运筹学通论I

胡晓东

应用数学研究所

中国科学院数学与系统科学研究院

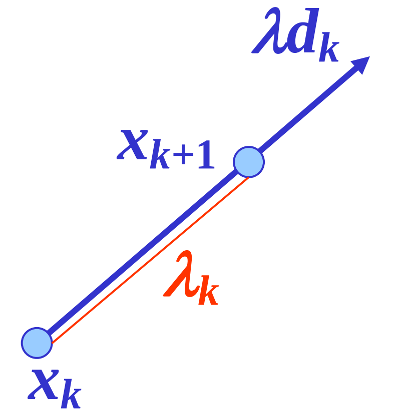
Http://www.amt.ac.cn/member/huxiaodong/

3. 无约束算法 - 线搜索

1-维线搜索是许多求解非线性规划问题的算法的基础。这些算法的步骤可以如下：

迭代算法

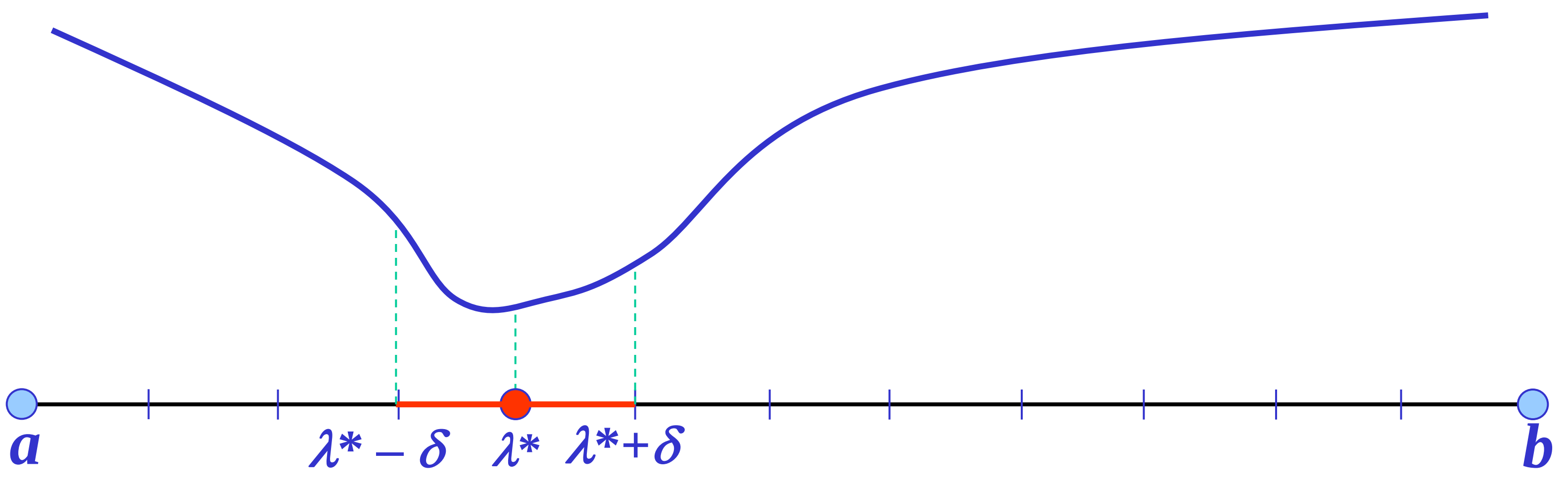
1. 给定一个当前解 x_k ,
2. 输出 x_k 如果它是所需要的解，否则
3. 生成一个搜索方向 d_k,
4. 确定一个合适的步长 λ_k,
5. 置 $x_{k+1} := x_k + \lambda_k d_k$.，并返回步骤1.



确定步长 λ_k 的问题实际上是求解一个子问题：求 $f(x_k+\lambda d_k)$的最小值，其中变量为λ，这是一个 1-维线搜索问题。 变量λ 可以取任意实数，也可以限定$\lambda > 0$ 使得 $x_k+\lambda d_k$ 是一个可行解。

3. 无约束算法 - 线搜索（续一）

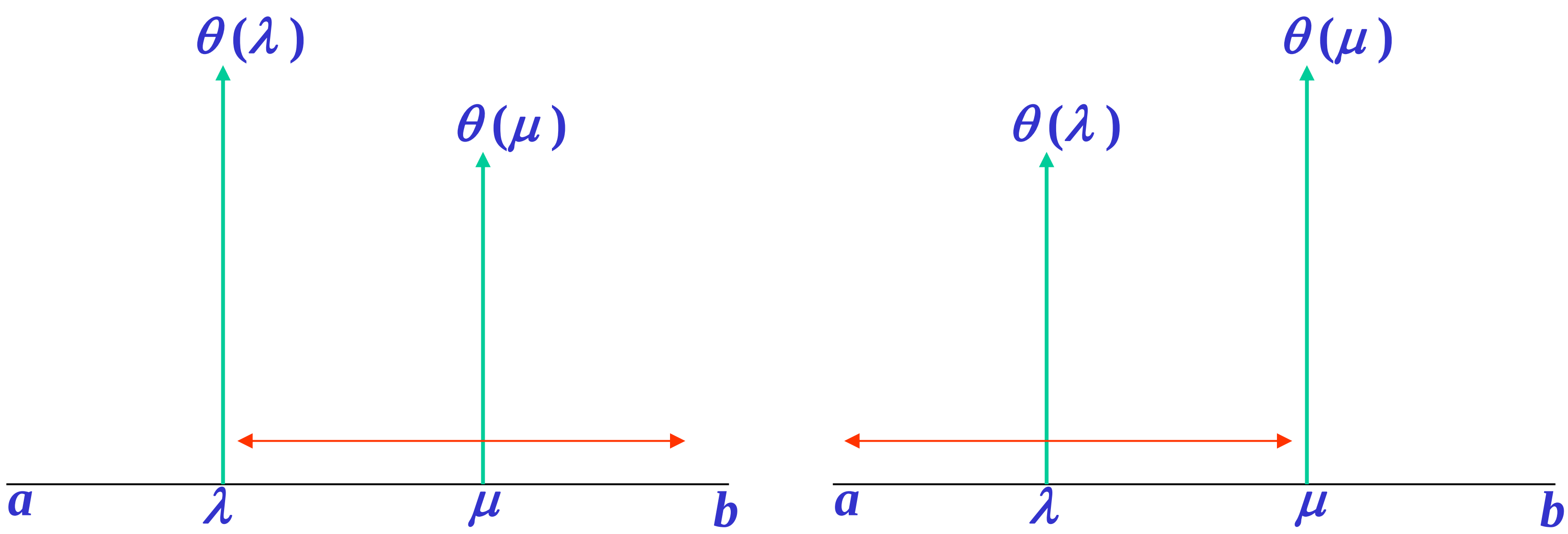
　　一致搜索属于**同步搜索**，它在进行搜索之前就已经确定了要计算目标函数在哪些点的值。我们通过格点 $a+k\delta$ 可以将不确定的区间 $[a, b]$ 分割成若干个小的子区间，其中 $k=1, 2, \ldots, n$，$b=a+(n+1)\delta$。计算函数 θ 在 n 个格点中的每一个格点处的函数值。设函数 θ 在格点 λ^* 处的函数值最小。若 θ 是一个严格拟凸函数，则函数 θ 的最小解位于区间 $[\lambda^*-\delta, \lambda^*+\delta]$ 中。



3. 无约束算法 - 线搜索（续二）

定理 34. 设$\theta: E^1 \to E^1$ 在区间$[a, b]$上严格拟凸的。设 $\lambda, \mu \in [a, b]$, 且 $\lambda < \mu$。若 $\theta(\lambda) > \theta(\mu)$，则对所有$z \in [a, \lambda)$，都有 $\theta(z) \geq \theta(\mu)$。若$\theta(\lambda) \leq \theta(\mu)$，则对所有$z \in (\mu, b]$，都有$\theta(z) \geq \theta(\lambda)$。

证明. (练习) 根据反证法和拟凸函数的定义，可知



3. 无约束算法 - 黄金分割法

与一致搜索方法不同：这种同步搜索方法，在进行任何搜索之前，就决定了要计算哪些点的目标函数值，**序贯线搜索**在进行每一步搜索时，要利用已经搜索过的点（的函数值）的信息。

为了比较各种序贯线搜索的效率，我们引入如下 降减比：

$$\frac{k\text{ 次搜索以后仍然不确定的区间的总长度}}{\text{在进行搜索以前不确定的区间的总长度}}$$

显然，比值越小的线搜索，其效率就越高。

3. 无约束算法 - 黄金分割法（续一）

步 1. 当$b_k - a_k < \varepsilon$, 终止 (最优解在$[a_k, b_k]$中)

若 $\theta(\lambda_k) > \theta(\mu_k)$, 则转到步2, 若$\theta(\lambda_k)\le\theta(\mu_k)$, 则转到步3。

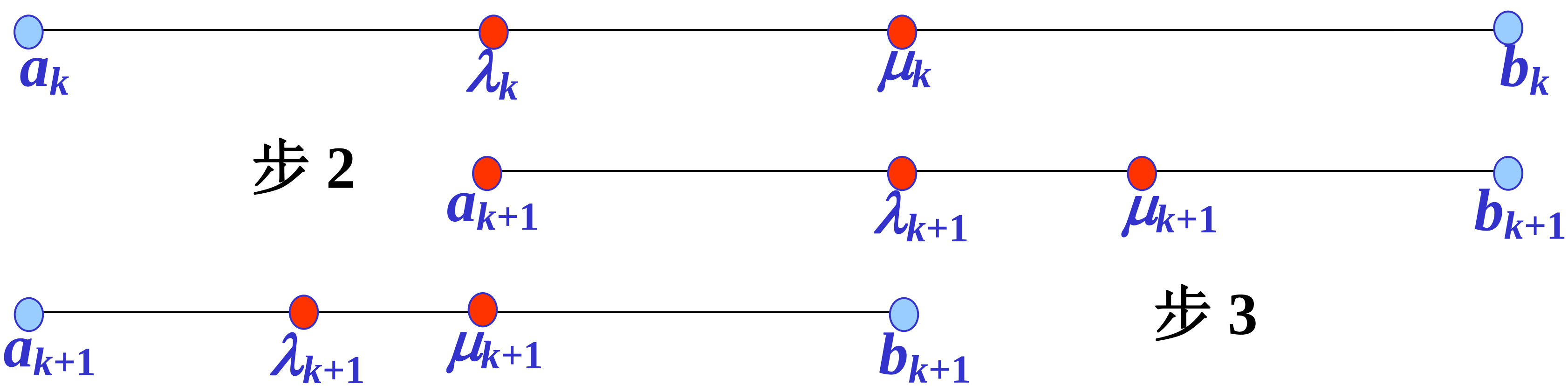
步 2. 置$a_{k+1} := \lambda_k$，$b_{k+1} := b_k$；再置$\lambda_{k+1} := \mu_k$，

并置$\mu_{k+1} := a_{k+1} + \alpha(b_{k+1} - a_{k+1})$；计算$\theta(\mu_{k+1})$，转到步4。

步 3. 置 $a_{k+1} := a_k$，$b_{k+1} := \mu_k$；再置 $\mu_{k+1} := \lambda_k$，

并置$\lambda_{k+1} := a_{k+1} + (1-\alpha)(b_{k+1} - a_{k+1})$；计算$\theta(\lambda_{k+1})$，转到步4。

步 4. 置 $k := k+1$，然后返回 步 1。



3. 无约束算法 - 黄金分割法（续二）

注意，当我们选取参数α使得 $\alpha + \alpha^2 - 1 = 0$，则有

$\lambda_k = a_k + (1 - \alpha)(b_k - a_k)$ 和 $\mu_k = a_k + \alpha(b_k - a_k)$。

因而有$b_{k+1} - a_{k+1} = \alpha(b_k - a_k)$。

在第 k 次迭代后，不确定的区间的长度是原来不确定的区间的长度的 α 倍。在第一次迭代以后，需要计算 λ_k 和 μ_k 这两个数值，但是在随后的每一次迭代中，我们只要计算一个值即可，这是因为此时或者 $\lambda_{k+1} = \mu_k$ 或者 $\mu_{k+1} = \lambda_k$。

除了一致搜索法和黄金分割法以外，还有许多其他的搜索法，比如二分搜索法和斐波纳契搜索法。这些简单的搜索方法都不需要计算目标函数的导数。而其他一些搜索方法，则需要计算目标函数的导数。

3. 无约束算法 - 点-集映射

当设计求解非线性规划问题 $\min\{ f(x) \mid x \in S\}$ 的算法时，我们当然希望能在有限步内找到全局最优解，或者找到一个点列收敛到全局最优解。然而，对于大多数函数 $f(x)$，这只能是奢望。在实际求解时，我们只能希望找到一个点列，它收敛到一个事先设定的**解集 Ω**。比如：

$\Omega=\{ x \in S \mid x \in S$ 是一个局部最优解$\}$；

$\Omega=\{ x \in S \mid f(x) \le b\}$，其中 b 是一个可接受的目标函数值；

$\Omega=\{ x \in S \mid f(x) \le L_b+ \varepsilon \}$，其中 L_b 是最优目标函数值的下界；

$\Omega=\{ x \in S \mid x$ 是一个 **K-K-T**点$\}$。

等等。

3. 无约束算法 - 点-集映射（续一）

定义 13. 称一个点-集映射 $A: X \to X$ 在 $Y \subseteq X$ 上是收敛的，若对于任意一个初始点 $x_1 \in Y$，$x_{k+1} \in A(x_k)$ 的任一个收敛子列的聚点都属于解集 Ω。

考虑求最小值问题 $\min \{ x^2 \mid x \geq 1\}$。设$\Omega$ 为该问题的全局最优解集合，即 $\Omega=\{1\}$。令

$$A(x) = \begin{cases} [3/2 + x/4,\ 1+x/2], & x \geq 2; \\ (x+1)/2, & x < 2. \end{cases}$$

容易验证，当$x_1 \geq 2$ 时，$A(x_k) \to x=2$；而当 $x_1 < 2$ 时，$A(x_k) \to x=1$；这个点-集映射是不收敛的，因为它在 $x=2$ 不具有“连续性”。

3. 无约束算法 - 点-集映射（续二）

考虑一个点-集映射 $A: X \subseteq E^n \to Y \subseteq E^m$。称其在点 $x \in X$ 处是闭的，如果

$$x_k \in X \to x,\ y_k \in A(x_k) \to y, \quad \Rightarrow \quad y \in A(x)。$$

称其在集合 $S \subseteq X$ 是闭的若它在 S 中的每一个点处都是闭的。我们可以将点-集映射的闭性视为函数的连续性的一种推广。

练习. 证明前页中给出的点-集映射在 $x=2$ 处是非闭的。

定理 35. $X \subseteq E^n$ 是一个非空闭集， $A: X \to Y$ 一个点-集映射，$\alpha: X \to E^1$ 是一个连续函数， $\Omega \subseteq X$ 是解集。假设 $x_{k+1} \in A(x_k)$ 属于 X 的一个紧子集，且若 $x \notin \Omega$， $y \in A(x)$， 则 $\alpha(y) < \alpha(x)$。若A 在 Ω 的补集上是闭的，则它收敛，且有 $\alpha(x_k) \to \alpha(x)$，其中 $x \in \Omega$。

3. 无约束算法 - 点-集映射（续三）

证明： 设 $\{x_k\}_K \to x \in X$ 是一个收敛子点列。α 是一个连续函数，所以 $\{\alpha(x_k)\}_K \to \alpha(x)$。因而对于任意 $\varepsilon > 0$，都存在一个正整数 N 使得 $\alpha(x_k) - \alpha(x) < \varepsilon$，$k \in K \geq N$。因此我们有 $\alpha(x_N) - \alpha(x) < \varepsilon$。现考虑充分大的 $k > N$，则有

$$\alpha(x_k) - \alpha(x) = \alpha(x_k) - \alpha(x_N) + \alpha(x_N) - \alpha(x) < 0 + \varepsilon = \varepsilon。$$

故 $\{\alpha(x_k)\} \to \alpha(x)$。

下面用反证法证明 $x \in \Omega$。假设 $x \notin \Omega$。考虑子列 $\{x_{k+1}\}_K$。注意，因为它属于 X 的一个紧集，所以它有一个收敛的子列 $\{x_{k+1}\}_{K'} \to x'$。显然，$\alpha(x') = \alpha(x)$。又因为 A 在 x 处是闭的，所以当 $k \in K'$ 时，有 $x_k \to x$，$x_{k+1} \in A(x_k)$，且 $x_{k+1} \to x'$。故有 $x' \in A(x)$。因此，$\alpha(x') < \alpha(x)$。产生矛盾。因而必有 $x \in \Omega$。

3. 无约束算法 - 最速下降法

最速下降法（柯西1847年）是求多变量可微函数最小值的一个最基本的方法。

引理 7. 设函数 $f(x): E^n \to E^1$ 在点 x^* 处可微，且 $\nabla f(x^*) \neq \mathbf{0}$。则最小值问题 $\min\{ f'(x^*; d) \mid \|d\| \leq 1\}$ 的最优解为

$$d^* \equiv -\nabla f(x^*)/\|\nabla f(x^*)\|。$$

证明. (练习*) 由函数 $f(x)$ 在点 x^* 处可微，可得

$$f'(x^*, d) = \lim_{\lambda \to 0+} \frac{f(x^*+\lambda d) - f(x^*)}{\lambda} = \nabla f(x^*) d。$$

因而引理中求最小值的问题转化为最小值问题

$$\min\{\nabla f(x^*) d \mid \|d\| \leq 1\}。$$

再由施瓦兹不等式，得 $\nabla f(x^*) d \geq -\|\nabla f(x^*)\| \cdot \|d\| \geq -\|\nabla f(x^*)\|$。

由此可知，$\nabla f(x^*) d$ 在 $d^* \equiv -\nabla f(x^*)/\|\nabla f(x^*)\|$ 处达到最小值。

3. 无约束算法 - 最速下降法（续一）

步 0. 取 $\varepsilon > 0$ 为算法终止参数。

选取一个初始解 x_k，置 $k:=1$，并转入 **步 1**。

步 1. 若 $\|\nabla f(x_k)\| < \varepsilon$，则算法终止；否则置 $d_k := -\nabla f(x_k)$。

步 2. 求关于 λ 的函数 $f(x_k+\lambda d_k)$ 的最小值，并令其为 λ_k。

步 3. 置 $x_{k+1} := x_k + \lambda_k d_k$，$k := k+1$，返回 **步 1**。

定理 36. 若最速下降法生成的点列属于一个紧集，则它收敛于一个梯度为 0 的点。

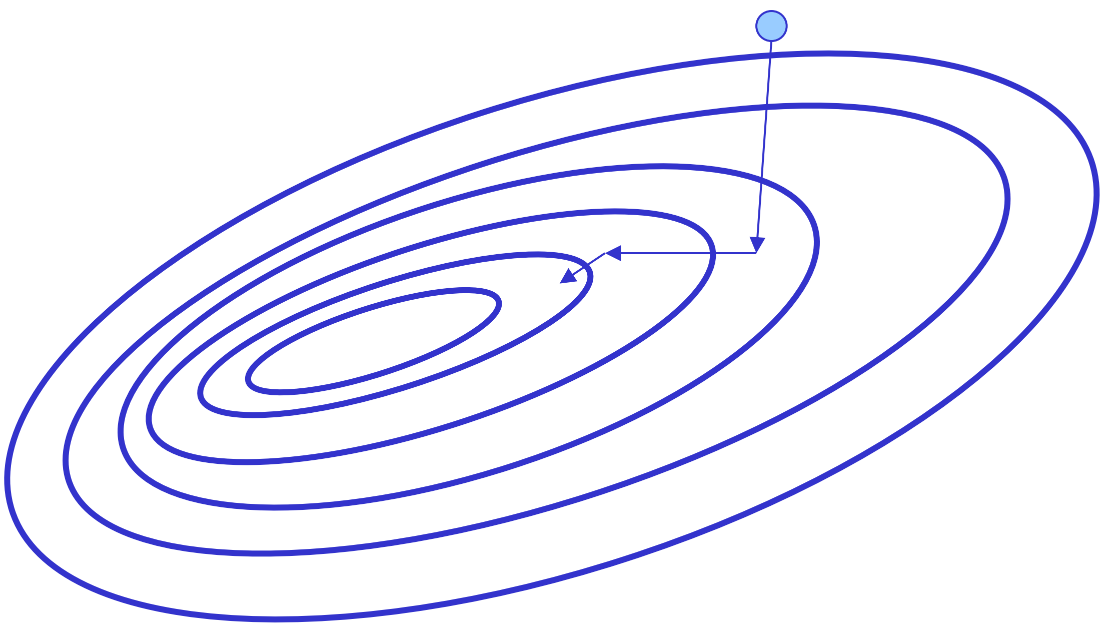
证明. (练习*) 应用**定理 35**。首先，取

$$A(x_k) = x_k + \lambda d_k,\quad \Omega = \{\, x \mid \nabla f(x) = 0 \,\},\quad \alpha(x) = f(x)。$$

然后，验证它们满足定理的所有条件即可。

3. 无约束算法 - 最速下降法（续二）

最速下降法在优化过程的开始阶段通常非常有效。然而，当接近驻点时，却常常会产生相互垂直的、非常小的搜索步（称**锯齿现象**）。



3. 非线性规划 - 算法中止规则

求解非线性规划的算法，在大多数情况下，会生成收敛到解集的一系列点（在极限意义下），因而在运行算法时，我们需要引入一些终止迭代的规则。常用的有以下几个：

1. $\|x_{k+N} - x_k\| < \varepsilon$，$N$ 次迭代以后**绝对距离**；
2. $\dfrac{\|x_{k+1} - x_k\|}{\|x_k\|} < \varepsilon$，$1$ 次迭代以后的**相对距离**；
3. $f(x_k) - f(x_{k+N}) < \varepsilon$，$N$ 次迭代以后**绝对改进量**；
4. $\dfrac{f(x_k) - f(x_{k+N})}{f(x_k)} < \varepsilon$，$1$ 次迭代以后**相对改进量**；
5. $\theta(x_k) - \theta(x_{solution}) < \varepsilon$，$N$ 次迭代以后与解集的**绝对距离**。

设数列 $\{r_k\}$ 收敛到 r^*，则其**收敛阶**定义为满足下述等式的非负 p 的上确界

$$\overline{\lim_{k\to\infty}} \frac{|r_{k+1} - r^*|}{|r_k - r^*|^p} = \beta < \infty$$

当 $p=1$ 时，称数列**线性收敛**如果收敛率 β 小于 1。当$p > 1$时，或者 $p = 1$且 $\beta = 0$ 时，称数列**超线性收敛**。

如果我们用 r_k 表示 $\alpha(r_k)$，即第 k 次迭代时的下降函数值，目标函数值 $f(x_k)$ 或者梯度函数 $\nabla f(x_k)$ 的模，那么 p 的值越大，相应的算法就收敛得越快。

练习. 构造一个超线性收敛数列。

3. 非线性规划 - 算法收敛性（续一）

然而，需要强调的是，不能将收敛阶和收敛率作为衡量算法收敛好坏的惟一标准，因为它们仅仅反应的是算法的迭代次数趋向于无穷大时，算法的收敛进程。

衡量和比较算法好坏的另外一个常用收敛标准是，它们求解二次函数的最小值时的收敛性。这是因为，在非线性函数的最小值点附近，线性近似它效果往往不太好，然而用二次函数近似它会有比较满意的效果。因此，我们可以认为，如果一个算法求二次函数的最小值时效果不佳，那么用它求解一般的非线性函数的最小值时，不太可能有优异的表现（尤其是当搜索到最优点附近时）。

3. 无约束算法 - 牛顿法

由于最速下降法是利用目标函数 $f(x)$ 的线性近似来确定搜索方向，故在最优点附近所生成的搜索方向并不是非常有效。这一点可以从下述表达式看出：

$$f(x_k+\lambda d) = f(x_k) + \lambda\nabla f(x_k)d + \lambda\,\|d\|\,\alpha(x_k\,;\,\lambda d),$$

其中当 $\lambda d \to 0$ 时 $\alpha(x_k;\lambda d) \to 0$，且 $d = -\nabla f(x_k)$。当 x_k 非常接近梯度为 0 的驻点时，$\lambda\,\|\nabla f(x_k)\|^2$ 会变得非常的小。

牛顿法通过用海森阵的逆矩阵乘以负梯度得到一个偏离最速下降方向的搜索方向。 这么做的原因是，构造目标函数的二次近似的一个适合方向，而不是像**最速下降法**那样，利用目标函数的一次近似。

1643—1727

3. 无约束算法 - 牛顿法（续一）

为了进一步说明牛顿法的原理，考虑函数 $f(x)$ 在点 x_k 处的二次近似 $q(x)$：

$$q(x) = f(x_k) + \nabla f(x_k)(x - x_k) + (x - x_k)^T H(x_k)(x - x_k)/2,$$

其中 $H(x_k)$ 为函数 f 在 x_k 处的海森阵。二次近似 q 在点 x 处达到最小值的必要条件是 $\nabla q(x) = 0$ 或者 $\nabla f(x_k)+H(x_k)(x - x_k)=0$。若假定海森阵 $H(x_k)$ 存在拟矩阵，则迭代后得到的点为

$$x_{k+1} := x_k - H(x_k)^{-1}\nabla f(x_k)$$

一般来讲， 牛顿法生成的点列并不一定收敛。理由很简单：海森阵 $H(x_k)$ 可能是奇异阵，也就无法生成新的迭代点 x_{k+1}。即使海森阵的逆存在，新的函数值 $f(x_{k+1})$ 也未必一定小于 $f(x_k)$！不过，如果初始点足够接近驻点 x^* ， 其梯度$\nabla f(x^*)=0$ 且海森阵 $H(x^*)$ 是满秩的，那么牛顿法会收敛到该驻点 x^*。

3. 无约束算法 - 共轭方向法

定义 14. 设 H 是一个 $n \times n$ 对称矩阵。称向量 $d_1, \ldots, d_k$ 为 **H-共轭**，或简单称**共轭**如果它们是线性无关的且 $d_i^T H d_j = 0$，$i \neq j$。

$\mathrm{Min}\ -12y + 4x^2 + 4y^2 - 4xy$

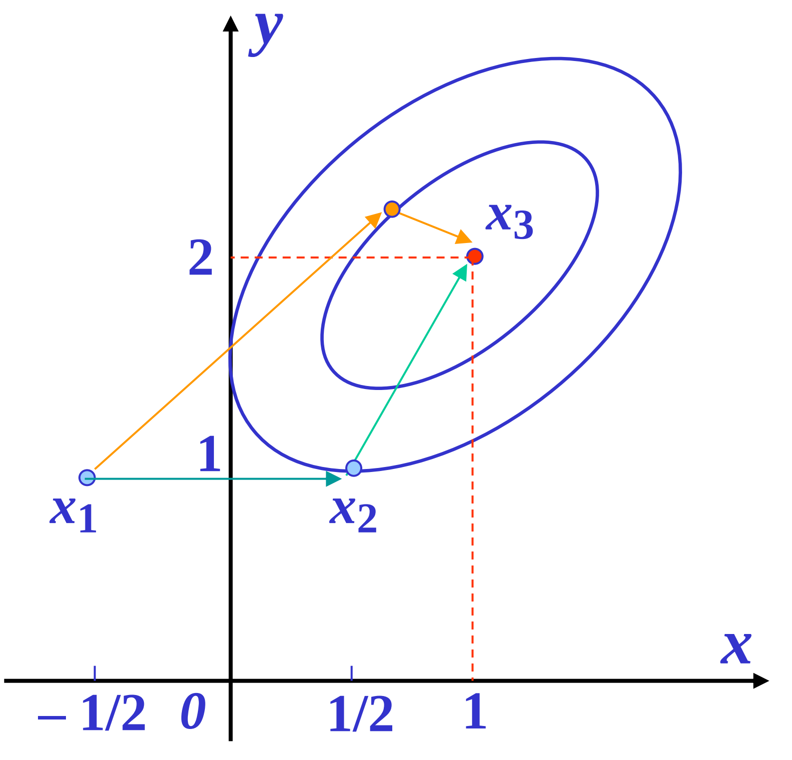
其海森阵为

$$H = \begin{bmatrix} 8 & -4 \\ -4 & 8 \end{bmatrix}$$

假定选取 $d_1=(1,0)$。那么与其共轭的方向 $d_2=(a, b)$ 必须满足

$$d_1^T H d_2 = 0 = 8a - 4b。$$

例如， 可选 $a=1$ 和 $b=2$。



3. 无约束算法 - 共轭方向法（续一）

定理 37. 设 $f(x) = cx + xHx/2$，其中 H 是一个 $n\times n$ 对称矩阵，$d_1, \ldots, d_n$ 是一组 H-共轭的向量，x_1 为一个初始解。另设 λ_k 是关于变量 $\lambda\in E^1$ 的最小值问题 $f(x_k+\lambda d_k)$ 的最优解。
若 $x_{k+1} := x_k + \lambda_k d_k$。则对于 $k = 1, 2, \ldots, n$，有

i) $\nabla f(x_{k+1})\, d_j = 0$，$j=1, \underline{2}, \ldots, k$；

ii) $\nabla f(x_1)\, d_k = \nabla f(x_k)\, d_k$；

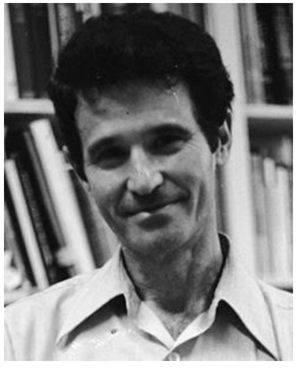
iii) x_{n+1} 是函数 $f(x)$ 的最优解。

证明： 一系列较复杂的计算。

考虑到，任意一个函数在其最小点附近可以用二次函数很好地近似，上述性质说明共轭梯度法不仅能求出二次函数的最小值，而且还能有助于求解非二次函数的最小值。

3. 无约束算法 - 共轭方向法（续二）

下面我们来说明如何生成共轭方向。最著名的一个方法是**变尺度法**(简称 **DFP**)，它是由 **W. C. Davidon** [1959]， **R. Fletcher** 和 **M. Powell** [1963]提出来的。



1927-2013

1939-2016

1936-2015

这个算法可以归为**拟牛顿类算法**，其搜索方向为 $-D_j\nabla f(y)$。将目标函数的梯度乘以 $-D_j$ 即可得到负梯度方向的一个偏离，其中 D_j 是一个 $n\times n$ 正定对称矩阵，它可以视为海森阵的逆的一个近似。在迭代的下一步，将 D_j 加上两个秩为**1**的对称矩阵即可得到 D_{j+1}。因此这个方法有时也称为**秩2-修正法**。

3. 无约束算法 - 共轭方向法（续三）

步 0. 取一个 $\varepsilon > 0$ 为算法终止参数。选取一个初始解 x_1 和一个初始对称正定矩阵D_1。置 $y_k := x_l$，$k:=j:=1$，转到**步 1**。

步 1. 若$\|\nabla f(x_k)\| < \varepsilon$，则终止；否则置 $d_j := -D_j \nabla f(y_j)$。

步 2. 求函数 $f(x_k + \lambda d_k)$ 的最小值，并令其为 λ_j。

步 3. 置 $y_{j+1} := y_j + \lambda_j d_j$。若 $j < n$，转入 **步 4**。若 $j = n$，置 $y_1 := x_{k+1} := y_{n+1}$，$k := k+1$，$j:=1$，并转入 **步 1**。

步 4. 置

$$D_{j+1} := D_j + \frac{p_j p_j^T}{p_j^T q_j} - \frac{D_j q_j q_j^T D_j}{d_j^T D_j q_j},$$

$$p_j := \lambda_j d_j,$$

$$q_j := \nabla f(y_{j+1}) - \nabla f(y_j)$$

置 $j := j+1$，然后返回 **步 1**。

3. 无约束算法 - 共轭方向法（续三）

引理 8. 设 $y_1 \in E^n$，D_1 是初始对称正定矩阵。对每个 $j = 1, 2, \ldots, n$，置 $y_{j+1} := y_j + \lambda_j d_j$，其中 $d_j := -D_j \nabla f(y_j)$，$\lambda_j$ 是关于 λ 的函数 $f(x_k + \lambda d_k)$ 的最小值点。若 $\nabla f(y_j) \neq 0$，$j = 1, 2, \ldots, n$，则 $D_1, D_2, \ldots, D_n$ 是对称正定矩阵且 $d_1, d_2, \ldots, d_n$ 都是下降方向。

定理 38. 设 $f(x) = c^T x + x^T H x/2$，其中 H 是一个 $n \times n$ 对称矩阵。假定任取初始 y_1 和对称正定矩阵 D_1，用变尺度法求函数 $f(x)$ 的最小值，$x \in E^n$。设 λ_k 是关于 $\lambda \geq 0$ 的函数 $f(y_k + \lambda d_k)$ 的最小值解，令 $y_{j+1} := y_j + \lambda_j d_j$。若对每一个 j，都有 $\nabla f(y_j) \neq 0$，则 $d_1, \ldots, d_n$ 是一组 H-共轭方向，且 $D_{n+1} = H^{-1}$。而且，y_{n+1} 是目标函数 $f(x)$ 的最小解。