最优化理论与方法

第六讲

无约束优化的下降算法

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

算法 :两个基本策略

- 线搜索法（下降算法）

$$\min_{\alpha>0} f(x_k + \alpha\, p_k)$$

- 信赖域法

$$\min_p m_k(x_k + p) = f_k + p^T \nabla f_k + \frac{1}{2} p^T B_k p$$

$$s.t. \; \| p \| \le \Delta_k$$

$$\min f(x), x \in R^n \tag{2.1}$$

第二章　无约束问题的下降算法与线性搜索

第一节 无约束问题的最优性条件

第二节 下降算法的一般步骤

第三节 线性搜索

我们先来回忆一下一元函数的极值条件(关于极小值点)

	$x \in R$		$x \in R^n (n>1)$
一阶必要条件:	$f'(x^*)=0$	$\Rightarrow$	$\nabla f(x^*)=0$
二阶必要条件:	$\begin{cases} f'(x^*)=0 \\ f''(x^*)\ge 0 \end{cases}$	$\Rightarrow$	$\begin{cases} \nabla f(x^*)=0 \\ \nabla^2 f(x^*)\ \text{半正定} \end{cases}$
二阶充分条件:	$\begin{cases} f'(x^*)=0 \\ f''(x^*)> 0 \end{cases}$	$\Rightarrow$	$\begin{cases} \nabla f(x^*)=0 \\ \nabla^2 f(x^*)\ \text{正定} \end{cases}$

下 降 方 向

考察无约束问题

$$\min f(x), x \in R^n \tag{2.1}$$

这里函数$f: R^n \to R$是连续可微的

定义2.1.1 设$x,d \in R^n$,若存在数$\bar{\alpha} > 0$,使得

$$f(x+\alpha d) < f(x), \forall \alpha \in (0,\bar{\alpha})$$

则称d是函数f在x处的一个下降方向；若$-d$是下降,则称d上升

定理2.1.1 设函数f连续可微且$\nabla f(x) \neq 0$，则

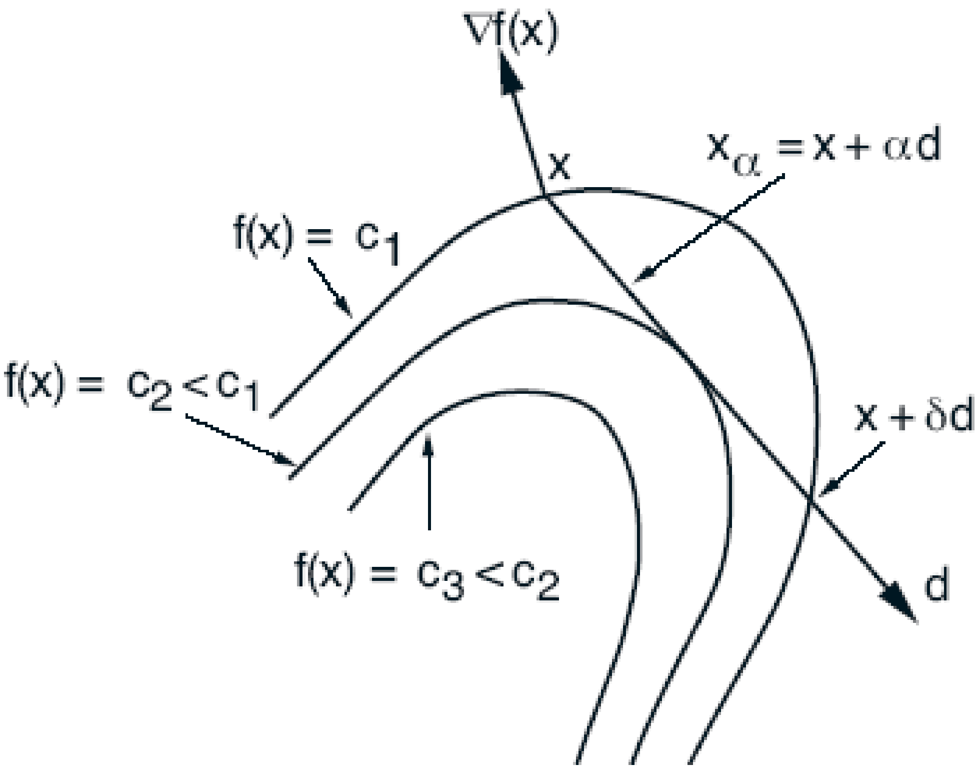
(1) 若向量d满足$\nabla f(x)^{\mathrm{T}} d < 0$,则$d$是$f$在$x$处的一个下降方向

(2) 若n阶矩阵H对称正定,则向量$d = -H\nabla f(x)$是f在x处的一个下降方向.特别,$d = -\nabla f(x)$是f在x处的一个下降方向.

练习：习题2的第5题

下降方向

$\boxed{d \text{ 是函数 } f \text{ 在点 } x \text{ 处的一个下降方向}}$



$$\nabla f(x)' d < 0,$$

下降算法的一般步骤

算法 2.1 *(求解无约束问题的下降算法)*

步 *1* 给定初始点 $x^{(0)} \in R^n$, 精度 $\epsilon > 0$. 令 $k := 0$.

步 *2* 若 $\|\nabla f(x^{(k)})\| \le \epsilon$, 则终止算法, 得解 $x^{(k)}$. 否则, 转步 *3*.

步 *3* 确定下降方向 $d^{(k)}$, 使得

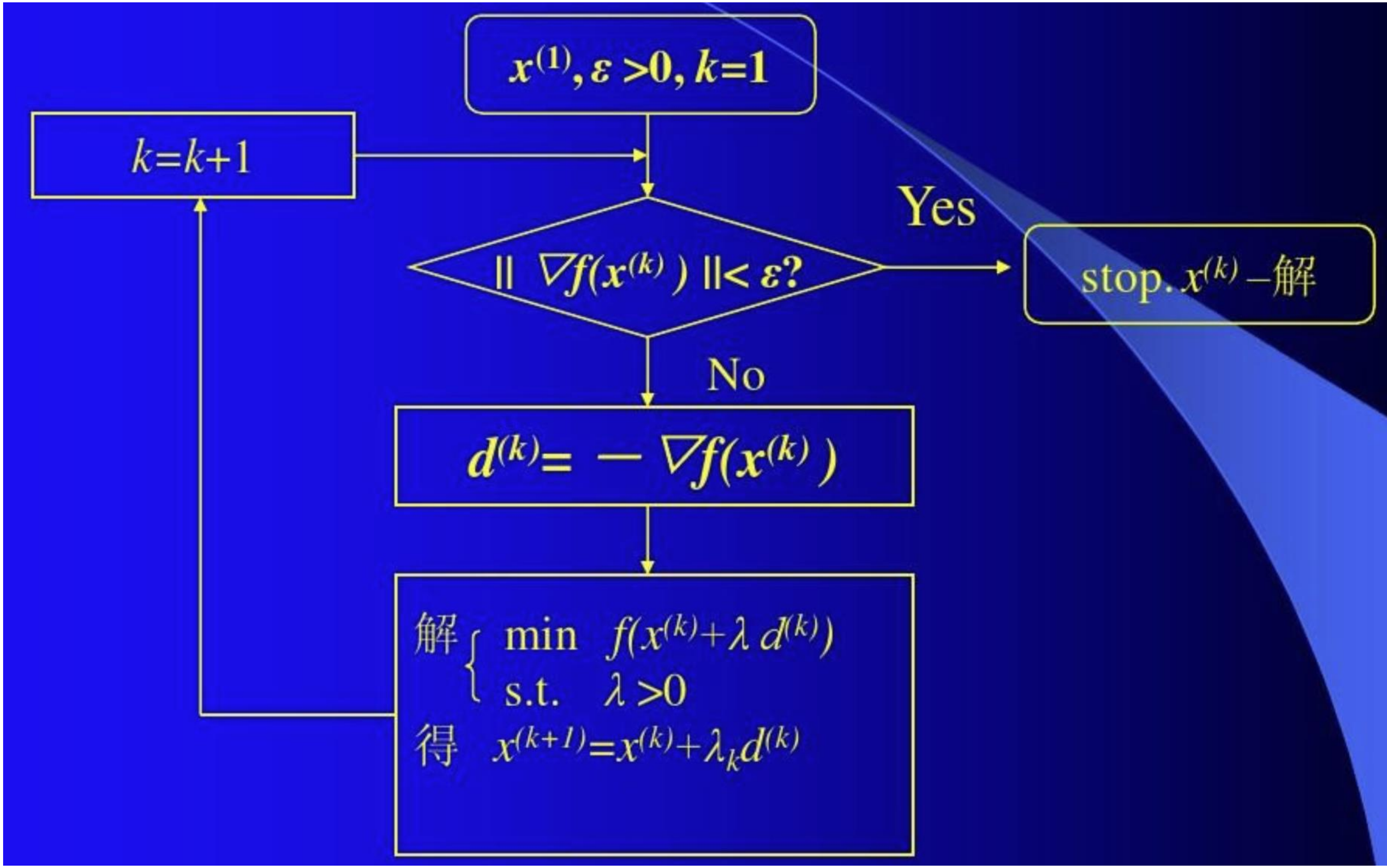
$$\nabla f(x^{(k)})^T d^{(k)} < 0.$$

步 *4* 确定步长 $\alpha_k > 0$ 使得

$$f(x^{(k)} + \alpha_k d^{(k)}) < f(x^{(k)}).$$

步 *5* 令 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$, $k := k + 1$. 转步 *2*.

最速下降算法流程图


$x^{(1)}, \varepsilon > 0, k=1$
$k=k+1$
$\| \nabla f(x^{(k)}) \| < \varepsilon?$
Yes
stop. $x^{(k)}$ —解
No
$d^{(k)} = -\nabla f(x^{(k)})$
解 $\begin{cases} \min & f(x^{(k)}+\lambda d^{(k)}) \\ \text{s.t.} & \lambda > 0 \end{cases}$
得 $x^{(k+1)} = x^{(k)} + \lambda_k d^{(k)}$

由下降算法的结构知,为构造一个使用的下降算法,我们的工作主要有两部分：

已知近似最优解x_k,

1. 计算下降方向d_k满足: $\nabla f(x_k)^T d_k < 0$

基本的构造方法有两个:

(1) $d_k = -H_k \nabla f(x_k)$,其中H_k是某一对称正定矩阵

(2) $d_k = -a_k \nabla f(x_k) + b_k d_{k-1}$, 即$-\nabla f(x_k)$和$d_{k-1}$,的线性组合

d_k的不同构造方式对应不同的最优化算法,具体将在以后各章介绍

2. 计算步长$\alpha_k > 0$满足: $f(x_k + \alpha_k d_k) < f(x_k)$

这一步主要通过线性搜索来完成——一元函数

$$\phi(\alpha) = f(x_k + \alpha\ d_k)$$

求极值,具体在下一节介绍

$$\boxed{x^{k+1} = x^k + \alpha^k d^k, \qquad k = 0, 1, \ldots}$$

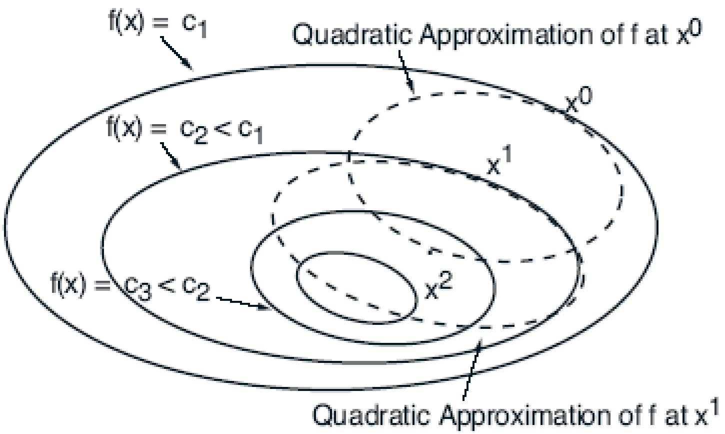
- 最速下降法：

$$\boxed{x^{k+1} = x^k - \alpha^k \nabla f(x^k), \qquad k = 0, 1, \ldots}$$

- Newton法：

$$\boxed{x^{k+1} = x^k - \alpha^k \left(\nabla^2 f(x^k)\right)^{-1} \nabla f(x^k),}$$

最速下降法具有慢的收敛速度



Newton法具有很快的收敛速度（二阶）

步长 α^k 的选取：线性搜索

计算步长α_k的两种线性搜索：精确搜索和非精确搜索：

精确搜索：α_k是一维优化问题的解

$$\min_{\alpha>0} f(x_k+\alpha d_k)$$

即α_k满足

$$\nabla f(x_k+\alpha_k d_k)^{\mathrm{T}} d_k = 0$$

非精确搜索：α_k按照某种规则计算使之满足

$$f(x_k+\alpha_k d_k) < f(x_k)$$

即函数值有一定程度下降.

步长 α^k 的选取

精确线性搜索：α_k是一维优化问题的解

$$\min_{\alpha>0} f(x_k+\alpha d_k)$$

令 $\phi(\alpha)=f(x_k+\alpha d_k)$,则 $\phi'(\alpha)=\nabla f(x_k+\alpha d_k)^{\mathrm{T}} d_k$

由$\phi'(\alpha)=0$得解α_k. 所以计算α_k等价于解方程

$$\nabla f(x_k+\alpha d_k)^{\mathrm{T}} d_k=0 \tag{100}$$

如果方程(100)简单,直接解方程求得α_k;否则需用数值方法求近似解

- 数值方法：
  - 试探法：
    - 二分法
    - 黄金分割法
    - 分数法
    - ……
  - 曲线拟合法：
    - 切线法
    - 抛物线法
    - 有理插值法
    - ……

直接解方程计算步长α_k的例子：

如二次函数

$$f(x)=\frac{1}{2}x^TQx+q^Tx+c, \qquad x\in R^n$$

其中矩阵Q对称且正定

由于

$$\phi(\alpha)=f(x_k+\alpha d_k)=\frac{\alpha^2}{2}d_k^TQd_k+\alpha\nabla f(x_k)^Td_k+f(x_k)$$

则 $$\phi'(\alpha)=\alpha d_k^TQd_k+\nabla f(x_k)^Td_k$$

令 $\phi'(\alpha)=0$, 得解

$$\alpha_k=-\frac{\nabla f(x_k)^Td_k}{d_k^TQd_k}$$

一、 精确线性搜索——黄金分割法（0.618法）

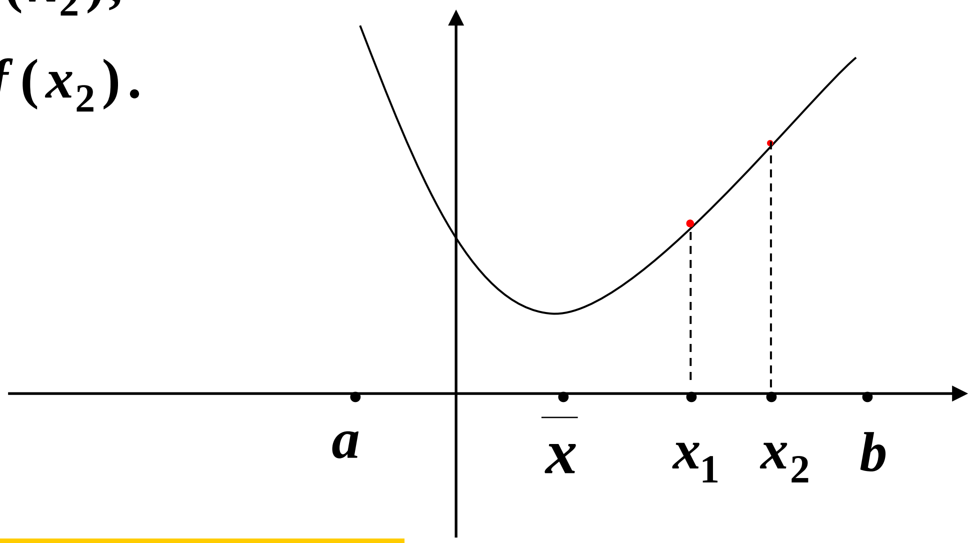
1. 单峰函数

定义：设 $f(x)$ 是区间 $[a,b]$ 上的一元函数，$\bar{x}$ 是 $f(x)$ 在 $[a,b]$ 上的极小点，且对任意的 $x_1, x_2 \in [a,b], x_1 < x_2$，有

（a）当 $x_2 \le \bar{x}$ 时，$f(x_1) > f(x_2)$;

（b）当 $x_1 \ge \bar{x}$ 时，$f(x_1) < f(x_2)$.

则称 $f(x)$ 是单峰函数。



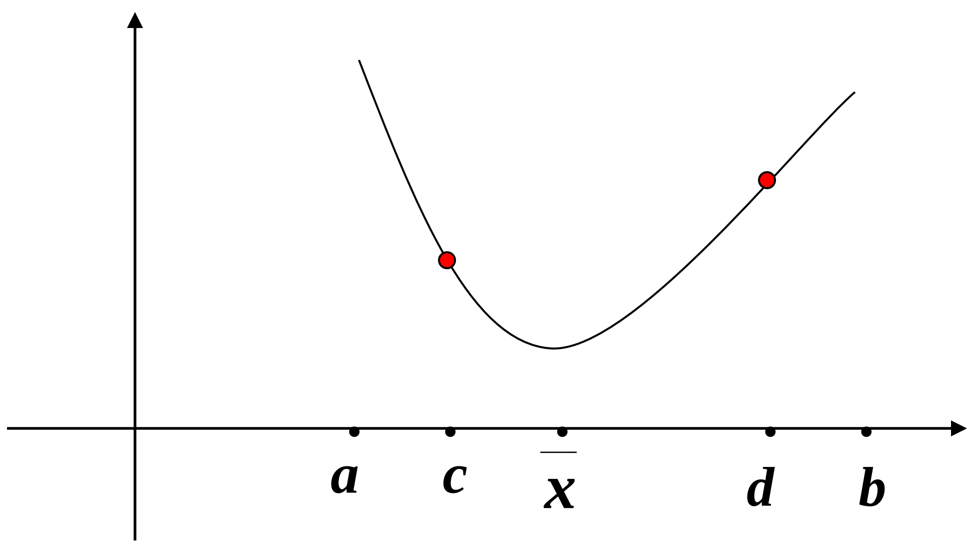
思考：根据 $f(x_1)$ 和 $f(x_2)$ 的大小关系，能否判定最小值点 $\bar{x}$ 落在哪个区间？

性质：通过计算区间 $[a,b]$ 内两个不同点的函数值，就可以确定一个包含极小点的子区间。

定理 设 $f(x)$ 是区间 $[a,b]$ 上的单峰函数，$\bar{x}$ 是 $f(x)$ 在 $[a,b]$ 上的极小点。任取点 $c<d\in[a,b]$，则有

（1）如果 $f(c)>f(d)$，则 $\bar{x}\in[c,b]$;

（2）如果 $f(c)\le f(d)$，则 $\bar{x}\in[a,d]$。

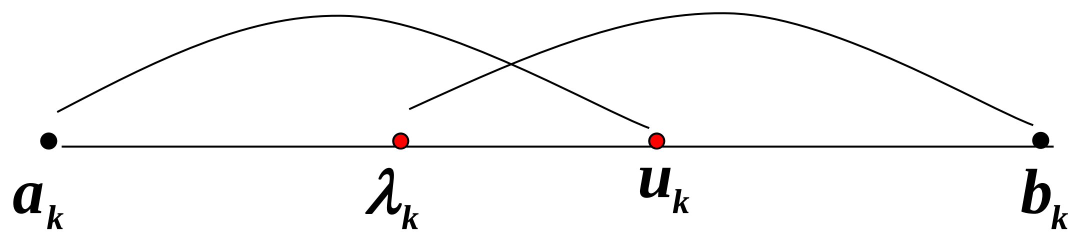


2. 黄金分割法

思想：通过选取试探点使包含极小点的区间按相同比例不断缩短，直到区间长度小到一定程度，此时区间上各点的函数值均接近极小值。

下面推导黄金分割法的计算公式

设 $f(x)$ 在 $[a_1, b_1]$ 上单峰，极小点 $\bar{x} \in [a_1, b_1]$.



第 k 次迭代前 $\bar{x} \in [a_k, b_k]$，取 $\lambda_k, \mu_k \in [a_k, b_k]$，规定 $\lambda_k < \mu_k$.

计算 $f(\lambda_k)$ 和 $f(\mu_k)$，分两种情况：

1. 若 $f(\lambda_k) > f(\mu_k)$，则令 $a_{k+1} = \lambda_k$，$b_{k+1} = b_k$；

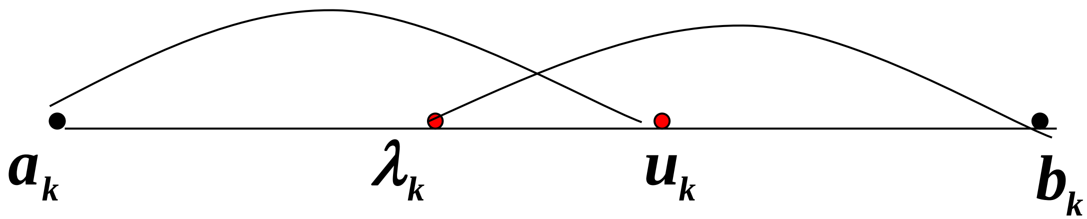
2. 若 $f(\lambda_k) \le f(\mu_k)$，则令 $a_{k+1} = a_k$，$b_{k+1} = \mu_k$.

如何确定 λ_k 与 μ_k?

要求其满足以下两个条件：

1. λ_k和μ_k到$[a_k, b_k]$端点等距，即

$$b_k - \lambda_k = \mu_k - a_k \qquad \textbf{(1)}$$



2. 每次迭代区间长度缩短 比率相同，即

$$b_{k+1} - a_{k+1} = \alpha(b_k - a_k) \qquad (\alpha > 0\text{为某一缩小比例}) \qquad \textbf{(2)}$$

$$\alpha = \frac{\sqrt{5}-1}{2} \approx 0.618 \qquad \text{黄金分割比例} \qquad \textbf{(3)}$$

(4)

黄金分割算法步骤：

1. 给定初始区间 $[a_1, b_1]$，精度要求 $\varepsilon > 0.$

令 $\lambda_1 = a_1 + 0.382\,(b_1 - a_1),\ \mu_1 = a_1 + 0.618\,(b_1 - a_1),$

并计算 $f(\lambda_1)$ 与 $f(\mu_1)$. 令 $k := 1.$

2. 若 $b_k - a_k < \varepsilon$，停止，且 $\bar{x} = \dfrac{b_k + a_k}{2}$. 否则，

当 $f(\lambda_k) > f(\mu_k)$ 时，转 3; 当 $f(\lambda_k) \le f(\mu_k)$ 时，转 4.

3. 令 $a_{k+1} = \lambda_k,\ b_{k+1} = b_k,\ \ \lambda_{k+1} = \mu_k,$

$\mu_{k+1} = a_{k+1} + 0.618(b_{k+1} - a_{k+1}),$ 计算 $f(\mu_{k+1})$, 令 $k := k+1,$ 转 2。

4. 令 $a_{k+1} = a_k,\ b_{k+1} = \mu_k,\ \mu_{k+1} = \lambda_k,$

$\lambda_{k+1} = a_{k+1} + 0.382(b_{k+1} - a_{k+1}),$ 计算 $f(\lambda_{k+1})$，令 $k := k+1,$ 转 2。

黄金分割法的迭代效果：第k次迭代后所得区间长度为初始区间长度的 $\left(\frac{\sqrt{5}-1}{2}\right)^k$ 倍。

黄金分割法的特点：实现比较简单,且不必预先知道探索点的个数

Fibonacci法

- 类似黄金分割法,都是分割方法,主要区别之一在于:搜索区间长度的缩短率不是采用黄金分割数,而是Fibonacci数。

二、 非精确线性搜索— Armijo 型线性搜索 和 Wolfe-Powell 型线性搜索

精确搜索计算量较大，非精确搜索计算量小，易于实现，其计算步长α_k使得$f(x_k+\alpha_k d_k)$较$f(x_k)$有一定下降量.

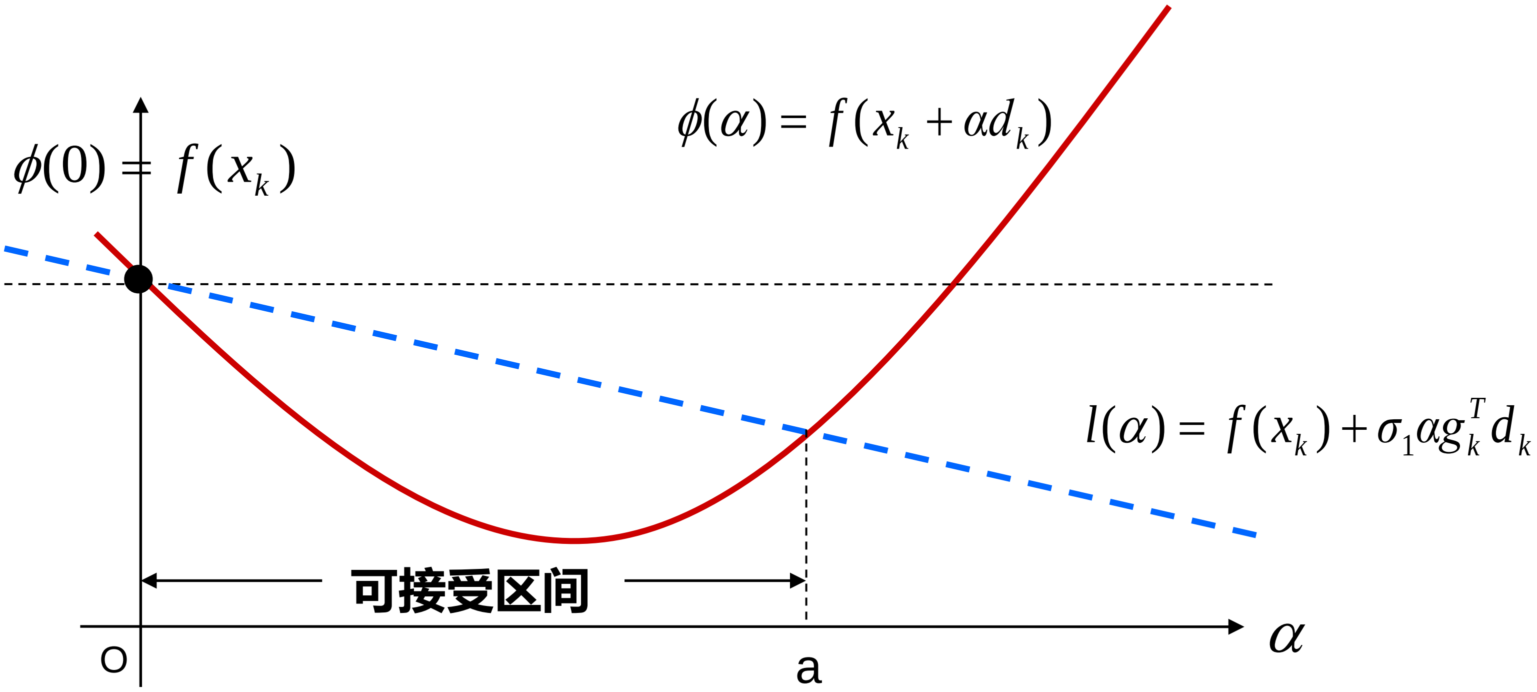
非精确搜索中最简单的是Armijo搜索

Armijo型线性搜索：给定$\sigma_1 \in (0,1)$,计算α_k满足

$$f(x_k+\alpha_k d_k) \le f(x_k)+\sigma_1 \alpha_k \nabla f(x_k)^{\mathrm{T}} d_k \qquad (2.6)$$

令$\phi(\alpha)=f(x_k+\alpha d_k)$,(2.6)等价于

$$\phi(\alpha_{\mathrm{k}}) \le \phi(0)+\sigma_1 \alpha_k \phi^{'}(0)$$



用进退法实现Armijo搜索：逐渐变小以满足Armijo搜索条件

算法2.3 (Armijo 型线性搜索)

步0 若$\alpha_k = 1$满足(2.6), 则取$\alpha_k = 1$.否则转下一步;

步1 给定常数$\beta > 0, \rho \in (0,1)$.令$\alpha_k = \beta$;

步2 若α_k满足(2.6), 则得到步长α_k, 终止计算.否则转步3;

步3 令$\alpha_k := \rho\alpha_k$, 转步2.

注意: 单位步长$\alpha_{\mathrm{k}} = 1$很重要, 它能使算法获得较快的收敛速度.

例2.3.2 给定无约束问题

$$\min\ f(x)=\frac{1}{2}x_1^2+x_2^2$$

设$x^{(0)}=(1,1)^T, d^{(0)}=(1,-1)^T$.用Armijo 搜索计算$\alpha_0=0.5^i$,使得

$$f(x^{(0)}+\alpha_0 d^{(0)})\le f(x^{(0)})+0.9\alpha_0\nabla f(x^{(0)})^{\mathrm{T}}d^{(0)}$$

解: $\nabla f(x)=(x_1, 2x_2)^{\mathrm{T}}, \nabla f(x^{(0)})=(1,2)^T, \nabla f(x^{(0)})^{\mathrm{T}}d^{(0)}=-1<0,$

故$d^{(0)}$是f在$x^{(0)}$处的一个下降方向.

由于 $x^{(0)}+\alpha d^{(0)}=(1+\alpha,\ 1-\alpha)^{\mathrm{T}},$

故 $f(x^{(0)}+\alpha d^{(0)})-f(x^{(0)})=\frac{1}{2}(1+\alpha)^2+(1-\alpha)^2=\frac{3}{2}\alpha^2-\alpha$

Armijo 搜索条件可以写成:

$$\frac{3}{2}\alpha^2-\alpha\le 0.9\alpha\times(-1)=-0.9\alpha$$

即 $\alpha\le\frac{1}{15}\approx 0.066666.$ 由$\alpha_0=0.5^{\mathrm{i}}\le\frac{1}{15}$,即得$\alpha_0=0.5^4=0.0625$

注：(i) Armijo 型线性搜索 (2.6) 中的参数 σ_1 可取为 $(0,1)$ 中的任何实数. 但当 $\sigma_1 \in (0,1/2)$ 时，可保证 Newton 法和拟 Newton 法的超线性收敛性.

(ii) 步长 $\alpha_k = 1$ 是很重要的步长. 它在算法的收敛速度分析中起到十分重要的作用.

Armijo型线性搜索的缺点：步长可能过小

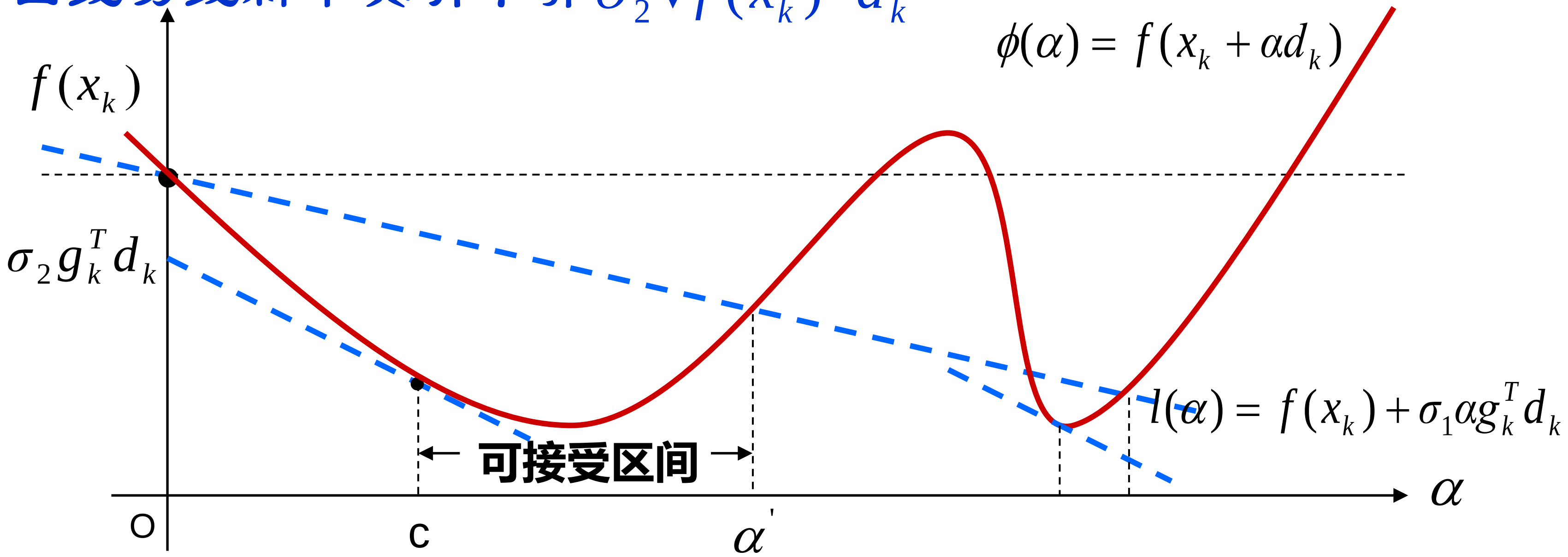
在上面的 Armijo 型线性搜索中，试探步按比例 ρ 缩小. 若 $\rho \in (0,1)$ 较大 (如 ρ 接近于 1)，则相邻两次试探步的改变相对较小. 此时，需要经过较多次搜索才能得到 α_k. 若 $\rho \in (0,1)$ 较小 (如 ρ 接近于 0)，则相邻两次试探步的改变相对较大. 此时，可经过相对较少的试探步得到 α_k. 但获得的步长 α_k 可能很小. 为了克服 Armijo 型线性搜索的这一缺陷，可采用下面的 Wolfe-Powell 型非精确线性搜索.

在上面关于Armijo 搜索中，参数ρ的选择非常重要，如果过大，则函数计算量很大；过小则产生过小的步长，

为克服Armijo 搜索的这一缺陷，人们在Armijo 搜索的下降条件基础上增加防止步长过小的条件：

$$\phi'(\alpha_k) = \nabla f(x_k + \alpha_k d_k)^T d_k \geq \sigma_2 \nabla f(x_k)^T d_k$$

即曲线切线斜率具有下界$\sigma_2 \nabla f(x_k)^T d_k$



从而得到了另一个重要的非精确搜索－Wolfe - Powell搜索。

Wolfe - Powell线性搜索：给定常数 $0<\sigma_1<\sigma_2<1$

$$\begin{cases} f(x_k+\alpha_k d_k)\le f(x_k)+\sigma_1\alpha_k\nabla f(x_k)^T d_k \\ \nabla f(x_k+\alpha_k d_k)^{\mathrm{T}} d_k \ge \sigma_2\nabla f(x_k)^T d_k \end{cases} \tag{2.7}$$

为保证α_k的存在性，通常取$0<\sigma_1<\dfrac{1}{2}$

令$\phi(\alpha)=f(x_k+\alpha d_k)$，上面的搜索条件等价于

$$\begin{cases} \phi(\alpha_k)\le\phi(0)+\sigma_1\alpha_k\phi'(0) \\ \phi'(\alpha_k)\ge\sigma_2\phi'(0) \end{cases}$$

关于Wolfe - Powell搜索中α_k的存在性，

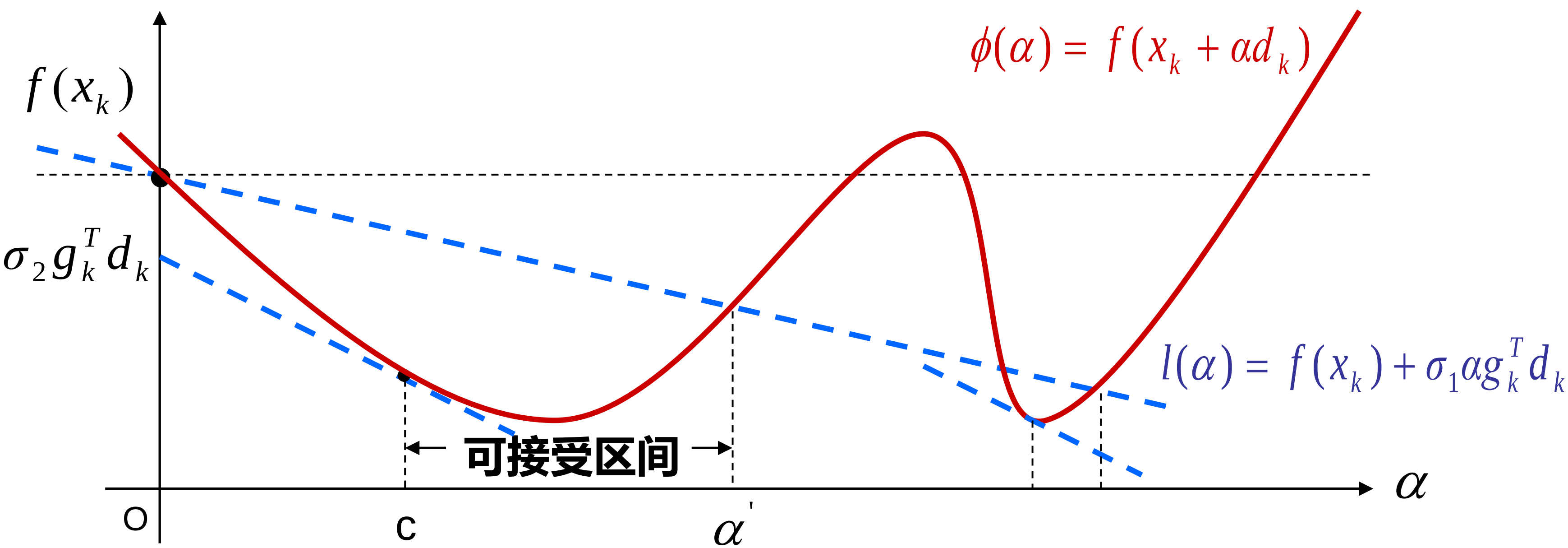
我们有下面结论

定理 设$f: R^n \to R$是连续可微的，且有下界，如果$\nabla f(x_k)^{\mathrm{T}} d_k < 0$，则存在一个区间$[a,b]$，使得$\forall \alpha \in [a,b]$满足(2.7)成立。

证明：因为曲线$\phi(\alpha) = f(x_k + \alpha d_k)$有下界，而直线

$$l(\alpha) = f(x_k) + \alpha \sigma_1 \nabla f(x_k)^{\mathrm{T}} d_k$$

单调下降，两者必定至少相交一次，



令$\alpha' > 0$是其中最小的交点，则有

$$f(x_k + \alpha' d_k) = f(x_k) + \alpha' \sigma_1 \nabla f(x_k)^{\mathrm{T}} d_k \qquad (1)$$

而对任意的$\alpha \in (0, \alpha')$, 有

$$f(x_k + \alpha d_k) \le f(x_k) + \alpha \sigma_1 \nabla f(x_k)^{\mathrm{T}} d_k$$

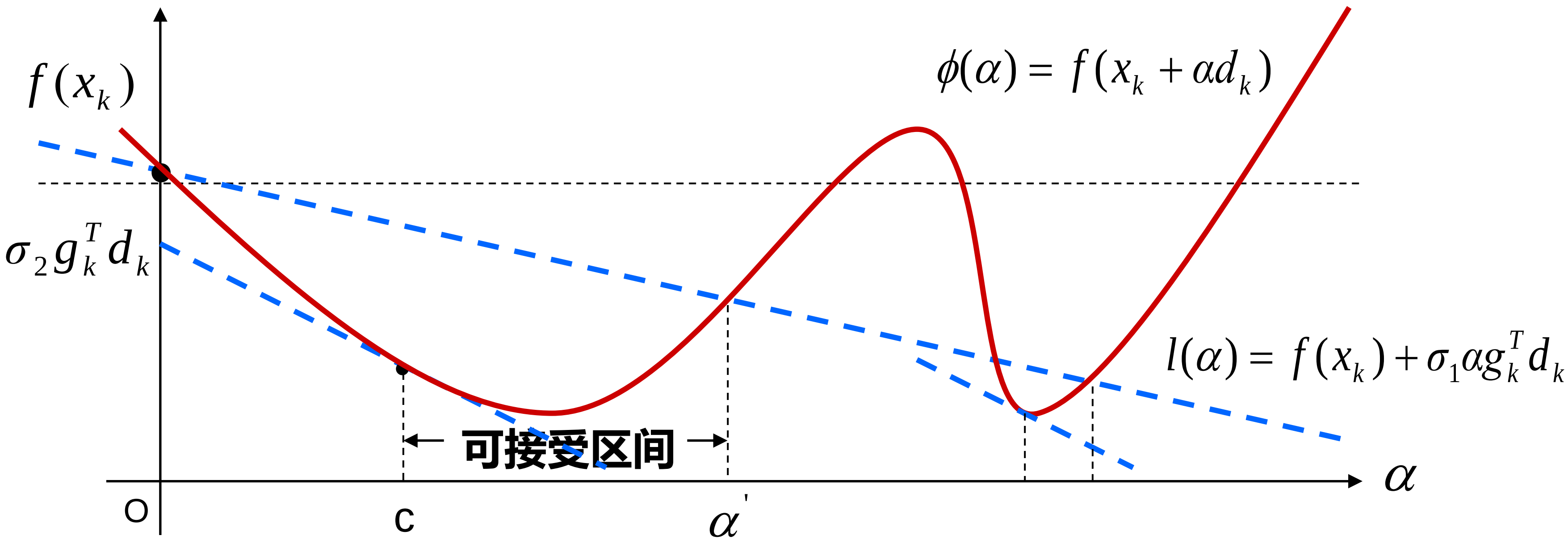
由中值定理，存在$\alpha'' \in (0, \alpha')$满足

$$f(x_k + \alpha' d_k) - f(x_k) = \alpha' \nabla f(x_k + \alpha'' d_k)^{\mathrm{T}} d_k \qquad (2)$$

由（1）,(2)得

$$\nabla f(x_k + \alpha'' d_k)^{\mathrm{T}} d_k = \sigma_1 \nabla f(x_k)^{\mathrm{T}} d_k > \sigma_2 \nabla f(x_k)^{\mathrm{T}} d_k$$

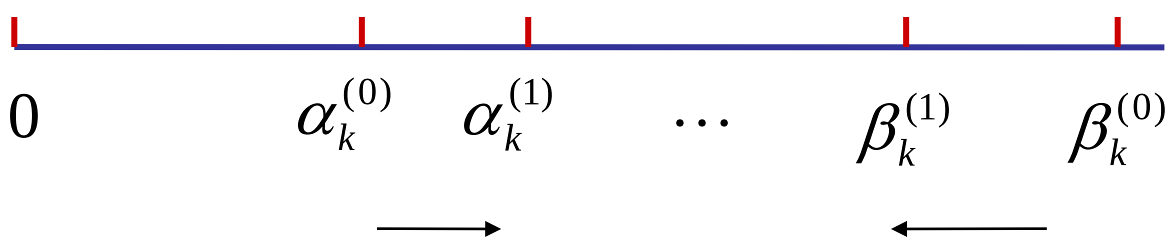
即α''满足Wolfe - Powell条件，由于f是连续可微的，故必存在一个小的区间$\alpha \in [a, b]$满足Wolfe - Powell条件(2.7).



用进退法实现Wolfe-Powell搜索：

逐渐变小以满足搜索条件的第一个条件

逐渐增大以满足搜索条件的第二个条件



具体实现如下：

步0: 若$\alpha_k = 1$满足(2.7), 取若$\alpha_k = 1$; 否则转下一步；

步2: 给定常数$\beta > 0, \rho, \rho_1 \in (0,1)$. 令$\alpha_k^{(0)}$是集合$\{\beta\rho^j \mid (j$可正可负$)\}$

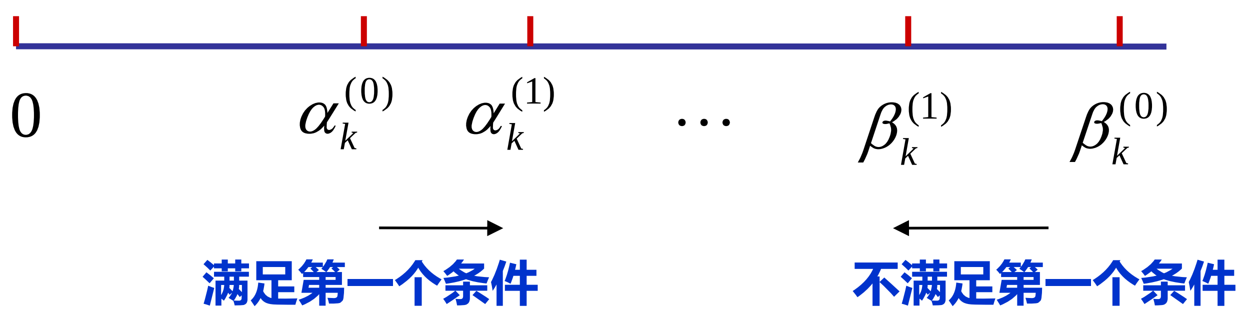
中满足第一个不等式的最大者. 令$i = 0$;

步2: 若$\alpha_k^{(i)}$满足第二个条件， 则取$\alpha_k = \alpha_k^{(i)}$; 否则, 令$\beta_k^{(\mathrm{i})} = \rho^{-1}\alpha_k^{(i)}$.

步3: 令$\alpha_k^{(i+1)}$是集合

$$\{\alpha_k^{(i)} + \rho_1^{\mathrm{j}}(\beta_k^{(\mathrm{i})} - \alpha_k^{(i)}),\ j = 0,1,2,\cdots\}$$

中使得(2.7)第一个不等式成立的最大者， 令$i = i + 1$, 转步2.



进退法计算Wolfe-Powell搜索的计算量比较大，但比较稳定，另一种常用的方法是二点插值法，计算简单，但不太稳定.(感兴趣的同学，详看袁亚湘老师书:75-93页)

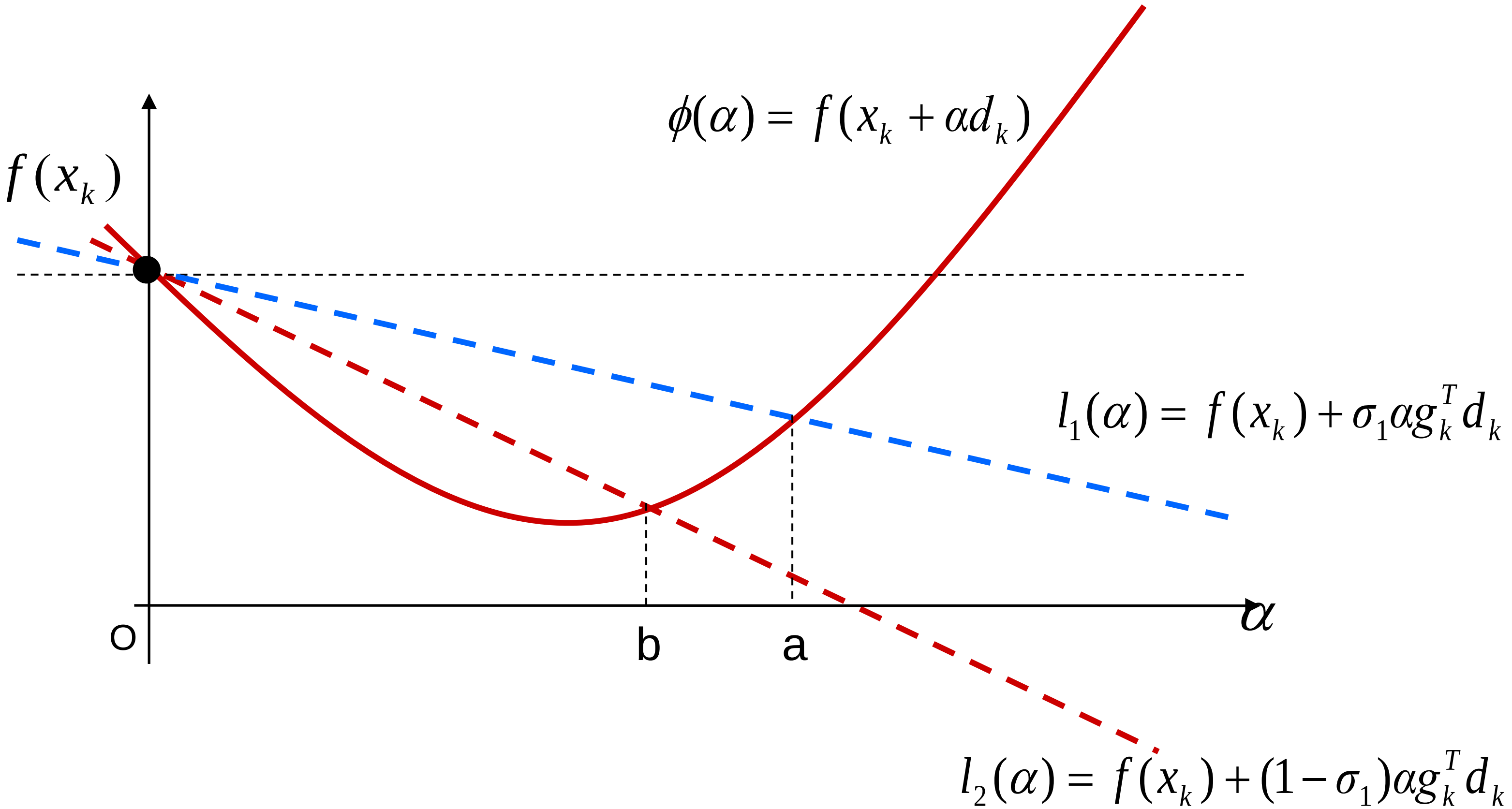
已知函数φ，及两点α_1, α_2

插值法之一：已知两点α_1, α_2及函数值$\varphi(\alpha_1), \varphi(\alpha_2)$, 导数$\varphi'(\alpha_1)$

$$\alpha_{k+1} = \alpha_k - \frac{(\alpha_k - \alpha_{k-1})\varphi_k'}{2\left[\varphi_k' - \dfrac{\varphi_k - \varphi_{k-1}}{\alpha_k - \alpha_{k-1}}\right]}$$

插值法之二：已知两点α_1, α_2及函数值$\varphi(\alpha_1)$, 导数$\varphi'(\alpha_1)$，$\varphi'(\alpha_2)$

$$\alpha_{k+1} = \alpha_k - \frac{(\alpha_k - \alpha_{k-1})\varphi_k'}{\varphi_k' - \varphi_{k-1}'}$$



第三种重要的线性搜索：Goldstein搜索

$$f(x_k + \alpha d_k) \le f(x_k) + \sigma_1 \alpha g_k^T d_k, \qquad 0 < \sigma_1 < \frac{1}{2}$$

$$f(x_k + \alpha d_k) \ge f(x_k) + (1-\sigma_1)\alpha g_k^T d_k$$

上机1

- 请用Matlab编程，实现黄金分割方法，采用这种方法，计算习题2的第1题

  $f(x)=(x_1-x_2)^2+x_2^2$在点$x=(0,1)^T$处沿着方向

  $d=-(1,1)^T$的精确搜索步长。

- 再分别用Armijo型和Wolfe-Powell型线性搜索法确定上面的步长.**请编写Matlab程序实现算法，分析参数对算法结果的影响.**
- 比较上面3种方法，书写实验报告（含程序，计算结果的比较：迭代次数，最优目标值，总结和体会），本周日21：00前交。

课后：预习下降算法的收敛性和收敛速度定理的证明。

思 考

- 二分法？如何采用二分法计算步长 α_k ？
- 能否用Matlab编程，实现二分法，或Fibonacci法？

作业：习题2：5