

第3章 常用的一维搜索方法

*n*元函数 $f : R^n \rightarrow R$

定理(必要条件) 设 $f : D \subseteq R^n \rightarrow R$

(1) x^* 为 D 的一个内点;

求解无约束优化问题 $\min_{x \in R^n} f(x)$

(2) $f(x)$ 在 x^* 可微;

(3) x^* 为 $f(x)$ 的极值点;

则 $\nabla f(x^*) = 0$.

定理(充分条件)

如果 f 是可微函数,

设 $f : D \subseteq R^n \rightarrow R$

根据一阶必要条件,

(1) x^* 为 D 的一个内点;

可令 $\nabla f(x) = 0$, 求得驻点;

(2) $f(x)$ 在 x^* 二次连续可微;

然后用充分条件进行判别, 求出所要的解

(3) $\nabla f(x^*) = 0$;

(4) $\nabla^2 f(x^*)$ 正定;

则 x^* 为 $f(x)$ 的严格局部极小点。

- 对某些较简单的函数, 这样做是可行的;
- 但对一般*n*元函数 f 来说, 由 $\nabla f(x) = 0$ 得到的是一个非线性方程组, 求解相当困难;
- 若 f 不可微, 则不能利用最优性条件求极小点;

为此常使用迭代法直接求解无约束优化问题。

1

2

迭代法的基本思想

为了求函数 f 的极小点, 首先给出一个初始估计 x^1 ,

然后按某种规则(算法)找出比 x^1 更好的估计 x^2 , $f(x^2) < f(x^1)$,

再按该规则找出比 x^2 更好的估计 x^3 , 从而得到一个序列 $\{x^k\}$,

若这个序列有极限 x^* , $\lim_{k \rightarrow \infty} \|x^k - x^*\| = 0$, 则称它收敛于 x^* .

若算法是有效的, 则序列 $\{x^k\}$ 收敛于 f 的极小点。

计算机只能进行有限次迭代, 故迭代法一般很难得到精确解, 只能得到近似解。当达到精度要求后, 计算机即可停止迭代。

迭代法的终止条件

停止迭代时要满足的条件称为终止条件。

理想的终止条件是

$$|f(x^k) - f(x^*)| < \varepsilon,$$

或者 $\|x^k - x^*\| < \varepsilon.$

注意 x^* 未知

3

4

迭代法的终止条件

实用的终止条件是利用相继两次的迭代结果给出

(1) 利用相继两次迭代绝对误差的终止条件

$$|f(x^{k+1}) - f(x^k)| < \varepsilon, \|x^{k+1} - x^k\| < \varepsilon,$$

(2) 利用相继两次迭代相对误差的终止条件

$$\frac{|f(x^{k+1}) - f(x^k)|}{|f(x^k)|} < \varepsilon, \frac{\|x^{k+1} - x^k\|}{\|x^k\|} < \varepsilon,$$

(3) 利用目标函数梯度的终止条件

$$\|\nabla f(x^k)\| < \varepsilon.$$

5

迭代法的收敛速度

设序列 $\{x^k\}$ 收敛于 x^* , 若存在与 k 无关的 $0 < \beta < \infty$ 和 $\alpha \geq 1$, 使 k 从某个 k_0 开始, 都有

$$\|x^{k+1} - x^*\| \leq \beta \|x^k - x^*\|^\alpha \quad (1)$$

则称 $\{x^k\}$ 收敛的阶为 α , 或者称 $\{x^k\}$ α 阶收敛。

在(1)中,

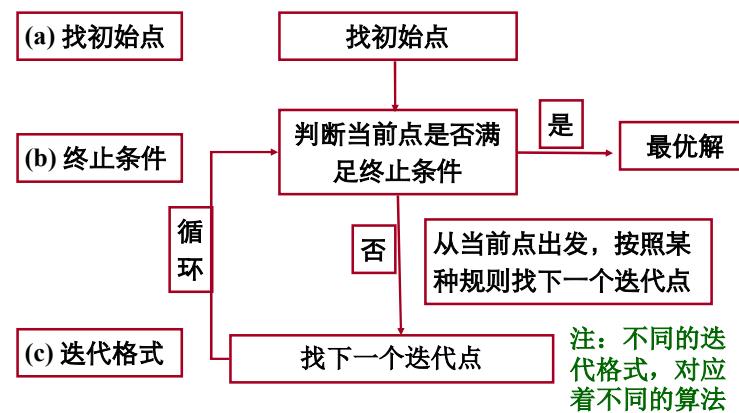
当 $\alpha = 2$ 时, 称为二阶收敛, 或称 $\{x^k\}$ 具有二阶收敛速度。

当 $1 < \alpha < 2$ 时, 称为超线性收敛。

当 $\alpha = 1$ 且 $0 < \beta < 1$ 时, 称为线性收敛或一阶收敛。

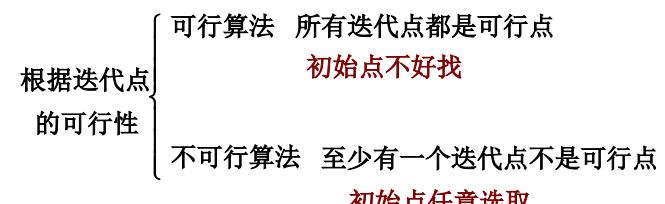
6

迭代法的一般框架



7

迭代法的分类



8

2

迭代法的分类

根据目标函数的下降特性

- 下降算法 每一迭代点的目标函数值都在下降
 $\forall k, f(x^{k+1}) \leq f(x^k)$
- 非单调下降算法 整体下降，局部上升
 $\exists k, f(x^{k+1}) > f(x^k),$
 $\exists l^k, f(x^{k+l^k}) \leq f(x^k)$
- 有更大的可能性找到全局极小点

9

迭代法的分类

根据是否计算函数的导数

- 直接法 不需要目标函数或者约束函数的导数信息
仅利用函数值，简单易用
- 解析法 需要目标函数或者约束函数的导数信息
利用导数信息，收敛性结果更强

10

迭代法的分类

根据迭代点是否沿某个方向产生

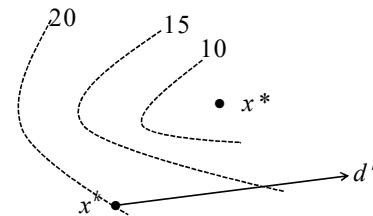
- 线搜索方法：迭代点沿某方向产生
每次迭代沿某个方向搜索下个迭代点，
最常见研究最多的方法
- 信赖域方法：迭代点在某区域内搜索产生
每次迭代在某区域内搜索下个迭代点，
近年来发展起来的一类方法

11

线搜索迭代法的基本思想

设当前迭代点是 x^k ，若 x^k 是在 R^n 上的局部极小点，则从 x^k 出发沿任意方向移动，都不能使目标函数值下降。

若从 x^k 出发沿方向 d^k 使得目标函数值下降，如图示



12

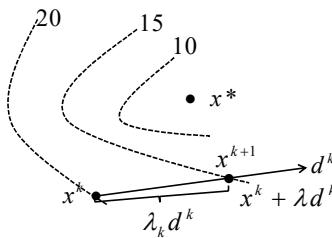
3

线搜索迭代法的基本思想

若从 x^k 出发沿方向 d^k 使得目标函数值下降，选定这个方向 d^k ，沿这个方向迈进适当一步，得到下一个迭代点 x^{k+1} ，使得 $f(x^{k+1}) < f(x^k)$ ，这相当于在射线 $x = x^k + \lambda d^k$ 上选定新点

$$x^{k+1} = x^k + \lambda_k d^k.$$

其中， d^k 称为搜索方向， λ_k 称为步长。



13

线搜索迭代法的步骤

步骤1 选定初始点 x^1 ，并令 $k := 1$ ；

步骤2 检验 x^k 是否满足终止条件，若是，则停止迭代，否则，转步骤3。

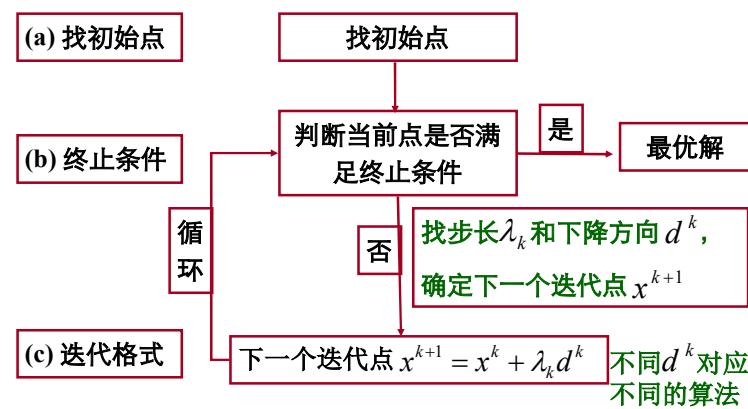
步骤3 确定搜索方向 d^k ；

步骤4 从 x^k 出发，沿方向 d^k 求步长 λ_k ，产生下一个迭代点 x^{k+1} ；令 $k := k + 1$ ，转步骤2。

各种算法的区别，主要在于搜索方向 d^k 的不同。

14

线搜索迭代法的框架分析



15

线搜索迭代法的框架分析

在确定了搜索方向后，下一步就要确定步长 λ_k ，常见的方法有3种：

(1) 令步长是一常数(例如令 $\lambda_k = 1$)，这样做不能保证目标函数值下降。
-----固定步长法

(2) 只要目标函数值下降，步长可任意选取。
-----可接受步长法

线搜索迭代法的框架分析----一维搜索

(3) 第三种方法的思路是:

沿搜索方向使目标函数值下降最多, 即沿射线

$x = x^k + \lambda d^k$ 求目标函数 f 的极小点:

$$\lambda_k := \arg \min f(x^k + \lambda d^k)$$

这项工作是求以 λ 为变量的一元函数 $f(x^k + \lambda d^k)$

的极小点 λ_k , 故这一过程称为 (精确) 一维搜索或

(精确) 线搜索或一维最优化, 确定的步长为最佳步长。

17

精确一维搜索的一个重要性质

在搜索方向上得到的最优点的梯度和该搜索方向正交。

定理1 设目标函数 f 具有一阶连续偏导数, x^{k+1} 按下述

$$\begin{cases} \lambda_k = \arg \min f(x^k + \lambda d^k) \\ x^{k+1} = x^k + \lambda_k d^k, \end{cases}$$

则有 $(\nabla f(x^{k+1}))^T d^k = 0$.

证明 构造函数 $\varphi(\lambda) = f(x^k + \lambda d^k)$, 则得 $\lambda_k = \arg \min \varphi(\lambda)$, 即 λ_k 是函数 $\varphi(\lambda)$ 的极小点, 故

$$0 = \varphi'(\lambda_k) = \varphi'(\lambda) \Big|_{\lambda=\lambda_k} = (\nabla f(x^k + \lambda d^k))^T d^k \Big|_{\lambda=\lambda_k} = (\nabla f(x^{k+1}))^T d^k.$$

18

一维搜索方法

精确一维搜索 非精确一维搜索

- (1) 试探法: 按照某种方式找试探点, 比较试探点处的函数值, 确定极小点。 “成功-失败” 法
- (2) 区间收缩法: 利用某种分割技术缩小最优解所在的区间 (称为搜索区间)。 二分法、0.618法
- (3) 函数逼近法: 用比较简单的函数近似原函数, 用简单函数的极小点估计原函数的极小点。

Newton法、插值法

19

一维搜索方法

非精确一维搜索:

通过少量的计算, 得到步长 λ_k , 使得下一个迭代点 $x^{k+1} = x^k + \lambda_k d^k$ 满足 $f(x^{k+1}) < f(x^k)$, 即目标函数“充分”下降。

(1) Armij-Goldstein准则

λ_k 要满足

$$\begin{aligned} f(x^k + \lambda_k d^k) &\leq f(x^k) + \rho \lambda_k g_k^T d^k, \\ f(x^k + \lambda_k d^k) &\geq f(x^k) + (1-\rho) \lambda_k g_k^T d^k, \end{aligned}$$

其中 $0 < \rho < \frac{1}{2}$.

20

5

一维搜索方法

非精确的一维搜索:

通过少量的计算, 得到步长 λ_k , 使得下一个迭代点

$x^{k+1} = x^k + \lambda_k d^k$ 满足 $f(x^{k+1}) < f(x^k)$, 即目标函数“充分”下降。

(2) Wolfe-Powell准则

λ_k 要满足

$$f(x^k + \lambda_k d^k) \leq f(x^k) + \rho \lambda_k g_k^T d^k$$

$$g_{k+1}^T d^k \geq \sigma g_k^T d^k,$$

其中 $\sigma \in (\rho, 1)$, $0 < \rho < 1$.

21

一维搜索方法

主要介绍下面几种方法

- “成功–失败”法(进退法)
- 0.618法(黄金分割法)
- 二分法(平分法)
- 牛顿法
- 插值法
- Armijii-Goldstein准则
- Wolfe-Powell准则

22

一维搜索方法

基本思想:

确定包含最优解的搜索区间;

采用某种插值或分割技术缩小搜索区间, 直至满足终止条件。

仅考虑一维无约束优化问题 $\min_{x \in R} f(x)$

约束优化问题 $\min_{a \leq x \leq b} f(x) = \min_{x \in R} F(x)$

$$F(x) = \begin{cases} f(x), & a \leq x \leq b \\ +\infty, & \text{否则} \end{cases}$$

23

常用的一维搜索方法

定义(搜索区间)

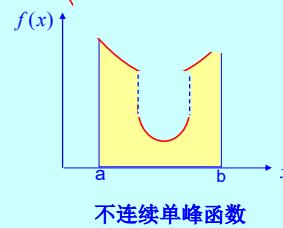
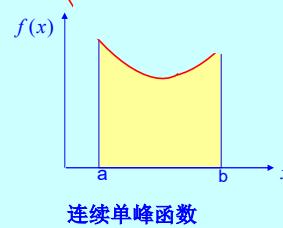
设 $f : R \rightarrow R$, x^* 是问题 $\min f(x)$ 的极小点, 若存在区间 $[a, b]$ 使得 $x^* \in (a, b)$, 则称 $[a, b]$ 为 $\min f(x)$ 的搜索区间。

定义(单峰函数)

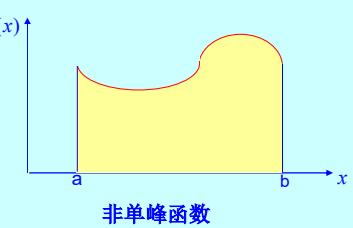
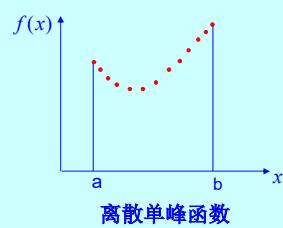
设 $f : R \rightarrow R$, x^* 是 $\min f(x)$ 的极小点, $[a, b]$ 是其搜索区间, 若 f 在 $[a, x^*]$ 上严格递减, 在 $[x^*, b]$ 上严格递增, 则称 f 为 $[a, b]$ 上的单峰函数。

24

常用的一维搜索方法



常用的一维搜索方法



25

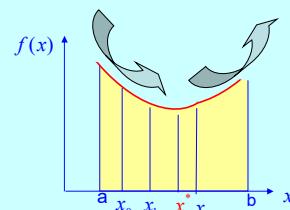
26

“成功—失败”法（进退法）

由单峰函数的性质可知，函数值在极小点左边严格下降，在右边严格上升。

基本思想：

从某个初始点出发,沿函数值下降的方向前进,直至满足终止条件。



“成功—失败”法（进退法）

步骤1 选取初始点 $x \in R$, 初始步长 $h > 0$ 及精度 $\varepsilon > 0$,
 $\varphi_1 = f(x)$. 注意: 初始步长不能选得太小

步骤2 计算 $\varphi_2 = f(x + h)$.

步骤3 若 $\varphi_2 < \varphi_1$ 搜索成功, 转步骤4;
 否则, 搜索失败, 转步骤5.

步骤4 令 $x := x + h$, $\varphi_1 := \varphi_2$, $h := 2h$, 转步骤 2.

步骤5 判断 $|h| \leq \varepsilon$? 若 $|h| \leq \varepsilon$, 停止迭代, $x^* = x$; 否则令
 $h = -h/4$, 转步骤 2.

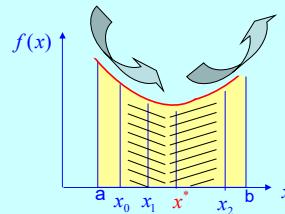
缺点: 效率低。优点: 可以求搜索区间。

27

28

“成功一失败”法（进退法）

由两边高、中间低的三点 x_0, x_1, x_2 , 可确定搜索区间为 $[x_0, x_2]$.



29

例1：设给定初始点为 a 及初始步长为 h , 求搜索区间 $[c, d]$.

前进运算

- 若 $f(a) > f(a+h)$, 则步长加倍, 计算 $f(a+3h)$;
- 若 $f(a+h) < f(a+3h)$, 则 $c = a, d = a+3h$;
- 否则将步长再加倍, 并继续前进.

后退运算

- 若 $f(a) < f(a+h)$, 则步长变小并反号, 计算 $f(a-\frac{1}{4}h)$;
- 若 $f(a) < f(a-\frac{1}{4}h)$, 则 $c = a - \frac{1}{4}h, d = a + h$;
- 否则将步长加倍, 并继续后退.

30

“成功一失败”法----算例

例 利用“成功-失败”法求 $f(x) = x^3 - 2x +$ 的搜索区间，取初始点 $x = -1/2$, 步长 $h = 1/2$.

解 取初始点 $x = -1/2$, 步长 $h = 1/2$,

$$f(x) = f(-1/2) = 15/8, f(x+h) = f(-1/2 + 1/2) = f(0) = 1,$$

$\because f(x) > f(x+h)$, 搜索成功, 步长加倍;

$$f(x+h+2h) = f(x+3h) = f(-1/2 + 3 \times 1/2) = f(1) = 0,$$

$\because f(x+h) > f(x+3h)$, 搜索成功, 步长加倍;

$$f(x+3h+4h) = f(x+7h) = f(-1/2 + 7 \times 1/2) = f(3) = 22,$$

$\therefore f(x+3h) < f(x+7h)$, 搜索失败, 停止迭代;

得到搜索区间为 $[x+h, x+7h] = [0, 3]$.

“成功一失败”法

使用成功失败法要注意

- h 取适当值

h 太大含多个单峰区间, h 太小迭代次数多;

- f 单调时算法不能终止

为终止算法, 对迭代次数加以限制。

31

32

0.618法（黄金分割法）

0.618法是求单峰函数极小值的一种试探法。

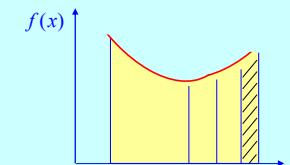
33

单峰函数的性质

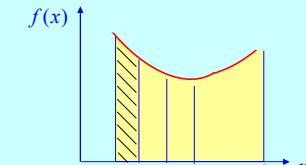
定理 设 f 是 $[a, b]$ 上的单峰函数, $a \leq x_1 < x_2 \leq b$,

- (1) 若 $f(x_1) \leq f(x_2)$, 则 $x^* \in [a, x_2]$, 如左下图
- (2) 若 $f(x_1) \geq f(x_2)$, 则 $x^* \in [x_1, b]$, 如右下图

若 $f(x_1) \leq f(x_2)$, 则 $x^* \in [a, x_2]$;

(1) 消去 $[x_2, b]$

若 $f(x_1) \geq f(x_2)$, 则 $x^* \in [x_1, b]$.

(2) 消去 $[a, x_1]$

单峰函数的性质

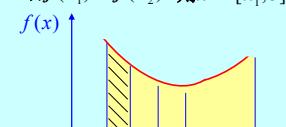
在单峰函数 f 的搜索区间 $[a, b]$ 上, 任取两个点, 比较这两点处的函数值, 可缩小搜索区间.

在新的搜索区间内, 保留了一点及其函数值, 只需再找一个点, 计算其函数值, 就可进一步缩小搜索区间.

若 $f(x_1) \leq f(x_2)$, 则 $x^* \in [a, x_2]$;

(1) 消去 $[x_2, b]$

若 $f(x_1) \geq f(x_2)$, 则 $x^* \in [x_1, b]$.

(2) 消去 $[a, x_1]$

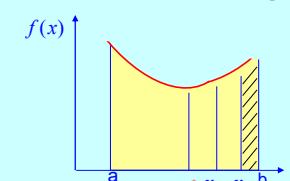
0.618法（黄金分割法）

定理 设 f 是 $[a, b]$ 上的单峰函数, $a \leq x_1 < x_2 \leq b$,

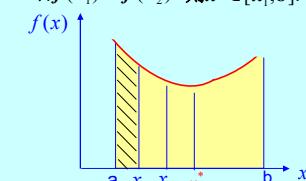
- (1) 若 $f(x_1) \leq f(x_2)$, 则 $x^* \in [a, x_2]$, 如左下图
- (2) 若 $f(x_1) \geq f(x_2)$, 则 $x^* \in [x_1, b]$, 如右下图

缩短区间的第一个原则---去坏留好原则

若 $f(x_1) \leq f(x_2)$, 则 $x^* \in [a, x_2]$;

(1) 消去 $[x_2, b]$

若 $f(x_1) \geq f(x_2)$, 则 $x^* \in [x_1, b]$.

(2) 消去 $[a, x_1]$

35

36

0.618法（黄金分割法）

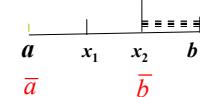
通过上述定理，选二点 $x_1 < x_2$ ，比较 $f(x_1)$ 与 $f(x_2)$ ，可去掉 $[a, x_1]$ 或者 $[x_2, b]$ 。考虑条件：

$$1^\circ \text{ 对称原则: } x_1 - a = b - x_2 \quad (1)$$

(使“坏”的情况去掉的区间长度不小于“好”的情况)

2° 保持缩减比，即 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， x_1 或 x_2 ，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 $[a, x_2]$ (去掉 $[x_2, b]$)，

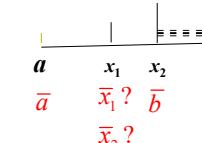


$$t = \frac{x_2 - a}{b - a} \quad (2)$$

37

0.618法（黄金分割法）

$$\begin{cases} x_1 - a = b - x_2 & (1) \\ t = \frac{x_2 - a}{b - a} & (2) \end{cases}$$



由(1)(2)可知，

$$\begin{cases} x_1 = a + (1-t)(b-a) \\ x_2 = a + t(b-a) \end{cases}$$

$t=?$ 可使得保留下来的 x_1 在下一次比较中成为一个试探点？

下一次比较中两个新的试探点是

$$\begin{cases} \bar{x}_1 = \bar{a} + (1-t)(\bar{b} - \bar{a}) = a + (1-t)(x_2 - a) = a + (1-t)t(b-a) \\ \bar{x}_2 = \bar{a} + t(\bar{b} - \bar{a}) = a + t(x_2 - a) = a + t^2(b-a) \end{cases}$$

38

0.618法（黄金分割法）

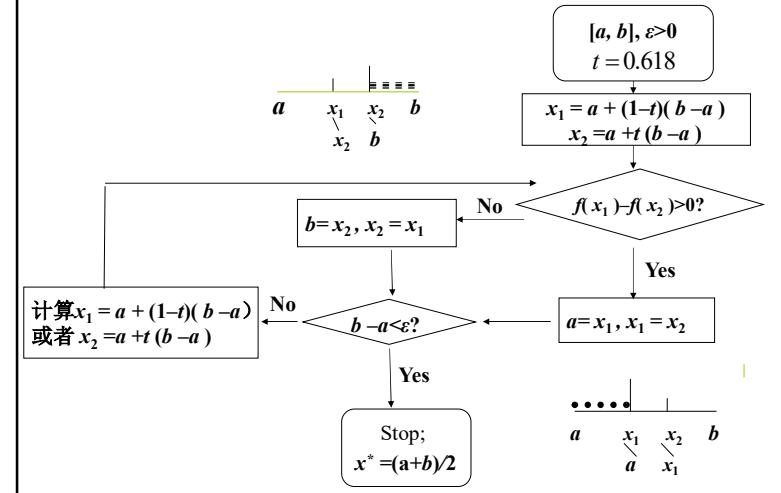
$$1-t=t^2, \text{ 故有 } t^2+t-1=0, \quad t = \frac{\sqrt{5}-1}{2} \approx 0.618$$

$$\begin{cases} x_1 = a + (1-t)(b-a) = a + 0.382(b-a) \\ x_2 = a + t(b-a) = a + 0.618(b-a) \end{cases}$$

当区间的长度充分小时，可将区间中点作为极小点的近似。

39

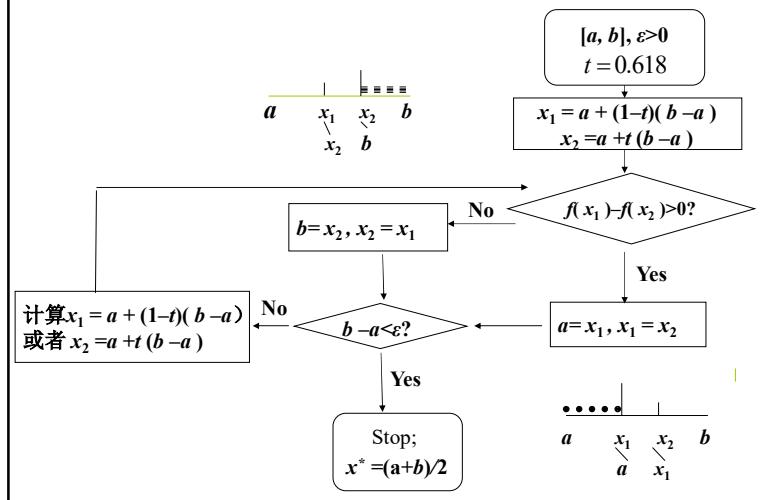
0.618 法----算法



40

10

0.618法----算法



41

0.618法（黄金分割法）

优点：不要求函数可微；
每次迭代只计算一次函数值，计算量小；
程序简单。

缺点：收敛速度慢。

0.618法----算例

$$x^* = \frac{\sqrt{6}}{3} \approx 0.816$$

例 试用0.618法求 $f(x) = x^3 - 2x + 1$ 的极小点。

给定初始区间[0,2], 收敛精度 $\varepsilon = 0.002$. $a = 0, h = \frac{2}{3}$

解 第一次缩短区间过程: $a = 0, b = 2$ $x_1 + x_2 = a + b$
计算两点及对应函数值:

$$x_1 = a + 0.382(b-a) = 0.764, f(x_1) = -0.0821,$$

$$x_2 = a + 0.618(b-a) = 1.236, f(x_2) = 0.4162,$$

作数值比较, 可见 $f(x_1) > f(x_2)$, x_2 再做置换 $x_1 = 1.236$

$$b := x_2 = 1.236, x_2 = 0.764, f(x_2) = -0.0821,$$

$$[a, b] = [0, 1.236], |b - a| = 1.236 > 0.002 = \varepsilon$$

43

$$[a, b] = [0, 1.236], x_2 = 0.764, f(x_2) = -0.0821,$$

第二次缩短区间过程:

$$x_1 + x_2 = a + b$$

$$x_1 = a + 0.382(b-a) = 0.472, f(x_1) = 0.1612,$$

作数值比较, $f(x_1) > f(x_2)$, x_1 再做置换:

$$a := x_1 = 0.472, x_1 = 0.764, f(x_1) = -0.0821,$$

$$[a, b] = [0.472, 1.236], |b - a| = 0.788 > 0.002 = \varepsilon;$$

第三缩短次区间过程:

$$x_2 = a + 0.618(b-a) = 0.944, \boxed{ }$$

作数值比较, $f(x_1) < f(x_2)$, x_2 再做置换:

$$b := x_2 = 0.944, x_2 = 0.764, f(x_2) = -0.0821,$$

$$[a, b] = [0.472, 0.944], |b - a| = 0.472 > 0.002 = \varepsilon$$

44

11

各次的迭代结果如下：

迭代次数	x_1	x_2	$f(x_1)$	$f(x_2)$	$[a,b]$	$ b-a $
第1次	0.764	1.236	-0.0821	0.4126	[0,1.236]	1.236
第2次	0.472	0.764	0.1612	-0.0821	[0.472,1.236]	0.788
第3次	0.764	0.944	-0.0821	-0.0468	[0.472,0.944]	0.472
第4次	0.652	0.764	-0.0268	-0.0821	[0.652,0.944]	0.292
第5次	0.764	0.832	-0.0821	-0.0881	[0.764,0.944]	0.230
第6次	0.832	0.906	-0.0881	-0.0683	[0.764,0.906]	0.124

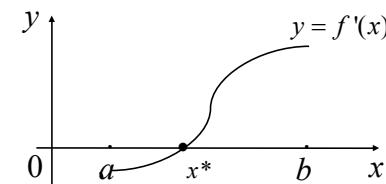
经过15次迭代可终止 ($2 * 0.618^k \leq 0.002 \Rightarrow k \geq 14.4$)

45

二分法---基本思想

给定可微函数 $f: [a,b] \rightarrow R$, 若 f 在 (a,b) 内存在极小点, 则该点处的导数一定为0.

如果 $f'(a) < 0, f'(b) > 0$,
则在 (a,b) 内一定存在一点 x^* , 使得 $f'(x^*) = 0$.

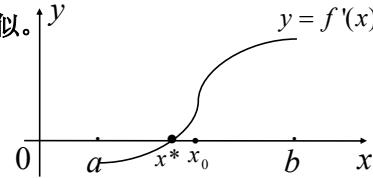


46

二分法---基本思想

为求 x^* , 可取 $x_0 = \frac{a+b}{2}$, $f'(a) < 0, f'(b) > 0$,
若 $f'(x_0) = 0$, x_0 为极小点, $x^* = x_0$;
 $f'(x_0) > 0$, x_0 在上升段, $x^* < x_0$, 去掉 $[x_0, b]$;
 $f'(x_0) < 0$, x_0 在下降段, $x^* > x_0$, 去掉 $[a, x_0]$.

用 $[a, x_0]$ 或者 $[x_0, b]$ 替代搜索区间 $[a, b]$, 继续这个过程,
逐步将搜索区间变小, 当搜索区间的长度充分小时, 可将
区间中点作为极小点的近似.



二分法---计算步骤

- 步骤1 计算 $x_0 = \frac{a+b}{2}$
- 步骤2 若 $f'(x_0) < 0$, 令 $a = x_0$, 转步骤3;
若 $f'(x_0) > 0$, 令 $b = x_0$, 转步骤3;
若 $f'(x_0) = 0$, 停止, $x^* = x_0$.
- 步骤3 若 $|b-a| < \varepsilon$, 则 $x^* = \frac{a+b}{2}$, 停止, 否则, 转步骤1.

47

48

12

二分法---优缺点

- 优点:**
- 计算量较少；
 - 总能收敛到一个局部极小点。
- 缺点:**
- 收敛慢。

49

二分法----算例

$$x^* = \frac{\sqrt{6}}{3} \approx 0.816$$

例 试用二分法求 $f(x) = x^3 - 2x + 1$ 的极小点，
给定初始区间 $[0,2]$ ，收敛精度 $\varepsilon = 0.004$ 。

解 $f'(x) = 3x^2 - 2$, 因为 $f'(0) = -2$, $f'(2) = 10$, 所以函数
 $f(x) = x^3 - 2x + 1$ 在 $(0,2)$ 内有导数为 0 点 x^* , 且
 $f''(x^*) = 6x^* > 0$ ($x^* \in (0,2)$), 故 x^* 是函数的极小点。

第一次缩短区间过程

$$x_0 = \frac{a+b}{2} = 1, \quad f'(x_0) = f'(1) = 1 > 0, \quad \text{故 } b := x_0 = 1,$$

$$[a,b] = [0,1], \quad |b-a| = 1 > 0.004 = \varepsilon;$$

50

第二次缩短区间过程

$$x_0 = \frac{a+b}{2} = \frac{1}{2}, \quad f'(x_0) = f'(\frac{1}{2}) = -\frac{5}{4} < 0, \quad \text{故 } a := x_0 = \frac{1}{2},$$

$$[a,b] = [\frac{1}{2},1], \quad |b-a| = \frac{1}{2} > 0.004 = \varepsilon;$$

第三次缩短区间过程

$$x_0 = \frac{a+b}{2} = \frac{3}{4}, \quad f'(x_0) = f'(\frac{3}{4}) = -\frac{5}{16} < 0, \quad \text{故 } a := x_0 = \frac{3}{4},$$

$$[a,b] = [\frac{3}{4},1], \quad |b-a| = \frac{1}{4} > 0.004 = \varepsilon;$$

经过9次迭代可终止 ($2 * 0.5^k \leq 0.004 \Rightarrow k \geq 8.97$)

51

各次的迭代结果如下:

$$x^* = \frac{\sqrt{6}}{3} \approx 0.816$$

迭代次数	$x_0 = (a+b)/2$	$f'(x_0)$	$[a,b]$	$ b-a $
第1次	$x_0 = 1$	1	[0,1]	1
第2次	$x_0 = 1/2$	-5/4	[1/2,1]	1/2
第3次	$x_0 = 3/4$	-5/16	[3/4,1]	1/4
第4次	$x_0 = 7/8$	19/64	[3/4,7/8]	1/8
第5次	$x_0 = 13/16$	-0.0195	[13/16,7/8]	1/16
第6次	$x_0 = 27/32$	0.0136	[13/16,27/32]	1/32
第7次	$x_0 = 53/64$	0.0574	[13/16,53/64]	1/64
第8次	$x_0 = 105/128$	0.0184	[13/16,105/128]	1/128
第9次	$x_0 = 209/256$	-0.0004	[209/256,105/128]	1/256

迭代9次后, $|b-a|=0.00391<0.004$, 故 $x^* \approx 0.81836$.

52

13

牛顿法 (Newton) --- 基本思想 函数逼近法

在极小点附近用二阶泰勒展开多项式近似 f , 用二阶泰勒展开多项式的极小点估计 f 的极小点。

f 在 x_k 处作二阶泰勒展开:

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2} f''(x_k)(x - x_k)^2 + o((x - x_k)^2)$$

略去高阶项得

$$f(x) \approx q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2} f''(x_k)(x - x_k)^2$$

两边对 x 求导, $f'(x) \approx q'(x) = f'(x_k) + f''(x_k)(x - x_k)$

令 $q'(x)=0$, 得到

$$x^* \approx x_k - \frac{f'(x_k)}{f''(x_k)}$$

当 f 是二次函数时, 一次迭代就可得到极小点。

53

牛顿法 (Newton) --- 基本思想

取 $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$, 作为新的迭代点,

若 $|f'(x_{k+1})| < \varepsilon$, 则停止, 得到近似驻点,
否则继续迭代, 直到达到精度, 得到 f 的一个驻点 x^* .

以上过程即 **Newton 法**。

在一定条件下 (例如 $f''(x^*) > 0$), 这个驻点是极小点。

54

牛顿法 (Newton) --- 计算步骤

步骤1 给定初始点 x_1 , $\varepsilon > 0$, 令 $k=1$.

步骤2 计算 $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$.

步骤3 若 $|f'(x_{k+1})| < \varepsilon$, 停止, $x^* \approx x_{k+1}$,

否则令 $k=k+1$, 转步骤2.

牛顿法 (Newton) --- 优缺点

优点:

- 收敛速度快;
- 局部二阶收敛。

缺点:

- 须计算二阶导数, 工作量大;
- 对初始点要求高(初始点离极小点不能远, 否则迭代序列可能发散或者收敛到非极小点);
- 局部收敛。

55

56

14

Newton法----算例 $f'(x) = 4(x-4)(x^2+x+1)$, $x^* = 4$.

例 试用Newton法求 $f(x) = x^4 - 4x^3 - 6x^2 - 16x + 4$ 的极小点 ($x_1 = 6, \varepsilon = 10^{-2}$).

解 $f'(x) = 4x^3 - 12x^2 - 12x - 16$, $f''(x) = 12x^2 - 24x - 12$,

$$x_2 = x_1 - \frac{f'(x_1)}{f''(x_1)} = 6 - \frac{f'(6)}{f''(6)} = 6 - \frac{89}{69} = 4.75$$

$f'(x_2) = f'(4.75) = 84.94 > 10^{-2}$, 继续迭代;

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

$$= 4.75 - \frac{f'(4.75)}{f''(4.75)} = 4.75 - \frac{84.94}{144.75} = 4.163$$

$f'(x_3) = f'(4.163) = 14.666 > 10^{-2}$, 继续迭代;

Newton法----算例

$$x_3 = 4.163$$

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

$$= 4.163 - \frac{f'(4.163)}{f''(4.163)} = 4.163 - \frac{14.666}{96.055} = 4.010$$

$f'(x_4) = f'(4.010) = 0.8436 > 10^{-3}$, 继续迭代;

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

$$= 4.010 - \frac{f'(4.010)}{f''(4.010)} = 4.010 - \frac{0.8436}{84.7212} = 4.00004$$

$f'(x_5) = f'(4.00004) = 0.0034 < 10^{-2}$,

得到近似解 $x^* \approx 4.00004$. **Newton法收敛速度快**

57

58

区间收缩法

给定 f 的一个搜索区间, 在区间内选择两个试探点, 比较试探点处的函数值, 将搜索区间变短, 当区间的长度充分小时, 可将区间中的任何一点作为 f 极小点的近似。

用成功失败法寻找“高-低-高”的三点 $x_1 < x_2 < x_3$, 得到搜索区间 $[x_1, x_3]$, x_2 作为区间收缩法中的一个试探点, 只需找到另一个试探点就可以缩短搜索区间。

另外一个试探点利用插值法寻找

插值法

用成功失败法寻找“高-低-高”的三点 $x_1 < x_2 < x_3$, 得到搜索区间 $[x_1, x_3]$, x_2 作为区间收缩法中的一个试探点, 只需找到另一个试探点就可以缩短搜索区间。

另外一个试探点利用插值法寻找

利用 f 在 2 个或 3 个点的函数值或导数, 构造一个 2 次或 3 次多项式, 以该多项式的极小点作为一个试探点。

3 点 2 次, 2 点 2 次, 4 点 3 次, 3 点 3 次, 2 点 3 次等插值法。下面以 3 点 2 次插值法 (二次插值法) 为例。

59

60

15

插值法

下面以3点2次插值法（二次插值法或称抛物线法）为例：

利用 $y = f(x)$ 在区间 $x_1 < x_2 < x_3$ 的函数值 $f(x_1) > f(x_2) < f(x_3)$
作出如下的二次插值多项式 $P(x) = a_0 + a_1x + a_2x^2$

它应满足条件

$$P(x_1) = a_0 + a_1x_1 + a_2x_1^2 = f_1 = f(x_1) \quad (1)$$

$$P(x_2) = a_0 + a_1x_2 + a_2x_2^2 = f_2 = f(x_2) \quad (2)$$

$$P(x_3) = a_0 + a_1x_3 + a_2x_3^2 = f_3 = f(x_3) \quad (3)$$

61

插值法---求二次插值多项式的极小点

$$P(x) = a_0 + a_1x + a_2x^2$$

从极值的必要条件 $P'(\bar{x}) = a_1 + 2a_2\bar{x} = 0$, 可求得

$$\bar{x} = -a_1 / 2a_2.$$

求出系数 a_1 和 a_2 , 就可得到极小点的表达式。

联立方程组(1)、(2)、(3)

$$a_1(x_1 - x_2) + a_2(x_1^2 - x_2^2) \equiv f_0 + f_1 x_1 + f_2 x_2^2 - \bar{x}_3 f_3 + a_2(f_1 x_1 x_2^2) = f_2(1)f_3 \quad (1)$$

$$P(x_2) = a_0 + a_1x_2 + a_2x_2^2 = f_2 = f(x_2) \quad (2)$$

$$a_1 = \frac{(x_2^2 - x_3^2)f_1 + (x_3^2 - x_1^2)f_2 + (x_1^2 - x_2^2)f_3}{(x_1^2 - x_2^2)(x_2^2 - x_3^2)} = f(x_3) \quad (3)$$

$$a_1(x_1 - x_2) + a_2(x_1^2 - x_2^2) \equiv f_1 - f_2, a_2 = -\frac{a_1(x_1^2 - x_2^2)}{a_1(x_1^2 - x_2^2)(x_2^2 - x_3^2)} = f_2 - f_3$$

62

插值法---求二次插值多项式的极小点

所以

$$\bar{x} = -a_1 / 2a_2 = \frac{1}{2} \frac{(x_2^2 - x_3^2)f_1 + (x_3^2 - x_1^2)f_2 + (x_1^2 - x_2^2)f_3}{(x_1^2 - x_2^2)(x_2^2 - x_3^2)} \quad (4)$$

(4)可以进一步简化

当 x_1, x_2, x_3 等距时, 即 $x_2 - x_1 = x_3 - x_2 = h$ 时, 上面的式子可化简

$$\begin{aligned} \bar{x} &= \frac{1}{2} \frac{-h(2x_2 + h)f_1 + 2h(x_2 + h + x_2 - h)f_2 - h(2x_2 - h)f_3}{-hf_1 + 2hf_2 - hf_3} \\ &= x_2 + \frac{1}{2} \frac{h(f_1 - f_3)}{f_1 - 2f_2 + f_3} \end{aligned} \quad (5)$$

63

插值法---求二次插值多项式的极小点

简化极小点的公式(4)

联立方程组(1)、(2)、(3), 可得

$$a_1(x_1 - x_3) + a_2(x_1^2 - x_3^2) = f_1 - f_3, a_1(x_1 - x_2) + a_2(x_1^2 - x_2^2) = f_1 - f_2$$

从而

$$a_1 + a_2(x_1 + x_3) = \frac{f_1 - f_3}{x_1 - x_3} := c_1, \quad a_1 + a_2(x_1 + x_2) = \frac{f_1 - f_2}{x_1 - x_2} := c_3$$

所以

$$a_1 = c_1 - a_2(x_1 + x_3), \quad a_2 = \frac{c_1 - c_3}{x_3 - x_2} = \frac{c_1 - f_1 - f_2}{x_1 - x_2} = \frac{f_1 - f_2 - c_1}{x_2 - x_3} := c_2$$

64

16

插值法---求二次插值多项式的极小点

简化极小点的公式(4)

$$a_1 = c_1 - a_2(x_1 + x_3), \quad a_2 = \frac{c_1 - c_3}{x_3 - x_2} = \frac{c_1 - f_2}{x_1 - x_2} = \frac{\frac{f_1 - f_2}{x_1 - x_2} - c_1}{x_2 - x_3} := c_2$$

因此

$$\begin{aligned}\bar{x} &= -a_1 / 2a_2 = \frac{c_1 - a_2(x_1 + x_3)}{2a_2} \\ &= \frac{c_1}{2a_2} - \frac{x_1 + x_3}{2} \\ &= \frac{c_1}{2c_2} - \frac{x_1 + x_3}{2} \\ &= \frac{1}{2} \left(x_1 + x_3 - \frac{c_1}{c_2} \right)\end{aligned}$$

65

插值法---求二次插值多项式的极小点

简化极小点的公式(4)

$$\bar{x} = \frac{1}{2} \left(x_1 + x_3 - \frac{c_1}{c_2} \right)$$

$$\begin{cases} \bar{x} = -a_1 / 2a_2 = \frac{1}{2} \left(x_1 + x_3 - \frac{c_1}{c_2} \right) \\ c_1 = \frac{f_1 - f_3}{x_1 - x_3}, \quad c_2 = \frac{\frac{f_1 - f_2}{x_1 - x_2} - c_1}{x_2 - x_3}. \end{cases} \quad (6)$$

66

插值法---求二次插值多项式的极小点

简化极小点的公式(4)到(6)

$$\bar{x} = -a_1 / 2a_2 = \frac{1}{2} \left(\frac{(x_2^2 - x_3^2) f_1 + (x_3^2 - x_1^2) f_2 + (x_1^2 - x_2^2) f_3}{(x_2 - x_3) f_1 + (x_3 - x_1) f_2 + (x_1 - x_2) f_3} \right) \quad (4)$$

$$\begin{cases} \bar{x} = -a_1 / 2a_2 = \frac{1}{2} \left(x_1 + x_3 - \frac{c_1}{c_2} \right) \\ c_1 = \frac{f_1 - f_3}{x_1 - x_3}, \quad c_2 = \frac{\frac{f_1 - f_2}{x_1 - x_2} - c_1}{x_2 - x_3}. \end{cases} \quad (6)$$

乘除的运算次数从14降为5

67

插值法---算法步骤:

步骤1 (用成功失败法) 寻找“高-低-高”三点: 即三点满足

$$x_1 < x_2 < x_3, f(x_1) > f(x_2) < f(x_3).$$

步骤2 确定二次插值多项式 $P(x) = a_0 + a_1x + a_2x^2$, 求出 P 的极小点 \bar{x} (因 $P(x_1) > P(x_2) < P(x_3)$, 故 $a_2 > 0$, $\bar{x} \in [x_1, x_3]$.)

步骤3 若 $|x_2 - \bar{x}| < \varepsilon$, 则迭代结束, 取 $x^* = \bar{x}$, 否则在点 x_1, x_2, x_3, \bar{x} 中, 选取使 f 最小的点作为新的 x_2 , 并使新的 x_1, x_3 各是新的 x_2 近旁的左右两点, 转步骤2, 继续迭代, 直到满足终止条件。

68

17

例 用二次插值法求函数 $f(x)=3x^3-4x+2$ 的极小点, 给定 $x_0=0$,

$$h=1, \varepsilon=0.2.$$

$$x^* = \frac{2}{3} \approx 0.667$$

解 1) 确定初始搜索区间

初始区间 $[a,b]=[0,3]$, 另有一中间点 $x_2=1$.

2) 用二次插值法逼近极小点

(1) 相邻三点及其函数值: $x_1=0, x_2=1, x_3=3$;

$$f_1=2, f_2=1, f_3=71.$$

根据公式计算插值多项式的极小点

$$\bar{x} = -a_1 / 2a_2 = \frac{1}{2}(x_1 + x_3 - \frac{c_1}{c_2}),$$

$$c_1 = \frac{f_1 - f_3}{x_1 - x_3}, \quad c_2 = \frac{\frac{f_1 - f_2}{x_1 - x_2} - c_1}{\frac{f_2 - f_3}{x_2 - x_3} - c_1}$$

$$\bar{x} = 0.542, \quad |x_2 - \bar{x}| = 1 - 0.542 = 0.458 > \varepsilon = 0.2,$$

69

$$x_1=0, x_2=1, x_3=3; f_1=2, f_2=1, f_3=71 \quad \bar{x} = 0.542, f(\bar{x}) = 0.31,$$

由于 $f(\bar{x}) = 0.31 < f_2 = 1, \bar{x} = 0.542 < x_2 = 1$,
故新区间 $[a,b]=[a,x_2]=[0,1]$,

在新区间, 相邻三点及其函数值: $x_1=0, x_2=0.542, x_3=1$;

$$f_1=2, f_2=0.31, f_3=1.$$

(2) 根据公式计算插值多项式的极小点

$$\bar{x} = -a_1 / 2a_2 = \frac{1}{2}(x_1 + x_3 - \frac{c_1}{c_2}),$$

$$c_1 = \frac{f_1 - f_3}{x_1 - x_3}, \quad c_2 = \frac{\frac{f_1 - f_2}{x_1 - x_2} - c_1}{\frac{f_2 - f_3}{x_2 - x_3} - c_1}$$

$$\bar{x} = 0.608,$$

70

$$x_2 = 0.542, \quad \bar{x} = 0.608,$$

$$|x_2 - \bar{x}| = 0.608 - 0.542 = 0.066 < \varepsilon = 0.2,$$

迭代终止。

故 $x^* \approx \bar{x} = 0.608, f^* \approx f(\bar{x}) = 0.242$.

$$x^* = \frac{2}{3} \approx 0.667$$

71

非精确一维搜索方法

为什么考虑非精确一维搜索方法?

精确一维搜索通过求解一维优化 $\min f(x^k + \lambda d^k)$

得到最佳步长 $\lambda_k := \arg \min f(x^k + \lambda d^k)$.

精确一维搜索

- 计算量往往很大;
- 当迭代点远离极小点时, 效率很低;
- 很多方法的收敛速度并不依赖于精确一维搜索的过程。

72

18

非精确一维搜索方法

60s中期以前，线搜索方法是精确线搜索方法一统天下。
Armijo(1966), Goldstein(1965)提出非精确线性搜索方法，这种方法由于计算量小、效率高成了现在流行的线搜索方法。

保证目标函数在每次迭代都有满意下降量的方法，就是非精确一维搜索方法或称为可接受一维搜索方法。

73

非精确一维搜索方法

保证目标函数在每次迭代有满意下降量的方法，就是非精确一维搜索方法或称为可接受一维搜索方法。

非精确一维搜索方法可大大节省计算量。

在非精确一维搜索中，通常要求 $f(x^{k+1})$ 比 $f(x^k)$ 下降一定的数量，从而使得 $f(x^{k+1}) - f(x^k)$ 达到一个满意水平，此时的 λ_k 就是可接受步长。

74

非精确一维搜索方法---Armijo-Goldstein准则

如何选取可接受步长 λ_k 呢？

d^k 是下降方向，记 $g_k = \nabla f(x^k)$ ，则 $g_k^T d^k < 0$.

为减少计算量，可给出 f 下降量的一个下界和上界，

比如 $-\lambda \rho g_k^T d^k \left(0 < \rho < \frac{1}{2} \right)$ 和 $-\lambda(1-\rho)g_k^T d^k \left(0 < \rho < \frac{1}{2} \right)$ ，即要求

$$f(x^k) - f(x^k + \lambda d^k) \geq -\lambda \rho g_k^T d^k,$$

$$f(x^k) - f(x^k + \lambda d^k) \leq -\lambda(1-\rho)g_k^T d^k.$$

75

非精确一维搜索方法---Armijo-Goldstein准则

这就是Armijo-Goldstein准则，即 λ_k 要满足

$$f(x^k + \lambda_k d^k) - f(x^k) \leq \lambda_k \rho g_k^T d^k, \quad (*)$$

和

$$f(x^k + \lambda_k d^k) - f(x^k) \geq \lambda_k (1-\rho) g_k^T d^k, \quad (**)$$

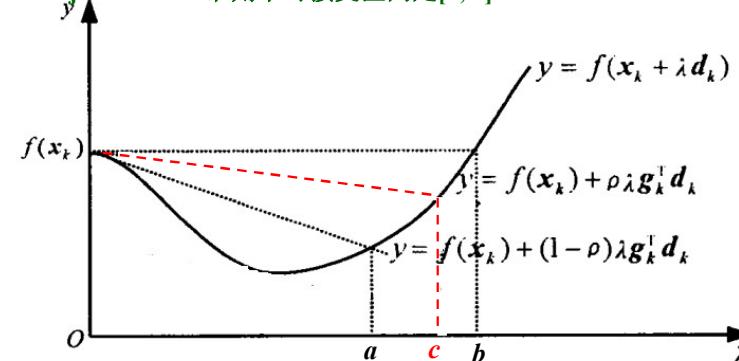
其中 $0 < \rho < \frac{1}{2}$.

76

19

非精确一维搜索方法---Armijo-Goldstein准则

Armijo-Goldstein准则下可接受区间是 $[a, b]$



77

非精确一维搜索方法---Armijo-Goldstein准则

令 $\varphi(\lambda) = f(x^k + \lambda d^k)$ ，则 Armijo-Goldstein 准则

$$\varphi(\lambda_k) \leq \varphi(0) + \rho \lambda_k \varphi'(0)$$

$$f(x^k + \lambda_k d^k) - f(x^k) \leq \lambda_k \rho g_k^T d^k$$

和

$$\varphi(\lambda_k) \geq \varphi(0) + (1 - \rho) \lambda_k \varphi'(0)$$

$$f(x^k + \lambda_k d^k) - f(x^k) \geq \lambda_k (1 - \rho) g_k^T d^k$$

78

非精确一维搜索方法---Armijo-Goldstein准则

Armijo-Goldstein准则计算 λ_k 的步骤

步骤1 选取初始数据 比可接受区间大

在区间 $[0, +\infty)$ 或 $[0, \lambda_{\max}]$ 中取定初始点 λ_0 ，计算 $\varphi(0), \varphi'(0)$. 给定 $\rho \in (0, \frac{1}{2})$, $\alpha > 1$. 令 $a_0 = 0, b_0 = +\infty$ 或 $\lambda_{\max}, l = 0$.

步骤2 检查下降量要求

计算 $\varphi(\lambda_l)$ ，若 $\varphi(\lambda_l) \leq \varphi(0) + \rho \lambda_l \varphi'(0)$ ，转步骤3.
否则，令 $a_{l+1} = a_l, b_{l+1} = \lambda_l$ ，转步骤4.

非精确一维搜索方法---Armijo-Goldstein准则

Armijo-Goldstein准则计算 λ_k 的步骤

步骤3 检查避免探索点过小的要求

若 $\varphi(\lambda_l) \geq \varphi(0) + (1 - \rho) \lambda_l \varphi'(0)$ ，令 $\lambda_k = \lambda_l$ ，算法停止。

否则，令 $a_{l+1} = \lambda_l, b_{l+1} = b_l$ ，若 $b_{l+1} < +\infty$ ，转步骤4.

否则，令 $\lambda_{l+1} = \alpha \lambda_l, l = l + 1$ ，转步骤2.

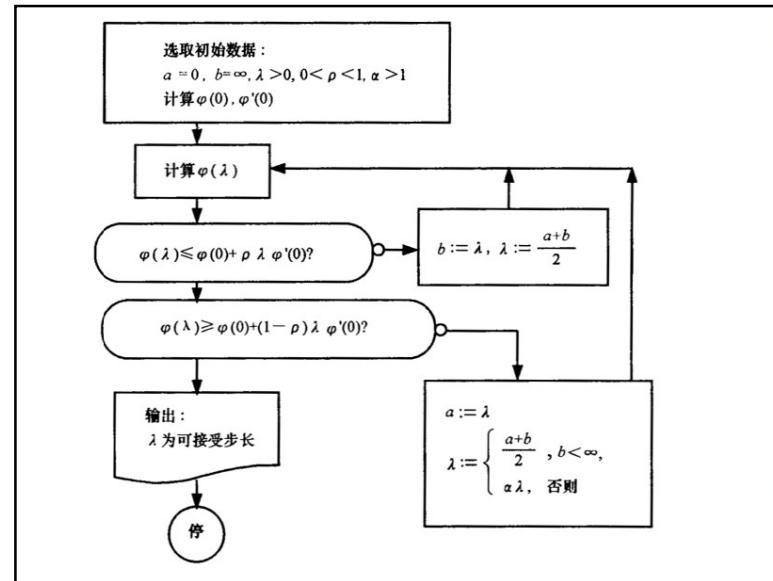
步骤4 选取新的试探点

取 $\lambda_{l+1} = \frac{a_{l+1} + b_{l+1}}{2}, l = l + 1$ ，转步骤2.

79

80

20



81

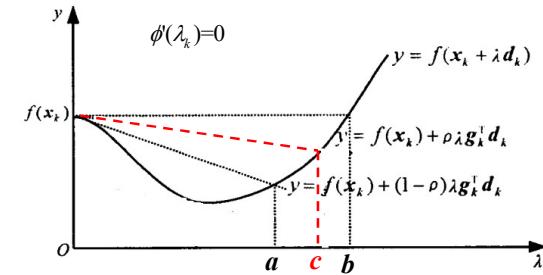
非精确一维搜索方法---Armijo-Goldstein准则

A-G 方法会出现什么问题？

把最佳步长排除在可接受区间内

改进：可接受区间包含最佳步长 $\phi(\lambda_k) \geq \sigma g_k^T d^k (g_k^T d^k < 0)$

$$\lambda_k = \arg \min f(x^k + \lambda d^k) := \phi(\lambda)$$

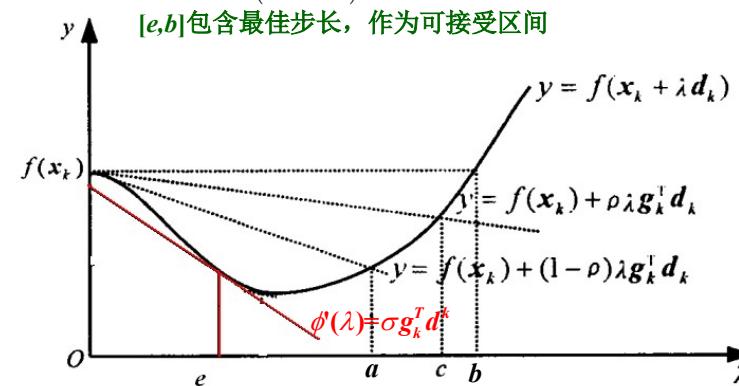


82

非精确一维搜索方法---Wolfe-Powell准则

$$\phi'(\lambda_k) \geq \sigma g_k^T d^k (g_k^T d^k < 0)$$

$[e, b]$ 包含最佳步长，作为可接受区间



83

非精确一维搜索方法---Wolfe-Powell准则

Wolfe-Powell给出了更简单的条件来代替(**)

$$f(x^k + \lambda_k d^k) - f(x^k) \geq \lambda_k (1-\rho) g_k^T d^k \quad (**)$$

$$g_{k+1}^T d^k \geq \sigma g_k^T d^k, \sigma \in (\rho, 1) \quad (***)$$

$$\phi'(\lambda_k) \geq \sigma \nabla f(x^k)^T d^k \quad (****)$$

$$\frac{\nabla f(x^k + \lambda_k d^k)^T d^k}{\nabla f(x^k)^T d^k} \geq \sigma \phi'(0)$$

可接受点处的斜率大于等于初始点处的斜率的 σ 倍

84

21

非精确一维搜索方法---Wolfe-Powell准则

这就是Wolfe-Powell准则，即 λ_k 要满足

$$f(x^k + \lambda_k d^k) - f(x^k) \leq \lambda_k \rho g_k^T d^k \quad (*)$$

和

$$g_{k+1}^T d^k \geq \sigma g_k^T d^k, \sigma \in (\rho, 1) \quad (***)$$

其中 $0 < \rho < 1, \rho < \sigma < 1$.

即 λ_k 要满足

$$\varphi(\lambda_k) \leq \varphi(0) + \rho \lambda_k \varphi'(0)$$

和

$$\varphi'(\lambda_k) \geq \sigma \varphi'(0)$$

85

非精确一维搜索方法---Wolfe-Powell准则

Wolfe-Powell准则计算 λ_k 的步骤

步骤1 选取初始数据 比可接受区间大

在区间 $[0, +\infty)$ 或 $[0, \lambda_{max}]$ 中取定初始点 λ_0 , 计算 $\varphi(0), \varphi'(0)$. 给定 $\rho \in (0, 1), \sigma \in (0, 1), \alpha > 1$.

令 $a_0 = 0, b_0 = +\infty$ 或者 $\lambda_{max}, l = 0$.

步骤2 检查下降量要求

计算 $\varphi(\lambda_l)$, 若 $\varphi(\lambda_l) \leq \varphi(0) + \rho \lambda_l \varphi'(0)$, 转步骤3.

否则, 令 $a_{l+1} = a_l, b_{l+1} = \lambda_l$, 转步骤4.

86

非精确一维搜索方法---Wolfe-Powell准则

Wolfe-Powell准则计算 λ_k 的步骤

步骤3 检查避免探索点过小的要求

计算 $\varphi'(\lambda_l)$, 若 $\varphi'(\lambda_l) \geq \sigma \varphi'(0)$, 令 $\lambda_k = \lambda_l$, 停止.

否则, 令 $a_{l+1} = \lambda_l, b_{l+1} = b_l$, 若 $b_{l+1} < +\infty$, 转步骤4.

否则, 令 $\lambda_{l+1} = \alpha \lambda_l, l = l+1$, 转步骤2.

步骤4 选取新的试探点

$$\text{取 } \lambda_{l+1} = \frac{a_{l+1} + b_{l+1}}{2}, l = l+1, \text{转步骤2.}$$

注: 在Wolfe-Powell准则中, 建议取
 $\rho=0.1, \sigma \in [0.6, 0.8], \alpha=2$.

87

非精确一维搜索方法---简单准则和后退法

实际上, 通常仅用准则(*),

$$f(x^k + \lambda_k d^k) - f(x^k) \leq \lambda_k \rho g_k^T d^k, \quad (*)$$

即 $\varphi(\lambda_k) \leq \varphi(0) + \rho \lambda_k \varphi'(0)$.

把仅用准则(*)的称作**简单准则**.

利用简单准则的非精确一维搜索方法称作**后退法**.

其基本思想是:

开始令 $\lambda=1$, 若 $x^k + \lambda d^k$ 不可接受, 减少 λ (后退),
 一直到 $x^k + \lambda d^k$ 可接受为止.

88

22

非精确一维搜索方法---简单准则和后退法

后退法计算 λ_k 的步骤

步骤1 选取初始数据

给定 $\rho \in (0, \frac{1}{2})$, $0 < l < u < 1$. 令 $\lambda = 1$.

步骤2 若 $f(x^k + \lambda d^k) - f(x^k) \leq \rho \lambda g_k^T d^k$, 则令 $\lambda_k = \lambda$,
停止. 否则, 转步骤3.

步骤3 令 $\lambda = w\lambda$, $w \in [l, u]$, 转步骤2.

第三章作业:

1. 各种方法的例题自己动手计算一下;
2. P65 3.1;
3. 用Matlab编写各种方法的程序, 并用编
写的程序求解各种方法的例题. (选做)