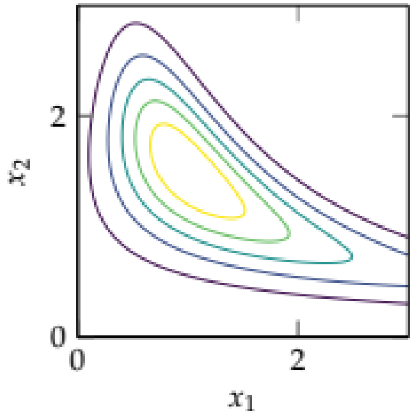


几个常见的测试函数

◆Wheeler函数

- ➢$f(x) = -e^{\left(-(x_1x_2-a)^2-(x_2-a)^2\right)}$
- ➢a设为1.5，在（1，-2/3)处取得极小值-1





几个常见的测试函数

◆Circle函数

- $f(x) = \begin{bmatrix} 1 - r\cos(\theta) \\ 1 - r\sin(\theta) \end{bmatrix}$
- $\theta = x_1, r = \frac{1}{2} + \frac{1}{2}\left(\frac{2x_2}{1+x_2^2}\right)$,Pareto前沿$r = 1, mod(\theta, 2\pi) \in \left[0, \frac{\pi}{2}\right]$或者$r = -1\,, mod(\theta, 2\pi) \in \left[\pi, \frac{3}{2}\pi\right]$
- 多目标测试函数

第4章 一维搜索(One-dimensional Search)

◆一元函数求极小及线性搜索均为一维搜索。常用于求：

$$\begin{cases} \min f\left(x^{(k)}+\lambda d^{(k)}\right)=\varphi(\lambda) \\ \text{s.t.} \quad \lambda \in S \end{cases}$$

S有3种情况（$-\infty$，$+\infty$）或（0，$+\infty$）或$[a,b]$

◆缩小区间的精确一维搜索：考虑问题(P)

$$\begin{cases} \min \varphi(\lambda) \\ s.t. \lambda \in [\alpha,\beta] \end{cases}$$

这里$\varphi(\lambda): R \to R$

1、不确定区间及单峰函数

不确定区间: $[\alpha,\beta]$含$\varphi(\lambda)$的最小点，但不知其位置

◆缩小区间的精确一维搜索（续）

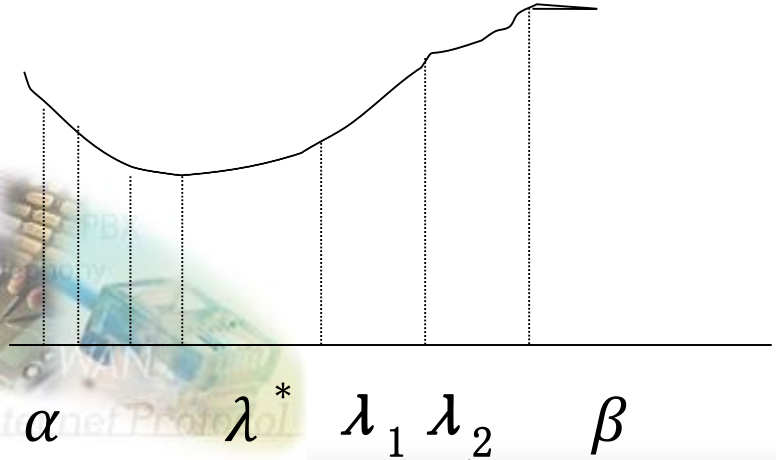
若对任意$\lambda_1, \lambda_2, \alpha \le \lambda_1 < \lambda_2 \le \beta$满足：

1º 若$\lambda_2 \le \lambda^*$，则$\varphi(\lambda_1) > \varphi(\lambda_2)$;

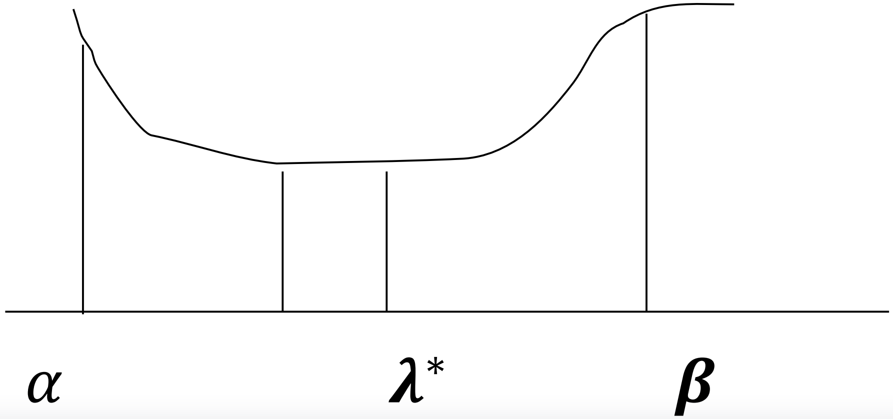
2º 若$\lambda_1 \ge \lambda^*$，则$\varphi(\lambda_1) < \varphi(\lambda_2)$.

则称$\varphi(\lambda)$在$[\alpha, \beta]$上强单谷。

若只有当$\varphi(\lambda_1) \ne \varphi(\lambda^*)$，$\varphi(\lambda_2) \ne \varphi(\lambda^*)$时，上述1º, 2º 式才成立，则称$\varphi(\lambda)$在$[\alpha, \beta]$上单谷。



强单谷



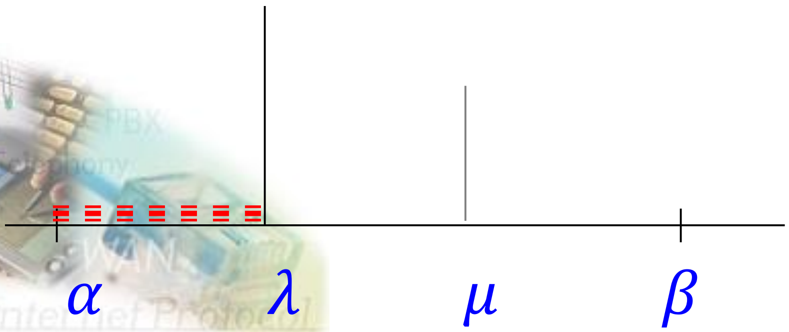
单谷

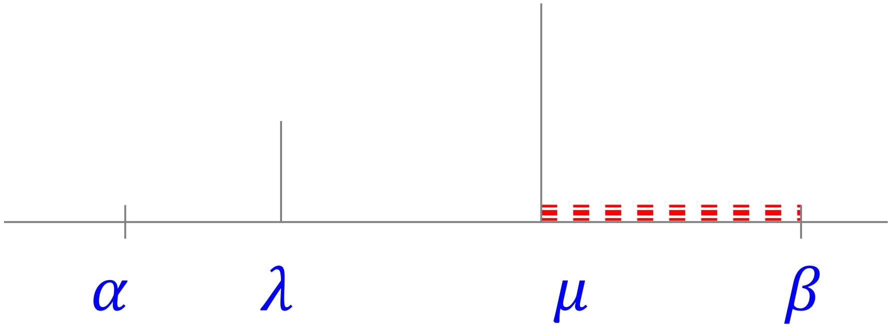
◆缩小区间的精确一维搜索（续）

定理： 设$\Phi: R \to R$ 在$[\alpha, \beta]$上单谷，$\alpha \le \lambda < \mu \le \beta$,则有:

1° 若$\Phi(\lambda) \ge \Phi(\mu)$，则$\Phi(\rho) \ge \Phi(\mu), \forall \rho \in [\alpha, \lambda]$；如左下图

2° 若$\Phi(\lambda) < \Phi(\mu)$，则$\Phi(\rho) \ge \Phi(\lambda), \forall \rho \in [\mu, \beta]$；如右下图





◆缩小区间的精确一维搜索（续）

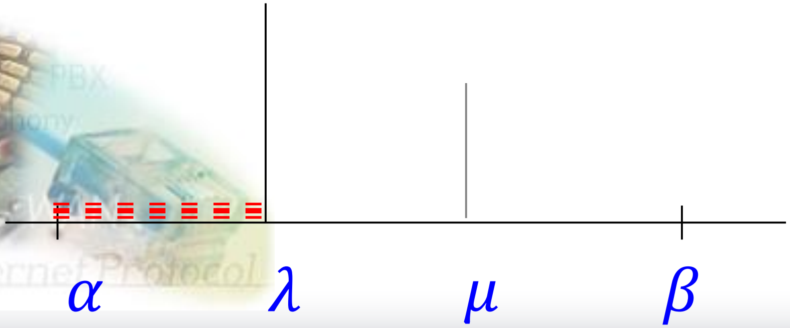
证明：1° 反证：设 $\lambda^* \in [\alpha,\beta]$为最小点,$\gamma \in [\alpha,\lambda]$及$\gamma < \lambda < \lambda^*$，使$\Phi(\gamma) < \Phi(\mu) < \Phi(\lambda)$，

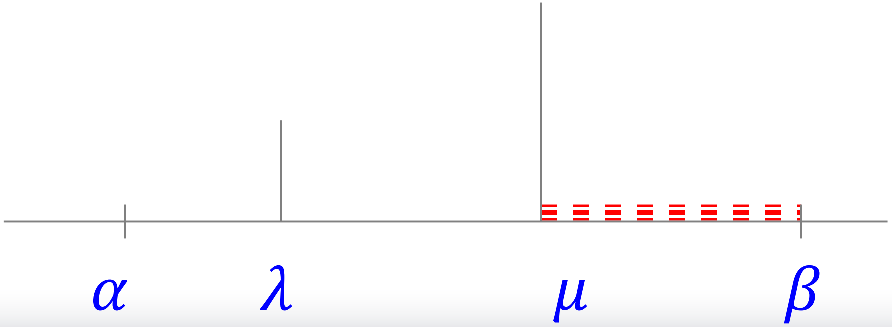
- 若$\lambda^* \in [\lambda,\beta]$，由定义$\Phi(\gamma) > \Phi(\lambda)$，矛盾（假设）；
- 若$\lambda^* \in [\alpha,\lambda)$，由定义及$\mu > \lambda \geq \lambda^*$，$\Phi(\mu) > \Phi(\lambda)$矛盾（条件）；

于是结论成立。

2° 的证明类似（略）。

注：上述定理为缩短区间的算法提供了理论根据





◆缩小区间的精确一维搜索（续）

2、黄金分割法（0.618 法）

通过上述定理，选二点$\lambda < \mu$,比较$\Phi(\lambda)$ 与 $\Phi(\mu)$,可去掉$[\alpha, \lambda]$或者$[\mu, \beta]$.考虑条件：

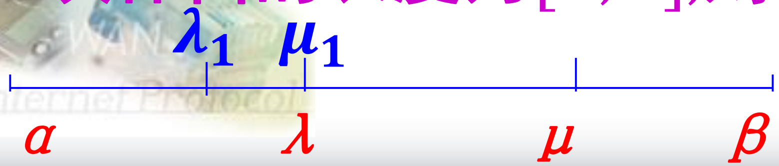
1° 对称： $\lambda - \alpha = \beta - \mu$ ……(1)

使“坏”的情况去掉，区间长度不小于“好”的情况

2° 保持缩减比：t =(保留的区间长度 / 原区间长度) 不变。

使每次保留下来的节点 λ或 μ，在下一次的比较中成为一个相应比例位置的节点

推导缩减比 t：如图设第一次保留$[\alpha, \mu]$ (去掉$[\mu, \beta]$),那么第二次保留的长度为$[\alpha, \lambda]$,则



$$t = \frac{\mu - \alpha}{\beta - \alpha} = \frac{\lambda - \alpha}{\mu - \alpha} \cdots\cdots(2)$$

◆缩小区间的精确一维搜索

2、黄金分割法（0.618 法）（续）

整理②：
$$\begin{cases} \mu = \alpha + t(\beta - \alpha) \\ \lambda = \alpha + t(\mu - \alpha) \end{cases}$$

结合①式：$t^2 + t - 1 = 0$

$$t = \frac{-1 \pm \sqrt{5}}{2}\text{(舍去负值)}$$

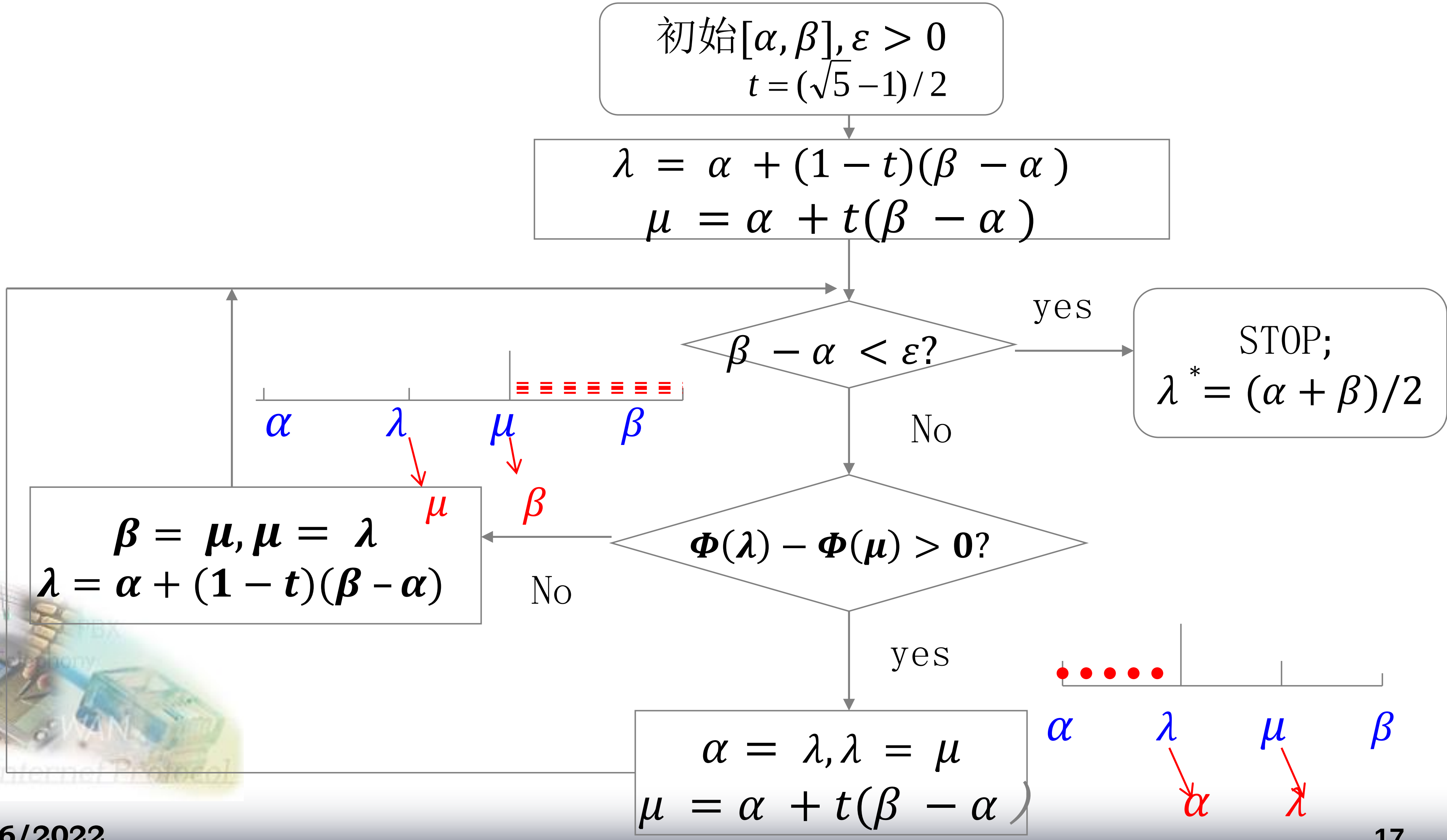
故 $t \approx 0.618$

注意 上式有 $t^2 = 1 - t$，故有

$$\begin{cases} \mu = \alpha + t(\beta - \alpha) \\ \lambda = \alpha + (1 - t)(\beta - \alpha) \end{cases}$$

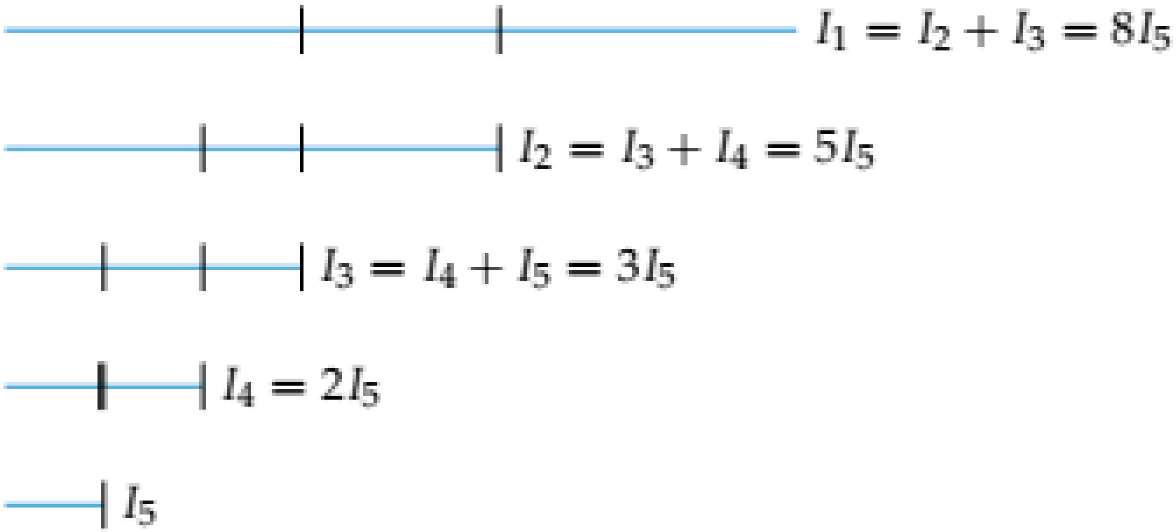
算法框图如下

◆缩小区间的精确一维搜索 之黄金分割法（0.618 法）



第4章 一维搜索(One-dimensional Search)

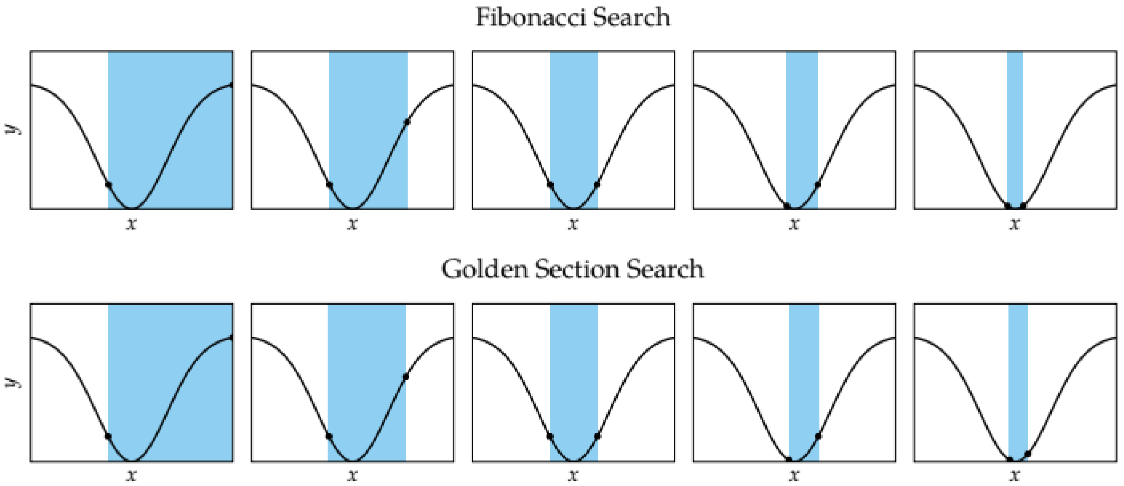
- **与黄金分割法类似的方法还有Fibonacci法，区别在于搜索区间长度的缩短率不是黄金分割数，而是Fibonacci数，**
  - $F_1 = F_2 = 1, F_n = F_{n-1} + F_{n-1}, n \geq 2$
  - 假设最后满足要求的区间为I_n,则倒数第二个区间宽度为$F_2 I_n$,倒数第3个区间宽度为$F_3 I_n \cdots$



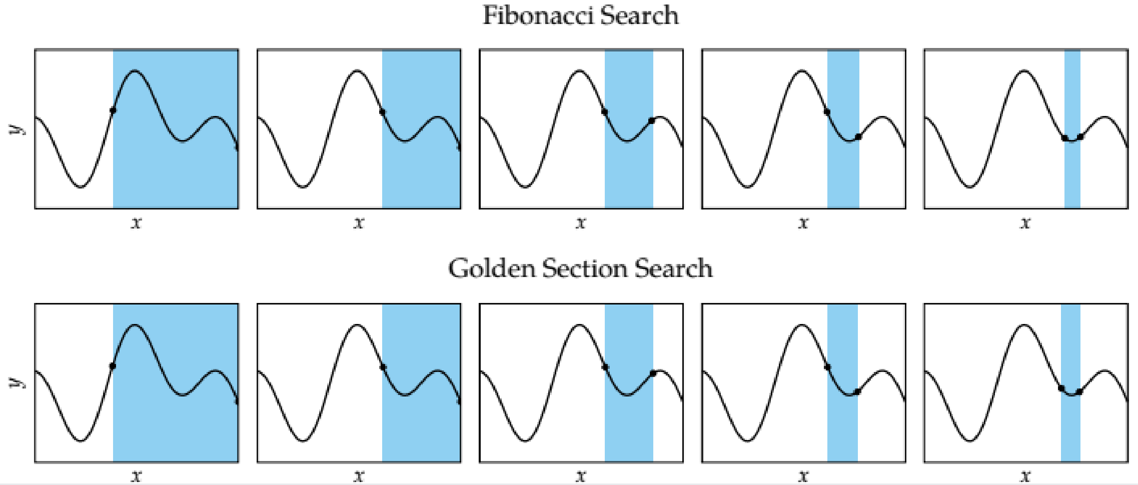
- **缩短率$b_{k+1} - a_{k+1} = \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k)$,要求$n$次后，$b_n - a_n \leq \delta$,实际上，当$n \to \infty$, $\lim \frac{F_n}{F_{n+1}} = (\sqrt{5}$-1)/2；$b_n - a_n = \frac{1}{F_n}(b_1 - a_1)$,选择合适的$n$,使得精度符合要求。**

◆比较Fibonacci和黄金分割搜索方法

➢单模函数



➢非单模函数



◆3、中点法(二分法)

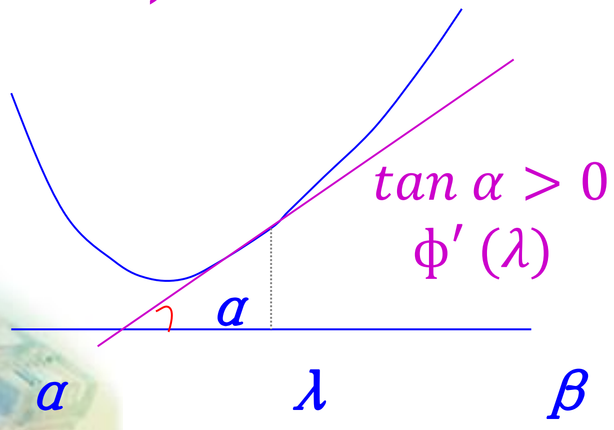
设$\phi(\lambda)$在$[\alpha, \beta]$上可微，且当导数为零时是解。取$\lambda=(\alpha+\beta)/2$，那么

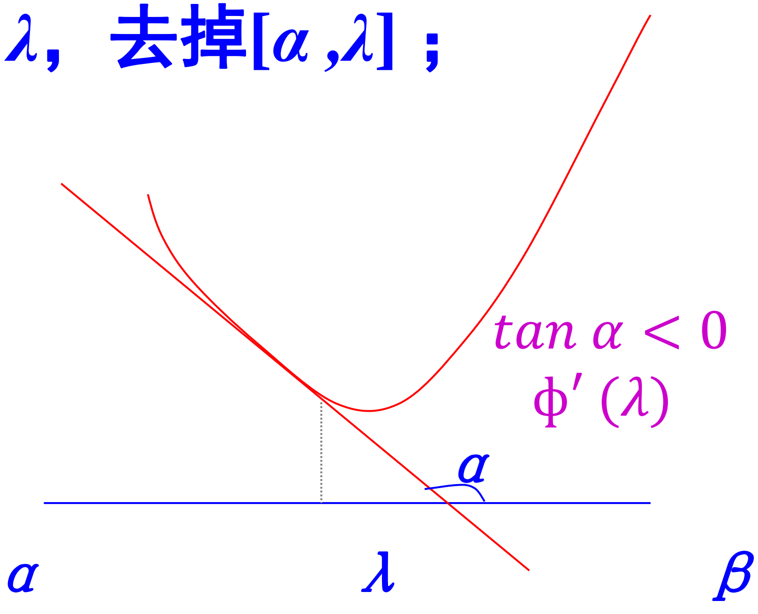
$\phi'(\lambda)=0$ 时，λ为最小点，$\lambda=\lambda^*$　；

$\phi'(\lambda)>0$ 时，λ在上升段，$\lambda^*<\lambda$，去掉$[\lambda,\beta]$；

$\phi'(\lambda)<0$ 时，λ在下降段，$\lambda^*>\lambda$，去掉$[\alpha,\lambda]$；

(算法框图略)





第4章 一维搜索(One-dimensional Search)

4、进退法求初始不确定区间

找三点使两端点的函数值大于中间点的函数值。

思路： 任取λ_0，步长$\delta>0$，取$\lambda_1=\lambda_0+\delta$，

1° 若$\Phi(\lambda_0)<\Phi(\lambda_1)$, 令$\delta=2\delta$（步长加倍），$\lambda_2=\lambda_0-\delta$，

- 若$\Phi(\lambda_2)<\Phi(\lambda_0)$,则令$\lambda_1=\lambda_0$，$\lambda_0=\lambda_2$,重复1°
- 若$\Phi(\lambda_2)>\Phi(\lambda_0)$,则停，$\alpha=\lambda_2$，$\beta=\lambda_1$ (图1)

2° 若$\Phi(\lambda_0)>\Phi(\lambda_1)$令$\delta=2\delta,\lambda_2=\lambda_1+\delta$，

- 若$\Phi(\lambda_2)<\Phi(\lambda_1)$,则令$\lambda_0=\lambda_1$，$\lambda_1=\lambda_2$,重复2°
- 若$\Phi(\lambda_2)>\Phi(\lambda_1)$,则停，$\alpha=\lambda_0$，$\beta=\lambda_2$ (图2)

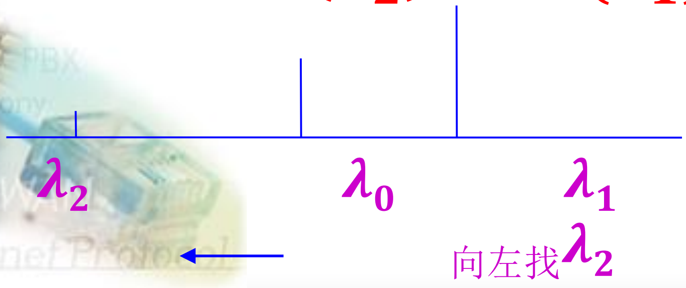


图1

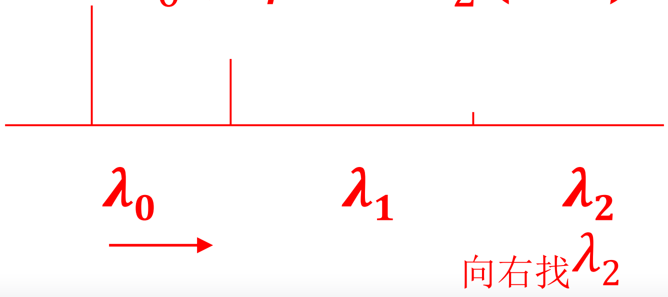


图2

4、进退法求初始不确定区间（续）

（自己画算法框图）

- 注意
  - δ 选择要适当。（太大含多个单峰区间，太小迭代次数增加）；
  - $\Phi(\lambda)$单调时无结果，(加迭代次数限制);
  - 可与中点法结合寻找单调区间（思考）

◆牛顿法（Newton）和插值法

➢插值法利用插值函数逼近所需求解的目标函数，把插值函数的极小点作为迭代点。常见的有三点二次插值，两点二次插值和三次插值多项式

1、Newton法：

对Φ在λ_k点展开：

$$\Phi(\lambda) = \Phi(\lambda_k) + \Phi'(\lambda_k)(\lambda-\lambda_k) + (1/2)\Phi''(\lambda_k)(\lambda-\lambda_k)^2 + O(\lambda-\lambda_k)^2$$

取二次式(略去高阶项):

$$q_{k(\lambda)} = \Phi(\lambda_k) + \Phi'(\lambda_k)(\lambda-\lambda_k) + \frac{1}{2}\Phi''(\lambda_k)(\lambda-\lambda_k)^2$$

◆牛顿法（Newton）和插值法

1、Newton法： （续）

用$q_k(\lambda)$作为$\Phi(\lambda)$的近似，当$\Phi''(\lambda_k) > 0$时，其驻点为极小点：

$$q'_k(\lambda) = \Phi'(\lambda_k) + \Phi''(\lambda_k)(\lambda - \lambda_k) = 0$$

得 $\lambda_{k+1} = \lambda_k - \Phi'(\lambda_k)/\Phi''(\lambda_k)$

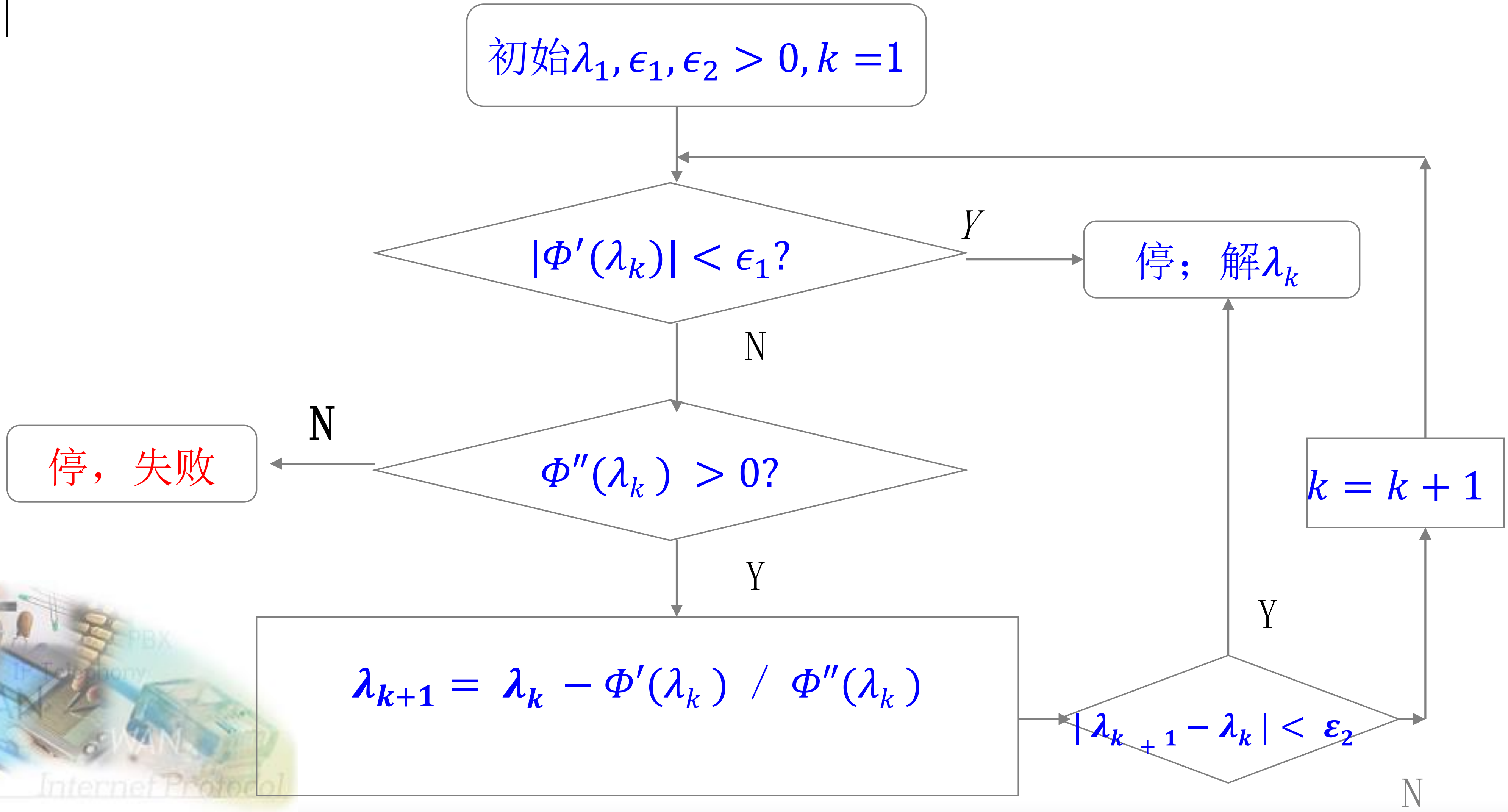
取λ_{k+1}为新的迭代点。以上过程即Newton法。

特点：*二阶、局部收敛*。

算法框图如下

第4章 一维搜索-Newton和插值法

◆Newton法算法框图：



二、牛顿法（Newton）和插值法

1、Newton法：（续）

例. 求 $\min \Phi(\lambda) = \int_0^{\lambda} \arctan t dt$

解： $\Phi'(\lambda) = arctan\lambda$， $\Phi''(\lambda) = 1/(1+\lambda^2)$

迭代公式：$\lambda_{k+1} = \lambda_k - (1+\lambda^2) arctan\lambda_k$

取$\lambda_1 = 1$，计算结果：

k	λ_k	$\Phi'(\lambda_k)$	$1/\Phi''(\lambda_k)$
1	1	0.7854	2
2	-0.5708	-0.5187	1.3258
3	0.1169	-0.1164	1.0137
4	-0.001095	-0.001095	

$\lambda_4 \approx \lambda^* = 0$

取$\lambda_1=2$，计算结果如下：

二、牛顿法（Newton）和插值法

1、Newton法：（续）

k	λ_k	$\Phi'(\lambda_k)$	$1/\Phi''(\lambda_k)$
1	2	1.1071	5
2	-3.5357	-1.2952	13.50
3	13.95	不收敛。	

- 牛顿法的基本思想是在迭代点$x^{(k)}$附近用二次函数$q^{(k)}(s)=f\left(x^{(k)}\right)+g_k^T s+\frac{1}{2}s^T G_k s$来逼近$f(x)$,并以$q^{(k)}(s)$的极小点$s^{(k)}$来修正$x^{(k)}$,得到$x^{(k+1)}=x^{(k)}+s^{(k)}$

二、牛顿法（Newton）和插值法

1、Newton法：（续）

k	λ_k	$\Phi'(\lambda_k)$	$1/\Phi''(\lambda_k)$
1	2	1.1071	5
2	-3.5357	-1.2952	13.50
3	13.95	不收敛	

2、插值法：

用$\phi(\lambda)$在2 或3 个点的函数值或导数值，构造2 次或3次多项式作为$\phi(\lambda)$的近似值，以这多项式的极小点为新的迭代点

3点2次，2点2次，4点3次，3点3次，2点3次等

以3点2次为例：

取λ_1，λ_2，λ_3，求出$\phi(\lambda_1)$，$\phi(\lambda_2)$，$\phi(\lambda_3)$

◆牛顿法（Newton）和插值法

2、插值法：（续）

设二次插值多项式：$a\lambda^2+b\lambda+c=\phi(\lambda)$

$$
\begin{cases}
a\lambda_1^2+b\lambda_1+c=\phi(\lambda_1)\\
a\lambda_2^2+b\lambda_2+c=\phi(\lambda_2)\\
a\lambda_3^2+b\lambda_3+c=\phi(\lambda_3)
\end{cases}
$$

解得a,b

$$a=-\frac{(\lambda_1-\lambda_2)\Phi(\lambda_3)+(\lambda_2-\lambda_3)\Phi(\lambda_1)+(\lambda_3-\lambda_1)\Phi(\lambda_2)}{(\lambda_1-\lambda_2)(\lambda_2-\lambda_3)(\lambda_3-\lambda_1)}$$

$$b=\frac{(\lambda_1^2-\lambda_2^2)\Phi(\lambda_3)+(\lambda_2^2-\lambda_3^2)\Phi(\lambda_1)+(\lambda_3^2-\lambda_1^2)\Phi(\lambda_2)}{(\lambda_1-\lambda_2)(\lambda_2-\lambda_3)(\lambda_3-\lambda_1)}$$

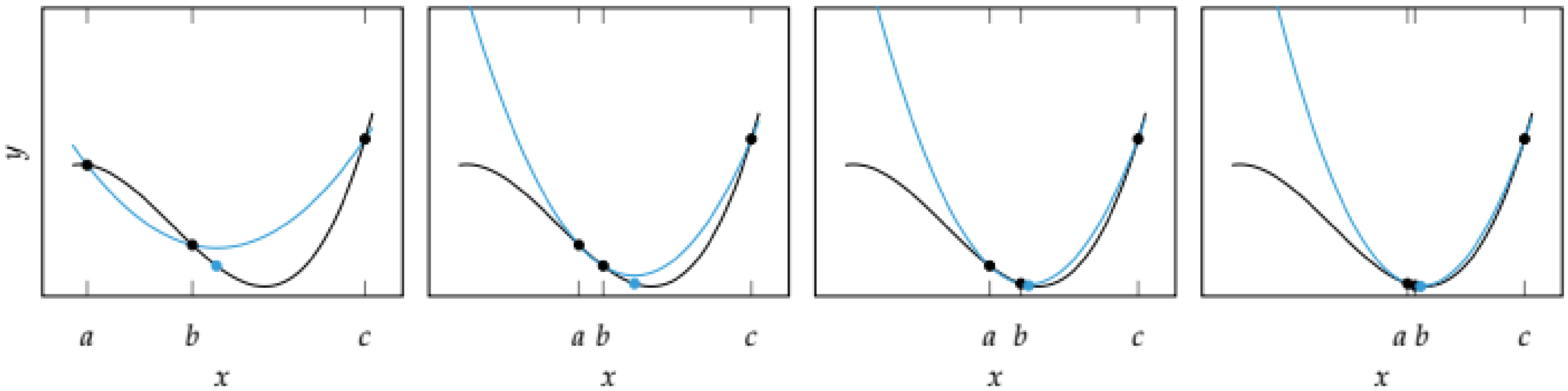
$$\bar{\lambda}=-\frac{b}{2a}$$

◆ 2、插值法：（续）

- 上述及得4个已知点，设为$\lambda_0, \bar{\lambda}, \lambda_1, \lambda_2$，然后从中舍去一个，保证最小点在区间内
- 怎么保证？初始三点如何选取？

◆ 两点两次

- 两点λ_0, λ_1函数值及一点的导数值，$\psi(\lambda) = a\lambda^2 + b\lambda + c$,从而可得



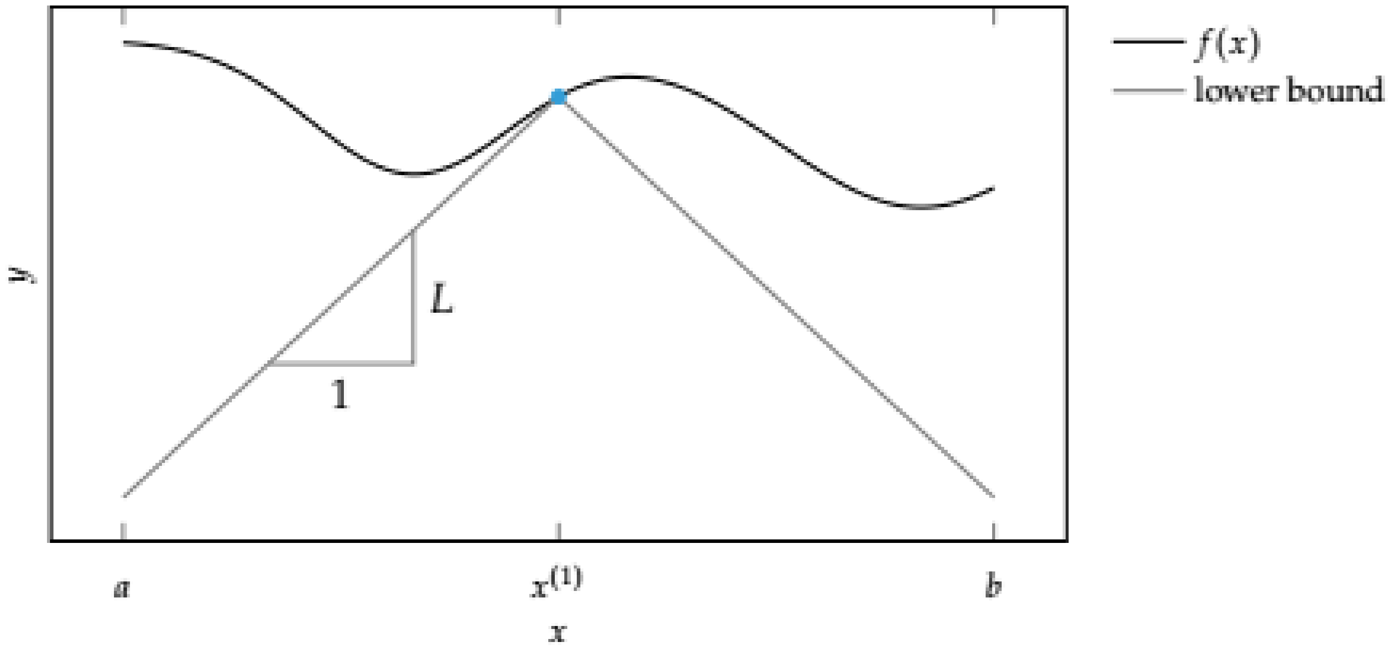
复这个过程
- 特点：简单，不必预先确定上下界，但函数单调时得不到结果，因此需要规定迭代次数限制

- 三次插值：四个参数
  - 因此需要四个条件：可以是四个点，3点加一点的导数值，2点加两点导数值
- 实用中，特别对单变量问题求最小点时，利用上面的方法相互配合，会收到较好的结果
- 总体来说，精确一维搜索需要花费很大的工作量，特别是当迭代点远离问题的解的时候，精确求解一个一维子问题通常不是十分有效
- 实际上，牛顿法和拟牛顿法的收敛速度并不依赖于精确一维搜索，因此只要保证目标函数值每一步都有满意的下降，这样可以大大节省工作量
- 上述方法一般都收敛到局部极小点，而无法收敛到全局极小点

第4章 一维搜索-Shubert-Piyavskii方法

- **能收敛到全局极小点**
- **如果函数是Lipschitz连续：即导数的幅度存在上界**
  - $|f(x)-f(y)| \le l|x-y|, \forall x, y \in [a, b]$,可推广到向量和范数
  - 给定点$\left(x_0, f(x_0)\right)$, 若$x > x_0$, 直线$f(x_0)-l(x-x_0)$； $x < x_0, f(x_0)+l(x-x_0)$形成函数f的下界
- **给定李普希兹常数l,算法从中点$x^{(1)}=\frac{a+b}{2}$,通过该点斜率为$\pm l$的直线构造f的锯齿样下界。如果常数l有效，这些直线总是在函数f的下面**

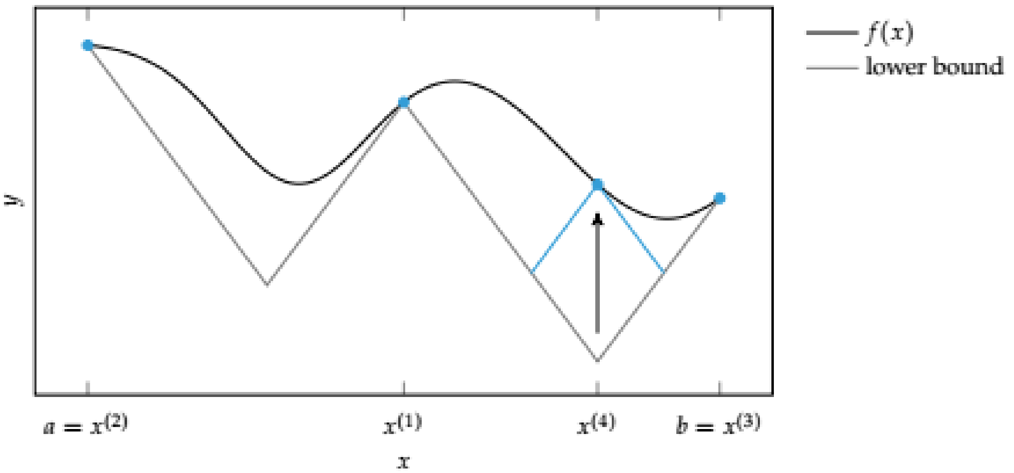
一步步构建函数的下界



◆Shubert-Piyavskii

- 算法在最小的锯齿点（$x^{(n)}, y^{(n)}$）与该点的函数值$f\left(x^{(n)}\right)$之差小于某个小常数的时候停止
- $y^{(n)} - f\left(x^{(n)}\right) < \epsilon$

不断迭代找到锯齿线的最小值点，并计算改点的函数值来更新锯齿

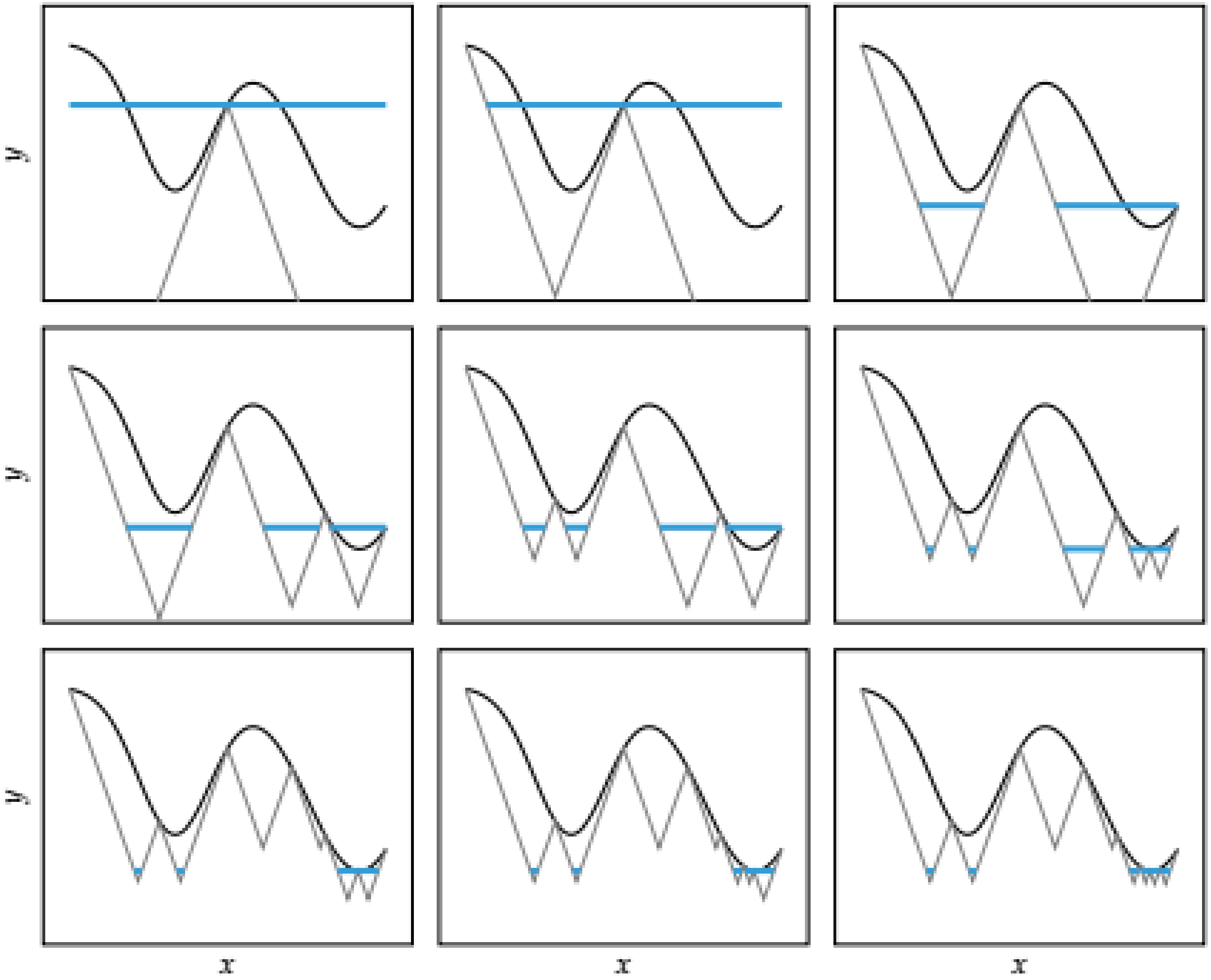


- 每次迭代中，最小值所在的不确定区域计算为：
- $\left[x^{(i)} - \frac{1}{l}\left(f(x_{min}) - y^{(i)}\right), x^{(i)} + \frac{1}{l}\left(y^{(i)} - y_{min}\right)\right]$
- 只有当$y^{(i)} < y_{min}$时，这个点对确定区域有贡献

这里$(x^{(i)}, y^{(i)})$为下锯齿顶点，(x_{min}, y_{min})为最小的上锯齿顶点

- 例子
- 缺点？

要求李普希兹常数L，越大下界越差



第4章 一维搜索-不精确搜索

◆**下降方向法$x^{(k+1)} = x^{(k)} + \lambda d^{(k)}$,确定方向后，需要计算步长**

- 大步长收敛快，但可能错过最优值
- 小步长收敛慢
- 固定步长有时称为学习率
- 或者使用$\lambda_k = \lambda_1 \gamma^{k-1}, \gamma \in (0, 1)$这样的衰减步长因子
- 是否还有别的方法？

◆**不精确一维搜索：** $\min f(x)$

考虑从$x^{(k)}$点出发，沿方向$d^{(k)}$寻找新迭代点：$x^{(k)}+\lambda_k d^{(k)}$

要求：1° $f(x^{(k)}+\lambda_k d^{(k)})<f(x^{(k)})$;

2° $\lambda_k>0$不能太小。

总体希望收敛快，每一步不要求达到精确最小，速度快，随着步数增加，则整个过程达到收敛。

◆**Goldstein法：是一个实用方法**

- 设 $f:R^n\to R$。在x取方向d，有$\nabla f^T(x)d<0$ (即d为下降方向), 令 $s^{(k)}=x^{(k+1)}-x^{(k)}=\lambda_k d^{(k)}$求$\lambda$使

$$(1)\ f\left(x^{(k+1)}\right)-f\left(x^{(k)}\right)\le\rho\nabla f^T\left(x^{(k)}\right)s^{(k)}$$

$$(2)\ f\left(x^{(k+1)}\right)-f\left(x^{(k)}\right)\ge(1-\rho)\nabla f^T\left(x^{(k)}\right)s^{(k)}$$

其中$\rho\in(0,\frac{1}{2})$,实际中常取$\rho=0.1$或更小

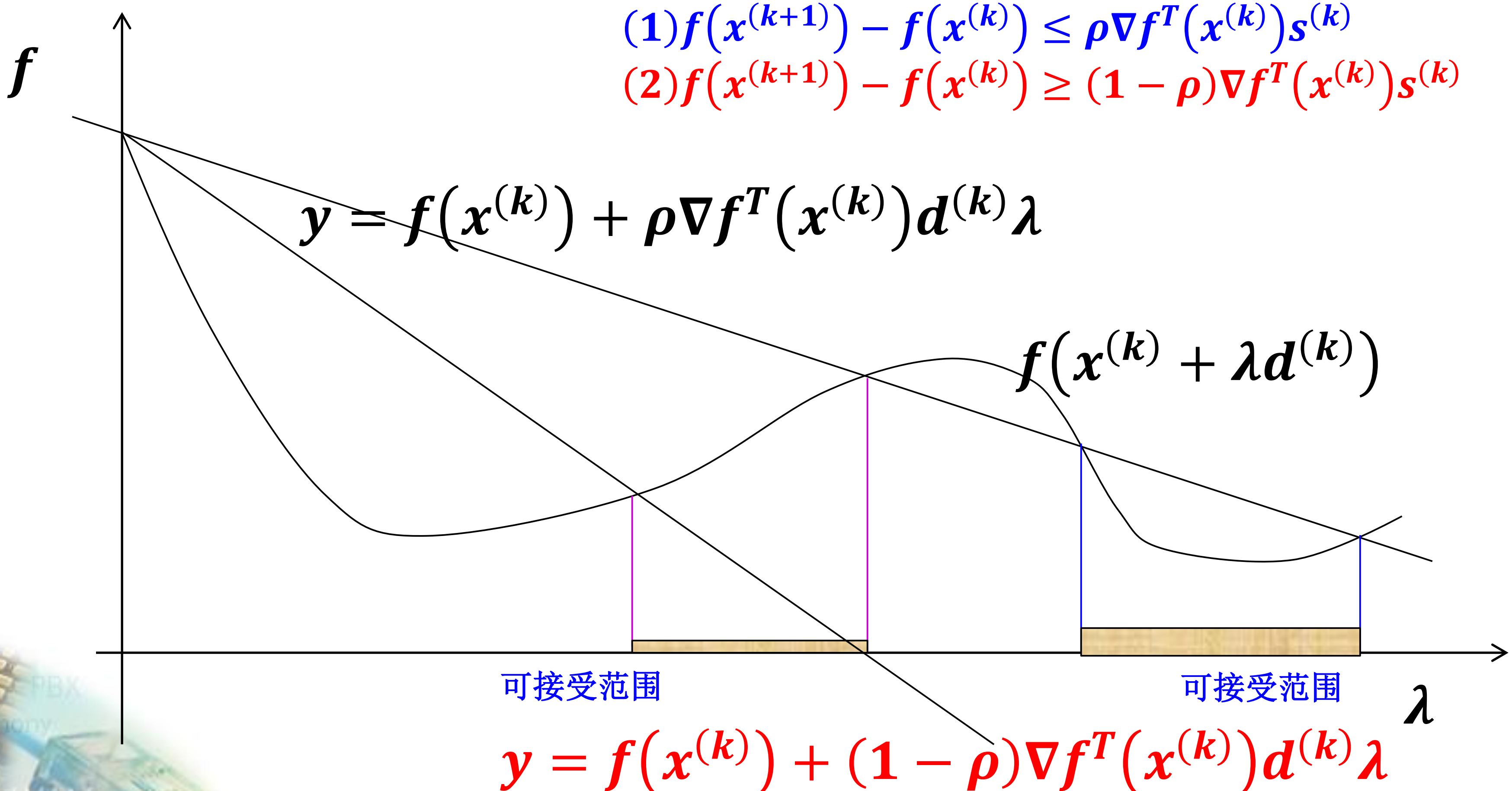
思考这两个条件的意义？

第4章 一维搜索-不精确搜索

◆不精确一维搜索(续)

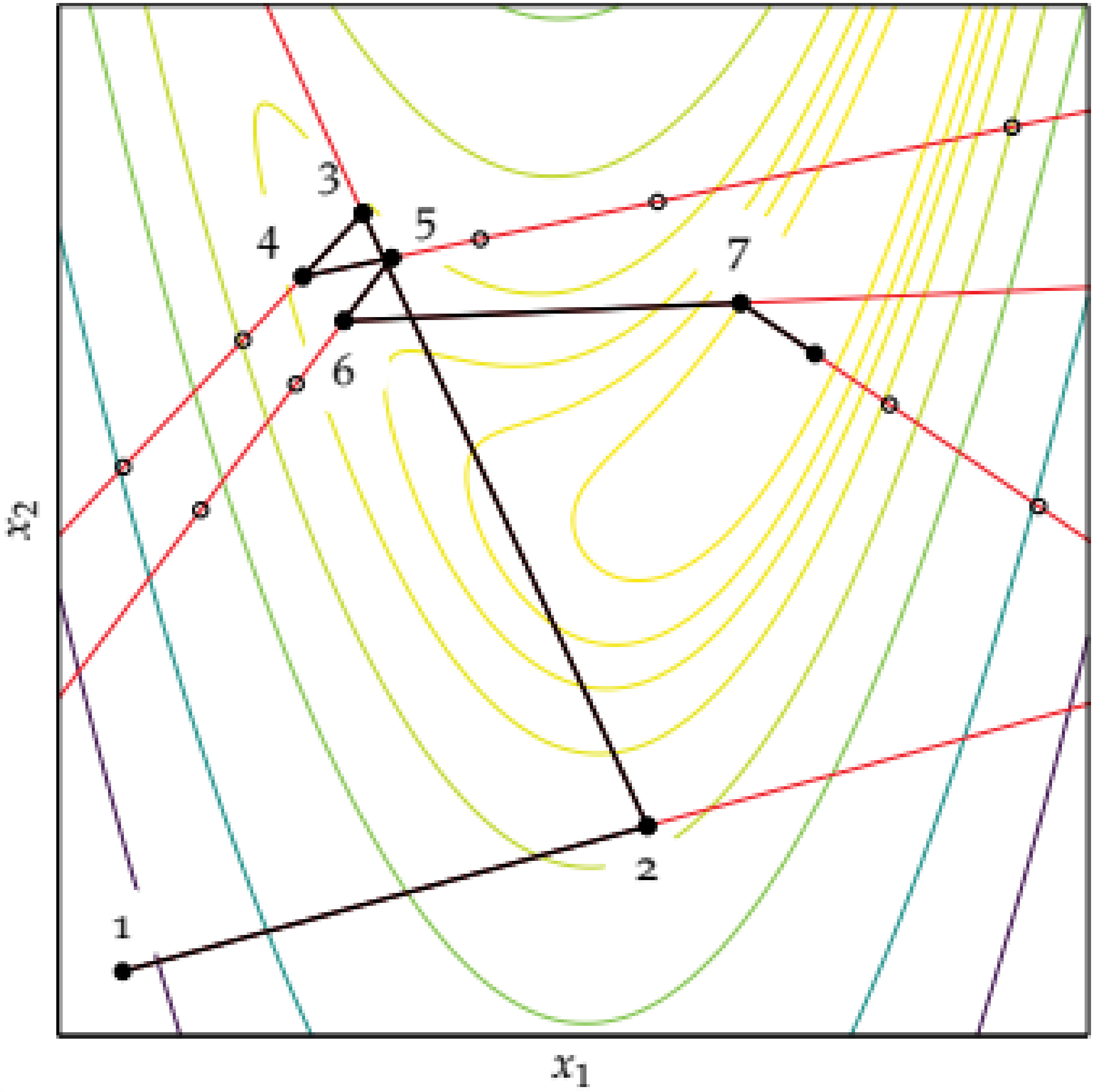
$$(1)f\left(x^{(k+1)}\right)-f\left(x^{(k)}\right)\le\rho\nabla f^T\left(x^{(k)}\right)s^{(k)}$$

$$(2)f\left(x^{(k+1)}\right)-f\left(x^{(k)}\right)\ge(1-\rho)\nabla f^T\left(x^{(k)}\right)s^{(k)}$$



◆经常在牛顿类方法中使用，但不太适合拟牛顿方法

◆七次线搜索

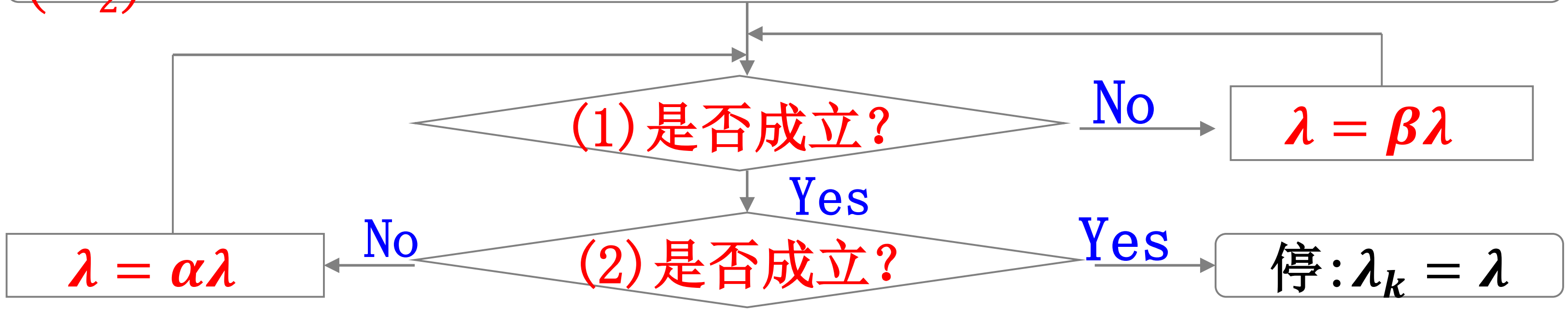


$(1) f\left(x^{(k+1)}\right)-f\left(x^{(k)}\right) \le \rho \nabla f^T\left(x^{(k)}\right) s^{(k)}$

$(2) f\left(x^{(k+1)}\right)-f\left(x^{(k)}\right) \ge (1-\rho) \nabla f^T\left(x^{(k)}\right) s^{(k)}$

◆不精确一维搜索(续)

$\rho \in \left(0,\frac{1}{2}\right), \lambda \in (0,\lambda_{\max})$，$\alpha > 1$(步长增大系数)，$\beta < 1$(步长缩短系数)



◆一般α, β的选择要适当，一般选取$\alpha = 1.5, \beta = 0.5$，一般情况下迭代几步就可得到解，利用这种不精确一维搜索，不少算法可以得到全局收敛性结果

◆1966年，Armijo给出与Goldstein法类似的不精确一维搜索规则，相当于替换(2)中的$(1-\rho) \to \mu\rho$,μ取5-10

◆1967年，Goldstein提出更一般的方法，把(2)式改为：

$$f\left(x^{(k+1)}\right) \ge f\left(x^{(k)}\right) + \sigma \nabla f^T\left(x^{(k)}\right) s^{(k)}, 其中\sigma \in (\rho, 1)$$

◆ Wolfe-Powell方法(1969,1976)

➢ 前面Goldstein方法中规则(2)改为对导数的要求

✓ (WP规则1) $f\left(x^{(k+1)}\right) \leq f\left(x^{(k)}\right) + \rho \nabla f^T\left(x^{(k)}\right) s^{(k)}$

✓ (WP规则2) $\nabla f^T\left(x^{(k+1)}\right) s^{(k)} \geq \sigma \nabla f^T\left(x^{(k)}\right) s^{(k)}$

➢ 其中 $\rho \in \left(0, \frac{1}{2}\right), \sigma \in (\rho, 1)$

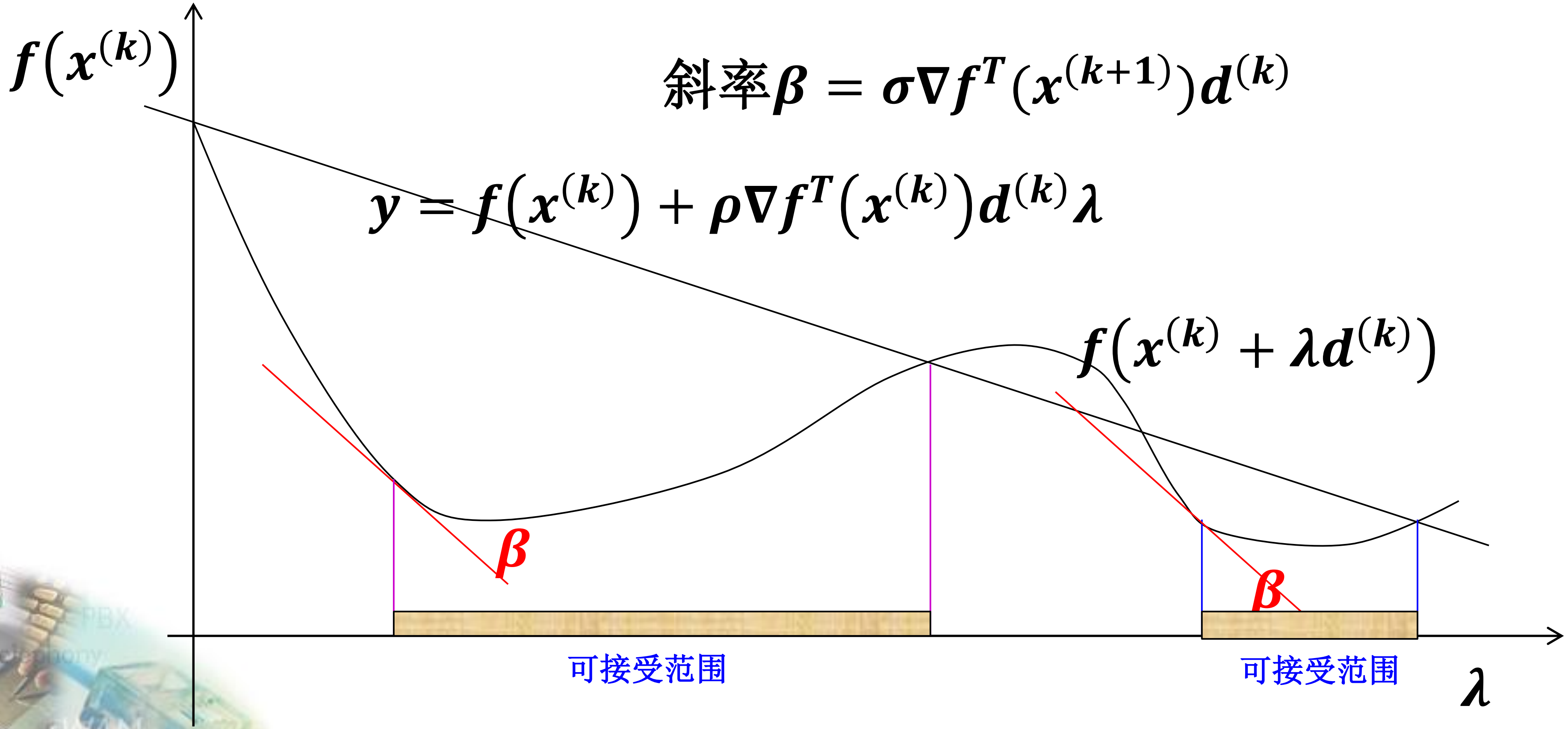
◆ WP规则(1)的意义如前

◆ WP规则(2)的意义？

➢ 表示要求在 $x^{(k+1)}$ 点对应的斜率不小于 $\sigma \nabla f^T\left(x^{(k)}\right) d^{(k)}$

第4章 一维搜索-不精确搜索

◆图示



第4章 一维搜索-不精确搜索

- **Wolfe-Powell方法(1969,1976)**
  - (WP规则1)$f\left(x^{(k+1)}\right) \le f\left(x^{(k)}\right) + \rho \nabla f^T\left(x^{(k)}\right) s^{(k)}$
  - (WP规则2)$\nabla f^T\left(x^{(k+1)}\right) s^{(k)} \ge \sigma \nabla f^T\left(x^{(k)}\right) s^{(k)}$
  - 其中$\rho \in \left(0, \frac{1}{2}\right), \sigma \in (\rho, 1)$

- 算法步骤如Goldstein类似，一般以$\lambda = 1$优先
- 此方法实算时，平均只需要2-3次函数值计算即可求得λ_k,对提高整体速度很有效。这个方法用于某些多变量算法中的线性搜索可得全局收敛性结果
- 如果需要较高的精度时，WP规则(2)可进一步改为:

$$(\text{WP改进规则2})\left|\nabla f^T\left(x^{(k+1)}\right) d^{(k)}\right| \le -\eta \nabla f^T\left(x^{(k)}\right) d^{(k)}, \eta \in (0, 1)$$

- 显然当$\eta = 0$,此即为精确一维搜索，但该方法不需要预先求出不确定区间，使用较为方便，又便于通过调整η来调整精确度要求，因而也常当做近似的精确一维搜索使用。当$\eta = 0.1$时，就可以看做近似的精确一维搜索
- 在绝大多数线搜索方法中使用，尤其适合拟牛顿方法

◆足够下降和回溯（Backtracking）方法

- 一般来说，目标函数值的充分下降并不能保证算法沿搜索方向有合理的进展，但如果线索搜通过采用所谓的**backtracking**方法可以获得合适的候选步长，也可以达到目标
- 算法如下（**backtracking** 线搜索）：

  选择 $\alpha_0 > 0, \rho \in (0,1), c \in (0,1)$; 令$\alpha \leftarrow \alpha_0$;

  Repeat until $f\left(x^{(k)} + \alpha p_k\right) \le f\left(x^{(k)}\right) + c\alpha \nabla f_k^T p_k$

  $$\alpha \leftarrow \rho\alpha;$$

  End(repeat)

  终止令$\alpha_k = \alpha$.

✓初始步长α_0在牛顿和拟牛顿方法中设为1，但在最速下降或共轭梯度中采用别的值，ρ值在迭代中可以变化，例如$0 < \rho_{low} < \rho < \rho_{high} < 1$

✓方法比较适合牛顿法，但不太适合拟牛顿和共轭梯度法

◆ Zoutendijk定理

- 优化中的迭代步骤中，方向p_k为下降方向，α_k满足Wolfe条件（别的条件也都可以）.假设f在R^n上下有界并在包含水平集$L=\{x: f(x)<f(x^{(0)})\}$的开集N上连续可微，$x^{(0)}$为迭代初始点，并假设∇f在N上是Lipschitz连续的，即：

$$||\nabla f(x)-\nabla f(\tilde{x})|| \le L||x-\tilde{x}||, \forall x, \tilde{x} \in N$$

- 则有：$\sum_{k\ge 0} cos^2\theta_k||\nabla f_k||^2 < \infty$---zoutendijk条件
- 注意：$cos\theta_k = \frac{-\nabla f_k^T p_k}{||\nabla f_k||||p_k||}$
- Lipschitz连续一般由平滑性假设都可以获得，因此在实际问题中一般都满足。由$zoutendijk$条件意味着$cos^2\theta_k||\nabla f_k||^2 \to 0$,如果角度满足$cos\theta_k \ge \delta > 0, \forall k$，则立得：$lim_{k\to\infty}||\nabla f_k|| = 0$
- 在拟牛顿方法中，$p_k = -B_k\nabla f_k$,若正定矩阵$||B_k||||B_k^{-1}|| \le M, \forall k$,则有$cos\theta_k \ge \frac{1}{M}$,利用zoutendijk条件，则有$lim_{k\to\infty}||\nabla f_k|| = 0$，意味着满足Wolfe条件的牛顿和拟牛顿方法，其偏转矩阵正定，具有有界条件数，则其全局收敛

第4章 一维搜索-最速下降法的收敛速率

◆ 对对称正定二次函数$f(x)=\frac{1}{2}x^TQx-b^Tx+c$应用最速下降法则有：

$$x^{(k+1)}=x^{(k)}-\left(\frac{\nabla f_k^T\nabla f_k}{\nabla f_k^T Q\nabla f_k}\right)\nabla f_k$$

◆ 令$||x||_Q^2=x^TQx$,则$\frac{1}{2}||x-x^*||_Q^2=f(x)-f(x^*)$-----误差范数,

$$\frac{1}{2}||x-x^*||_Q^2=\left\{1-\frac{\left(\nabla f_k^T\nabla f_k\right)^2}{\left(\nabla f_k^T Q\nabla f_k\right)\left(\nabla f_k^T Q^{-1}\nabla f_k\right)}\right\}||x^{(k)}-x^*||_Q^2$$

◆ 定理：当上述精确线搜索用于强凸二次函数时，其误差范数满足：

$$||\,x^{(k+1)}\text{-}x^*||_Q^2\le\left(\frac{\lambda_n-\lambda_1}{\lambda_n+\lambda_1}\right)^2||x^{(k)}-x^*||_Q^2$$

◆ 这里$\lambda_n\ge\lambda_{n-1}\ge\cdots\ge\lambda_2\ge\lambda_1>0$为Q的特征值

第4章 一维搜索-最速下降法的收敛速率

- 假设函数$f: R^n \to R$是连续二次可微，精确线索搜的最速下降法收敛到点x^*,该点的Hessian矩阵$\nabla^2 f(x^*)$正定。令$r \in \left(\frac{\lambda_n-\lambda_1}{\lambda_n+\lambda_1}, 1\right)$,这里$\lambda_n \geq \lambda_{n-1} \geq \cdots \geq \lambda_2 \geq \lambda_1 > 0$为$\nabla^2 f(x^*)$的特征值,则对足够大的$k$，有：
- $f\left(x^{(k+1)}\right) - f(x^*) \leq r^2\left[f\left(x^{(k)}\right) - f(x^*)\right]$
- 因此，一般不能期望非精确线索搜能改进收敛速度。即使其二阶Hessian矩阵具有良好的性质，例如条件数，也不能保证其快速收敛！
- 实际上，如何设计快速的能收敛的算法，一直是优化中的重点和难点!

第4章 一维搜索-结束

◆一维搜索

◆精确一维搜索

➢没有利用导数信息

✓黄金分割方法

✓斐波那契方法

➢利用导数信息

✓Newton法

✓插值法

◆近似一维搜索

➢**Goldstein法、Armijo**

➢**Wolfe-Powell方法**

✓一般2-3步找到