运筹学与优化方法

晁国清
计算机科学与技术学院

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

课程回顾

- 无约束优化问题的最优性条件

若$f(x)$为一般函数且$f(x)$在x^*处一阶可微，则

$$x^*\text{是最优解}\Rightarrow \nabla f(x^*)=0$$

若$f(x)$为一般函数且$f(x)$在x^*处二阶可微，则

$$x^*\text{是最优解}\Rightarrow \nabla^2 f(x^*)\succcurlyeq 0$$

若$f(x)$为一般函数且$f(x)$在x^*处二阶可微

$$\nabla f(x^*)=0\text{且}\nabla^2 f(x^*)\succ 0\Rightarrow x^*\text{是最优解}$$

- 迭代算法的基本思想

给定初始点x^0,产生点列$\{x^k\}_{k=1}^{\infty}$,
并且点列满足条件
$f(x^{k+1})<f(x^k)$

Step 1：给定初始点x_0，k=0；

Step 2:判断x^k是否满足终止条件；是，结束；不是，进行Step 3；

Step 3：寻找x^k处的下降方向d^k；

Step 4:选择合适的步长$\alpha_k>0$，使$f(x^k+\alpha_k d^k)<f(x^k)$成立；

Step 5:令$x^{k+1}=x^k+\alpha_k d^k$ 且k=k+1；转Step 2

- 一维(线性)搜索

精确搜索
- 区间试探法
  - 进退探索法
  - 黄金分割法
- 函数逼近法
  - 牛顿法
  - 抛物线法

课程内容

- 非精确的线搜索
- 变量轮换法
- 最速下降法

非精确线搜索

➢Armijo-Goldstein准则：定义$\phi(\alpha) = f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k)$, 则有

$\phi(0) + (1-\rho)\alpha\phi'(0) \le \phi(\alpha) \le \phi(0) + \rho\alpha\phi'(0), \rho \in (0,0.5)$ （＊）

假设$\boldsymbol{d}_k$是下降方向，对$f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k)$在$\boldsymbol{x}_k$处一阶泰勒展开：

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k) = f(\boldsymbol{x}_k) + \alpha \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k + O(\|\alpha \boldsymbol{d}_k\|)$$

已知$\boldsymbol{d}_k$是下降方向，则有$\nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k < 0$, 为保证下降，找$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k)$的一个合理下界：

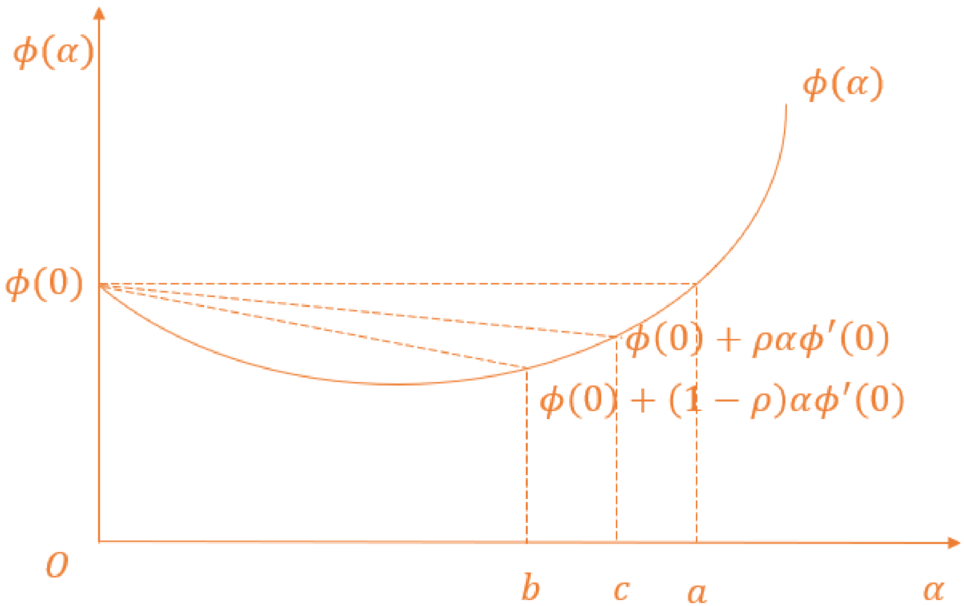
$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k) \ge -\rho\alpha \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k, \rho \in (0, 0.5) \quad (1)$$

再给其一个上界(α不能太小)

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k) \le -(1-\rho)\alpha \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k, \rho \in (0, 0.5) \quad (2)$$

将(1)和(2)写在一起，并定义$\phi(\alpha) = f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k)$, 就得到(*)

Armijo-Goldstein准则的几何意义



从此图可以看到，$\phi(0)+\rho\alpha\phi'(0)$在$\phi(0)$下方，在$\phi(0)+(1-\rho)\alpha\phi'(0)$的上方，对应区间$[b,c]$就是满足Armijo-Goldstein准则的步长

缺点： Armijo-Goldstein准则可能会把极值点排除在可接受的区间之外

Wolfe-Powell准则

- Wolf-Powell准则是为了解决Armijo-Goldstein准则可能会把极值点排除在可接受的区间之外的问题提出来的
- Wolfe-Powell准则的上界与Armijo-Goldstein准则是一样的，即

$$f(\boldsymbol{x}_k+\alpha \boldsymbol{d}_k)-f(\boldsymbol{x}_k)\le \rho\alpha\nabla f(\boldsymbol{x}_k)^T\boldsymbol{d}_k, \rho\in(0,0.5)$$

为了保证步长足够大，并且可接受的区间包含极值点，上界被定义为：

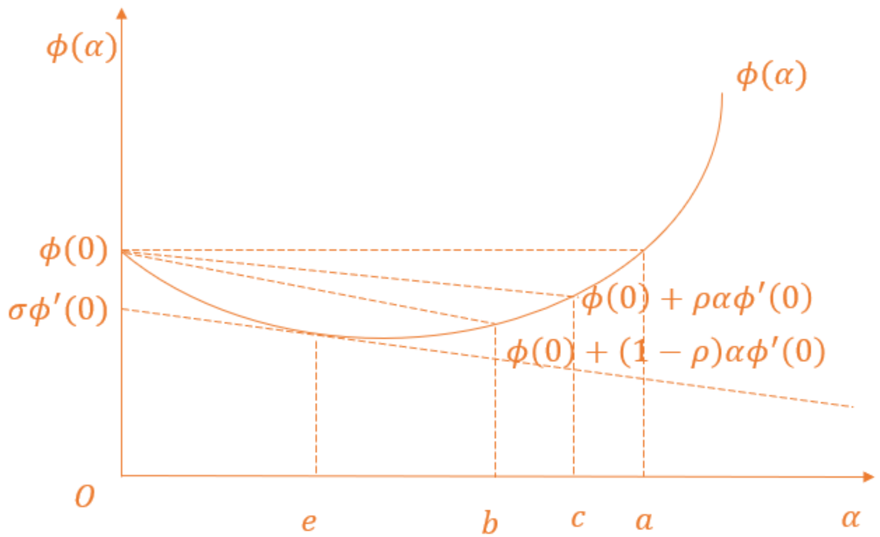
$$\nabla f(\boldsymbol{x}_{k+1})^T\boldsymbol{d}_k\ge \sigma\nabla f(\boldsymbol{x}_k)^T\boldsymbol{d}_k, \sigma\in(\rho,1)$$

即$\phi'(\alpha)\ge\sigma\phi'(0)$, 说明在$\alpha$处的斜率应该大于起始点斜率的$\sigma$倍。$\phi'(0)$是负值，所以上界的含义就是可接受范围中斜率接近于零的负值或者正值。

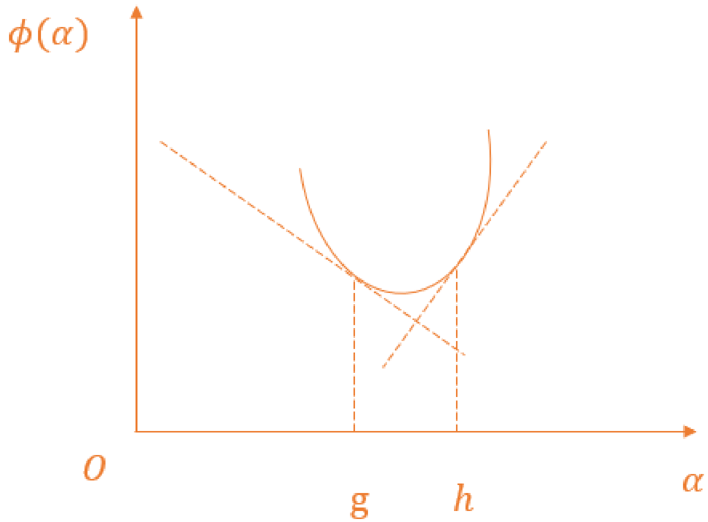
强Wolf条件：$|\phi'(\alpha)|\le-\sigma\phi'(0)$

强Wolf条件使可接受范围的斜率的正值不至于过大，从而将远离极值点的步长排除在外。一般σ越小，线搜索越精确，但同时工作量也越大，一般取$\rho=0.1, \sigma\in[0.6,0.8]$

Wolfe-Powell准则的几何意义



强Wolfe-Powell准则的意义



收敛速度问题

➢对于一个迭代算法，不仅要求其是收敛的，还要考虑收敛速度问题，一般用阶衡量收敛速度。

线性收敛：设$\boldsymbol{x}^*$为$\min f(\boldsymbol{x}), \boldsymbol{x} \in R^n$的最优解，由某算法产生的点序列$\{\boldsymbol{x}^k\}$收敛于$\boldsymbol{x}^*$，即

$$\lim_{k\to\infty} \boldsymbol{x}^k = \boldsymbol{x}^*$$

若存在一个与k无关的常数$\beta \in (0,1)$，对某正数k_0，当$k > k_0$时若有

$$\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\right\| \le \beta\left\|\boldsymbol{x}^k - \boldsymbol{x}^*\right\|$$

则称序列$\{\boldsymbol{x}^k\}$为线性收敛，也称该算法线性收敛。

α阶收敛：设算法产生序列$\{\boldsymbol{x}^k\}$收敛于$\boldsymbol{x}^*$，若存在一个与k无关的常数$\beta \in (0,1)$及$\alpha > 1$，对某正数k_0，当$k > k_0$时若有

$$\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\right\| \le \beta\left\|\boldsymbol{x}^k - \boldsymbol{x}^*\right\|^{\alpha}$$

则称序列$\{\boldsymbol{x}^k\}$为α阶收敛，也称该算法是α阶收敛。

说明：当$\alpha = 1$时， α阶收敛就是线性收敛。当$1 < \alpha < 2$时, 称算法是超线性收敛，当$\alpha = 2$时, 称算法是二阶收敛。阶数越高，收敛速度越快

构造搜索方向问题

构造搜索方向法
- 直接搜索法　变量轮换法
- 解析法
  - 最速下降法
  - 牛顿法
  - 共轭梯度法

➢直接搜索法：在计算过程中只用到目标函数值，不用计算导数

➢解析法：计算过程中要用到目标函数的导数计算

变量轮换法(坐标轴交替下降法)

- 变量轮换法：把多变量函数的优化问题转化为一系列单变量函数的优化问题求解

- 基本思路：搜索方向是各坐标轴的方向，轮流按各坐标轴方向搜索最优点。即从初始点出发，按第i个坐标轴方向搜索时，在n个变量中，只有x_i在变化，其余$n-1$个变量保持不变，依次从x_1到x_n做n次单变量的一维搜索，就完成了变量轮换法的一次迭代

- 设优化问题：

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

记$\boldsymbol{e}_i = (0, \cdots, 1, \cdots, 0)^T \quad (i = 1, 2, \cdots, n)$

即$\boldsymbol{e}_i$是第i个分量为1其余分量为0的单位分量

变量轮换法

➢算法步骤：

Step 1:给定初始点$\boldsymbol{x}^0=(x_1,x_2,\cdots,x_n)^T$, k=0;

Step 2:检查点$\boldsymbol{x}^k$是否满足终止条件，满足则结束。否则转Step 3;

Step 3: 完成变量轮换法的一次迭代

记$\boldsymbol{y}_0=\boldsymbol{x}^k$,

$for\ i=1{:}n$

从$\boldsymbol{y}_{i-1}$出发，沿$\boldsymbol{e}_i$进行线搜索，记求得的最优步长为α_i, 可得到新点$\boldsymbol{y}_i$

$$\boldsymbol{y}_i=\mathrm{y}_{\mathrm{i}-1}+\alpha_i\boldsymbol{e}_i$$

其中$\alpha_i=\arg\min f(\boldsymbol{y}_{i-1}+\alpha\boldsymbol{e}_i)$

Step4:令$\boldsymbol{x}^{k+1}=\boldsymbol{y}_n,\mathrm{k}=\mathrm{k}+1,$ 转至step 2

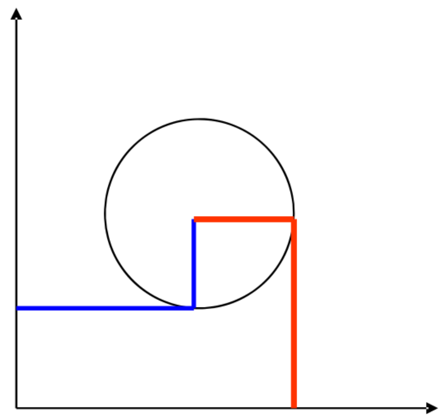
优点：搜索方向的获得不需要花费成本，基本思想简单、容易理解

缺点：收敛速度较慢，搜索效率较低，对于一般问题未必收敛。

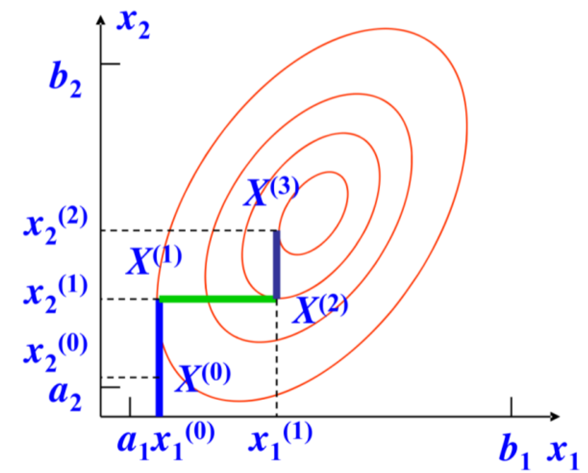
只适用于具有特殊结构的函数

变量轮换法

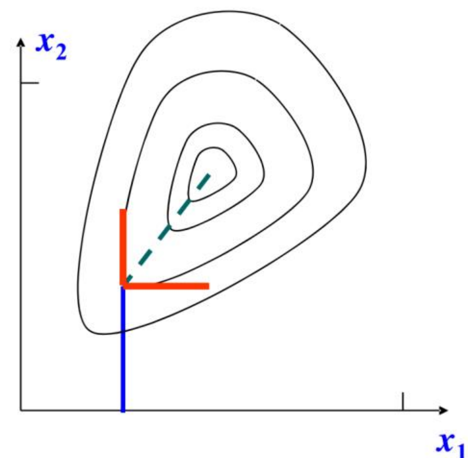
若目标函数的等值线为圆形(二维)、球形(三维)或长短轴平行于坐标轴的椭圆形，搜索最快。变量之间无交互作用



若目标函数的等值线类似于椭圆形，且长短轴不平行于坐标轴，搜索速度较慢。变量之间存在弱交互作用



若目标函数的等值线出现山脊时，该方法无效。变量之间存在强交互作用



最速下降法(梯度法)

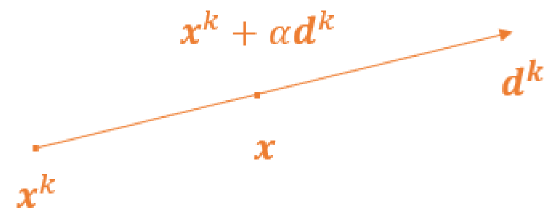
- 考虑无约束优化问题

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

$f(\boldsymbol{x})$具有一阶连续偏导数，有极小点$\boldsymbol{x}^*$

设已求得$\boldsymbol{x}^*$的第k次近似值$\boldsymbol{x}^k$, 为了求得$k+1$次近似值$\boldsymbol{x}^{k+1}$, 需选方向$\boldsymbol{d}^k$。看一下$\boldsymbol{d}^k$应该具备什么特征？

设$\boldsymbol{d}^k$已选定，作射线如图



$$\boldsymbol{x}^k + \alpha \boldsymbol{d}^k = \boldsymbol{x}$$

其中，$\alpha > 0.\ \|\boldsymbol{d}^k\| = 1.$ $\boldsymbol{d}^k$为某个下降方向

现将$f(\boldsymbol{x})$在$\boldsymbol{x}^k$点处作一阶泰勒展开：

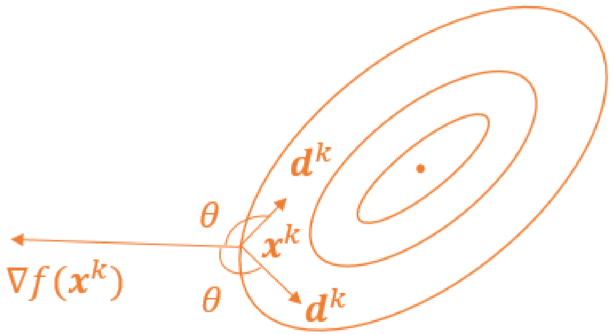
$$f(\boldsymbol{x}) = f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k) = f(\boldsymbol{x}^k) + \alpha \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k + o(\|\alpha \boldsymbol{d}^k\|)$$

有$f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k) - f(\boldsymbol{x}^k) \approx \alpha \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k$, 因为$f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k) - f(\boldsymbol{x}^k) < 0$

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

最速下降法(梯度法)

- 对于$\nabla f\left(\boldsymbol{x}^k\right)^T \boldsymbol{d}^k < 0$，即$\boldsymbol{d}^k$与$\boldsymbol{x}^k$点处梯度$\nabla f(\boldsymbol{x}^k)$的点积小于零，或者说两者之间的夹角应大于90°，如图所示



- 可以看到，满足这种条件的下降方向有很多，如何选择使目标函数值下降最快的方向？

因为$\nabla f\left(\boldsymbol{x}^k\right)^T \boldsymbol{d}^k = \left\|\nabla f(\boldsymbol{x}^k)\right\| \left\|\boldsymbol{d}^k\right\| \cos\theta, \theta$为$\nabla f(\boldsymbol{x}^k)$与$\boldsymbol{d}^k$之间夹角，显然当$\theta = 180°$时，目标函数值$f\left(\boldsymbol{x}^k\right)$在$\boldsymbol{x}^k$点附近下降最快，称为负梯度方向，即

$$\boldsymbol{d}^k = -\nabla f(\boldsymbol{x}^k)$$

为最速下降方向

最速下降法(梯度法)

- 确定搜索方向为负梯度方向后，该方向上的所有点可以表示为

$$\boldsymbol{x}^k + \alpha \boldsymbol{d}^k = \boldsymbol{x}^k - \alpha \nabla f(\boldsymbol{x}^k)$$

同时形成确定步长的优化问题

$$f\left(\boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k\right) = \min_{\alpha} f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k)$$

可以通过所有的线搜索方法求得步长因子α

基本思想：当前点$\boldsymbol{x}^k$处选择负梯度方向$-\nabla f(\boldsymbol{x}^k)$作为搜索方向

最速下降法的步骤：

Step 1：给定初始点$\boldsymbol{x}^0$，终止误差$\epsilon > 0$，k=0；

Step 2：判断是否满足终止条件$\left\|\nabla f(\boldsymbol{x}^k)\right\| < \epsilon$，满足则终止；否则转step 3；

Step 3：构造负梯度方向$d^k = -\nabla f(\boldsymbol{x}^k)$；

Step 4：进行线搜索求得α_k，令$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k = \boldsymbol{x}^k - \alpha_k \nabla f(\boldsymbol{x}^k)$，$k = k + 1$，转Step 2

最速下降法的优缺点

优点：

1. 思想简单，每一步迭代只需计算一阶导数，代价较小
2. 对于well-conditioned, strongly convex问题收敛快

well-conditioned: 函数$\boldsymbol{f(x)}$凸，一阶可微，且$\nabla \boldsymbol{f(x)}$是Lipschitz continuous with constant $\boldsymbol{L > 0}$，即

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, for\ any\ x, y$$

二阶可微函数，则$\nabla^2 f(x) \preccurlyeq L\boldsymbol{I}$

Strongly convex: $f(\boldsymbol{x}) - \frac{m}{2}\|x\|_2^2$ 对于某个$m > 0$是凸函数，如果函数二阶可微，则$\nabla^2 f(x) \succcurlyeq m\boldsymbol{I}$

缺点：

1. 对于不是well-conditioned, strongly convex问题收敛慢(线性收敛)
2. 不能处理一阶不可微的函数
3. 接近极小点时会出现锯齿(Zigzag)现象

锯齿现象的解释

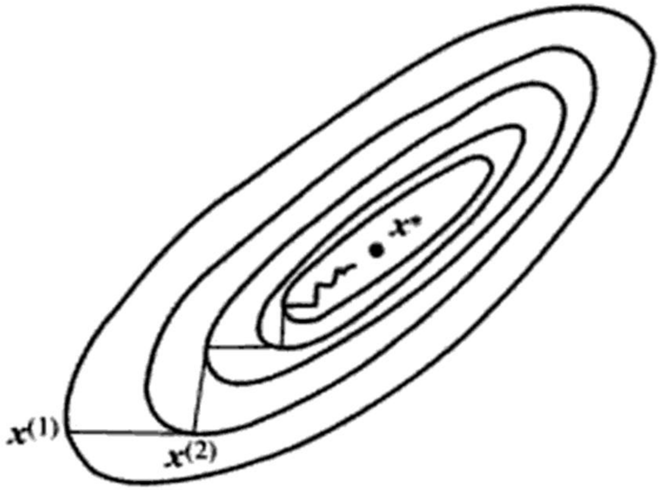
- 用最速下降法极小化目标函数时，相邻两个搜索方向是正交的
- 若迭代中步长α_k是由$\psi(\alpha)=f(\boldsymbol{x}^k+\alpha\boldsymbol{d}^k)$的精确最小点，则由$\psi'(\alpha)=0$, 即

$$\begin{aligned}\psi'(\alpha)&=\nabla f\left(\boldsymbol{x}^k+\alpha\boldsymbol{d}^k\right)^T\boldsymbol{d}^k\\&=-\nabla f\left(\boldsymbol{x}^{k+1}\right)^T\nabla f\left(\boldsymbol{x}^k\right)\\&=0\end{aligned}$$

也就是方向$\boldsymbol{d}^{k+1}=-\nabla f(\boldsymbol{x}^{k+1})$ 与方向$\boldsymbol{d}^k=-\nabla f(\boldsymbol{x}^k)$ 正交。

如此，迭代产生的点列$\{\boldsymbol{x}^k\}$所走的

路径就是“之”字形，如图示



参考文献

- 最优化方法，第4章
- 最优化基础理论与方法，第三章