凸集、凸函数与凸规划

凸集：S 中任意两点连线仍属于 S

凸函数：曲线上任意两点的弦不在曲线下方

凸函数性质：设 f(x）是二阶可微的， f(x)在开凸集 R 上为凸函数的充分必要条件是对一切 x∈ R，Hesse 矩阵 H(x)为半正定的。若 H(x)为正定的，则 f(x)为严格凸函数。

凸规划：求凸函数在凸集上的极小值

凸规划性质：局部最小就是全局最小

两点估计太高，一点估计太低：用凸函数两点之间的连线上的一点 R 来估计函数值 L，永远有 R>L，（图 1 ）。用凸函数的切线上的一点 R 来估计函数值 L，永远有 R<L，（图 2 ）。

斐波那契

给定初始区间$[a_1, b_1]$和最终区间长度 L. 求计算函数值的次数 n,使

$$F_n \geqslant (b_1 - a_1)/L,$$

置辨别常数 $\delta>0$. 计算试探点 λ_1 和 μ_1,

$$\lambda_1 = a_1 + \frac{F_{n-2}}{F_n}(b_1 - a_1), \quad \mu_1 = a_1 + \frac{F_{n-1}}{F_n}(b_1 - a_1).$$

计算函数值 $f(\lambda_1)$和 $f(\mu_1)$. 置 $k=1$.

(2) 若 $f(\lambda_k)>f(\mu_k)$,则转步骤(3);若 $f(\lambda_k)\leqslant f(\mu)$,则转步骤(4).

(3) 令 $a_{k+1}=\lambda_k$, $b_{k+1}=b_k$, $\lambda_{k+1}=\mu_k$,计算试探点 μ_{k+1},

$$\mu_{k+1} = a_{k+1} + \frac{F_{n-k-1}}{F_{n-k}}(b_{k+1} - a_{k+1}).$$

若 $k=n-2$,则转步骤(6);否则,计算函数值 $f(\mu_{k+1})$,转步骤(5).

(4) 令 $a_{k+1}=a_k$, $b_{k+1}=\mu_k$, $\mu_{k+1}=\lambda_k$ 计算 λ_{k+1},

$$\lambda_{k+1} = a_{k+1} + \frac{F_{n-k-2}}{F_{n-k}}(b_{k+1} - a_{k+1}).$$

若 $k=n-2$,则转步骤(6);否则,计算 $f(\lambda_{k+1})$,转步骤(5).

(5) 置 $k:=k+1$,转步骤(2).

置辨别常数 $\delta>0$. 计算试探点 λ_1 和 μ_1,

$$\lambda_1 = a_1 + \frac{F_{n-2}}{F_n}(b_1 - a_1), \quad \mu_1 = a_1 + \frac{F_{n-1}}{F_n}(b_1 - a_1).$$

计算函数值 $f(\lambda_1)$和 $f(\mu_1)$. 置 $k=1$.

(2) 若 $f(\lambda_k)>f(\mu_k)$,则转步骤(3);若 $f(\lambda_k)\leqslant f(\mu)$,则转步骤(4).

(3) 令 $a_{k+1}=\lambda_k$, $b_{k+1}=b_k$, $\lambda_{k+1}=\mu_k$,计算试探点 μ_{k+1},

$$\mu_{k+1} = a_{k+1} + \frac{F_{n-k-1}}{F_{n-k}}(b_{k+1} - a_{k+1}).$$

若 $k=n-2$,则转步骤(6);否则,计算函数值 $f(\mu_{k+1})$,转步骤(5).

(4) 令 $a_{k+1}=a_k$, $b_{k+1}=\mu_k$, $\mu_{k+1}=\lambda_k$ 计算 λ_{k+1},

$$\lambda_{k+1} = a_{k+1} + \frac{F_{n-k-2}}{F_{n-k}}(b_{k+1} - a_{k+1}).$$

若 $k=n-2$,则转步骤(6);否则,计算 $f(\lambda_{k+1})$,转步骤(5).

(5) 置 $k:=k+1$,转步骤(2).

(6) 令 $\lambda_n=\lambda_{n-1}$, $\mu_n=\lambda_{n-1}+\delta$,计算 $f(\lambda_n)$和 $f(\mu_n)$.

若 $f(\lambda_n)>f(\mu_n)$,则令

$$a_n = \lambda_n, \quad b_n = b_{n-1}.$$

若 $f(\lambda_n)\leqslant f(\mu_n)$,则令

$$a_n = a_{n-1}, \quad b_n = \lambda_n.$$

停止计算,极小点含于$[a_n, b_n]$.

黄金分割法和斐波那契法：斐波那契法在求解过程中利用斐波那契数列来简化运算.使得是下一步计算与上一步迭代相关，减少了计算量，另外，该算法在迭代次数较少的时候收敛速度大于黄金分割法，因此是效率和精度的最优平衡。由于黄金分割法和斐波那契法的渐进收敛率趋于一致，所以可以认为黄金分割法是斐波那契法的特例。

求 $f(t)=t^2-t+2$的近似极小点，区间为[-1,3]，

精度$\delta=0.5$

解

$f(t)$为下单峰函数，用微分法可知$t^*=0.5$, $f(t^*)=$

1 75

$L_n=\frac{L_1+F_{n-2}\varepsilon}{F_n}$ $\qquad \varepsilon=0$ 时 $L_n=\frac{L_1}{F_n}$

$L_n=\frac{L_1}{F_n}\le 0.5$ $\quad\rightarrow\quad F_n\ge\frac{3-(-1)}{0.5}$=8 $\quad\rightarrow\quad$ n=5

计算$L_2=\frac{F_{n-1}L_1}{F_n}=\frac{F_3L_1}{F_4}=\frac{5}{8}\times(4)$=2.5

计算a_1, b_1

x_1^1=$b_0-L_2=3-2\ 5$=0.5

x_2^1=$a_0+L_2=-1+2.5$=1 5

$f(x_1^1)=1.75\quad f(x_2^1)=2.75\rightarrow\quad f(x_2^1)>f(x_1^1)$

更新区间 a_1=$a_0=\quad 1,\ b_1$=$x_2^1=1.5$

更新点$x_2^2=x_1^1$=0.5, $x_1^2=a_1$+b_1-x_2^2=0

黄金分割法

求 $f(t)=t^2-t+2$的近似极小点，区间为[-1,3]，

精度$\delta=0.5$. 使用 0.618 法。

计算a_1, b_1.

x_1^1=$a_0+0.382\times(b_0-a_0)$ =0.528

x_2^1=$a_0+0.618\times(b_0\quad a_0)$=1 472

$f(x_1^1)=1.750\,8,\ f(x_2^1)=2.6948\rightarrow f(x_2^1)>f(x_1^1)$

更新区间 a_1=$a_0=\quad 1$

b_1=$x_2^1=1.472$

更新点 $\qquad x_1^2=a_1$+$0.382\times(b_1\quad a_1)$= -0.0 557

$x_2^2=x_1^1$=0.528

由于$\frac{b_3-a_3}{2}=0.4720<\delta=0.5$ $\qquad$ **算法终止**

$x^*=\frac{x_1^3+x_2^3}{2}$=0.4164 $\quad f(x^*)$=1.7570

拉格朗日函数

- 作业(6_5)
  - 求极大和极小
  - $f(x)=2x^2-3y^2-2x$

  $s.t\quad x^2+y^2\le 1$

  令：$\theta^2=1-x^2-y^2$

  $$L=2x^2-3y^2-2x+\lambda(\theta^2-1+x^2+y^2)$$

  - $\frac{\partial L}{\partial x}=4x-2+2\lambda x=0$
  - $\frac{\partial L}{\partial y}=-6y+2\lambda y=0$
  - $\frac{\partial L}{\partial \lambda}=\theta^2-1+x^2+y^2=0$
  - $\frac{\partial L}{\partial \theta}=2\lambda\theta=0$

- 1) $\theta=0$
- $4x-2+2\lambda x=0$
- $-6y+2\lambda y=0$
- $-1+x^2+y^2=0$

解得：

$$\begin{pmatrix}0.2\\ \pm\sqrt{0.96}\\ 3\end{pmatrix},\ \begin{pmatrix}1\\0\\-1\end{pmatrix},\ \begin{pmatrix}-1\\0\\-3\end{pmatrix}$$

-3.2，0，4

- 2) $\lambda=0$
- $4x-2=0$
- $-6y=0$
- $-1+x^2+y^2=\theta^2$

解得：

$$\begin{pmatrix}0.5\\0\\ \pm\sqrt{1.25}\end{pmatrix}$$

-0.5

- 3) $\lambda=0, \theta=0$
- $4x-2=0$
- $-6y=0$
- $-1+x^2+y^2=0$

无解。

故：极大值为4，极小值为-3.2

最速下降法-梯度法

- 取初始点$x^0\in E^n$，允许误差$\varepsilon>0$.
- 计算负梯度方向 $d^p=-\nabla f(x^p)$, $\overline{d^p}=-\frac{\nabla f(x^p)}{\|\nabla f(x^p)\|}$ （可
- 进行一维搜索$\min\limits_k f(x^p+kd^p)$
- 迭代： $x^{p+1}=x^p+kd^p$
- 精度判断为$\|d^p\|\le\varepsilon$

- 例4_3
  - 求$f(x)=4x_1+6x_2-2x_1^2-2x_1x_2-2x_2^2$ 的极值, $x^0=(1,1)^T, \varepsilon=0.5$
  - $\nabla f(x)=\begin{pmatrix}4-4x_1-2x_2\\6-2x_1-4x_2\end{pmatrix}$

  $\nabla^2 f(x)=\begin{pmatrix}-4&-2\\-2&-4\end{pmatrix}$

  - $AC-B^2>0$，且$A<0, C<0$具有极大值。

- 解：令$f(x)=-f(x)=-4x_1-6x_2+2x_1^2+2x_1x_2+2x_2^2$
- $\nabla f(x)=\begin{pmatrix}-4+4x_1+2x_2\\-6+2x_1+4x_2\end{pmatrix}$，$\|\nabla f(x_0)\|=2>\varepsilon$
- 1）第一次迭代

$d^0=-\nabla f(x_0)=\begin{pmatrix}-2\\0\end{pmatrix}$

$x^1=x^0+k^0d^0=\begin{pmatrix}1\\1\end{pmatrix}+k^0\begin{pmatrix}-2\\0\end{pmatrix}=\begin{pmatrix}1-2k^0\\1\end{pmatrix}$

- $g(k)=f(x^1)=-4(1-2k^0)-6\times1+2\times(1-2k^0)^2-2\times(1-2k^0)+2$

$$\frac{dg(k)}{dk}=-16k^0+4$$

$$\widehat{k^0}=1/4$$

$$x^1=\begin{pmatrix}1-2\times\frac{1}{4}\\1\end{pmatrix}=\begin{pmatrix}\frac{1}{2}\\1\end{pmatrix}$$

$\nabla f(x_1)=\begin{pmatrix}0\\-1\end{pmatrix}$,$\|\nabla f(x_0)\|=1>\varepsilon$

- 2）第二次迭代

$d^1=-\nabla f(x_1)=\begin{pmatrix}0\\1\end{pmatrix}$

$x^2=x^1+k^1d^1=\begin{pmatrix}\frac{1}{2}\\1\end{pmatrix}+k^1\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}\frac{1}{2}\\1+k^1\end{pmatrix}$

- $g(k)=f(x^2)=-4\left(\frac{1}{2}\right)-6\times(1+k^1)+2\times\left(\frac{1}{2}\right)^2+2\times(1+k^1)\times\frac{1}{2}+2\times(1+k^1)^2$

$$\frac{dg(k)}{dk}=4k^1-1$$

$$\widehat{k^1}=1/4$$

$$\widehat{k^1}=1/4,\ d^1=\begin{pmatrix}0\\1\end{pmatrix}$$

$$x^2=\begin{pmatrix}\frac{1}{2}\\1+\frac{1}{4}\end{pmatrix}=\begin{pmatrix}\frac{1}{2}\\\frac{4}{5}\end{pmatrix}$$

$\nabla f(x^2)=\begin{pmatrix}-1/2\\0\end{pmatrix}$，$\|\nabla f(x^2)\|=0.25<\varepsilon$

算法终止

问题：对于全局而言最速下降方向未必是最快的。在一般情况下，当用最速下降法寻找极小点时，其搜索路径呈直角锯齿状（如下图），在开头几步，目标函数下降较快；但在接近极小点时，收敛速度长久不理想了。特别适当目标函数的等值 线为比较扁平的椭圆时，收敛就更慢了。

收敛速度影响因素：搜索的方向。。。

优化：在实用中常用最速下降法和其他方法联合应用，在前期使用最速下降法，而在接近极小值点时，可改用收敛较快的其他方法。

牛顿法

Step1 给定初始点 x0，允许误差 ε>0.置 k=1，

Step2 若$\|\nabla f(x^k)\|$< ε,停止，得解x^k，否则，令

$x^{k+1}=x^k\quad\nabla^2 f(x^k)\ ^1\nabla f(x^k)$ ，k=k+1，转 step2

例题

求 $f(x)=4x_1^2+2x_1x_2+2x_2^2+x_1+x_2$ 的近似极小点。使用牛顿法。

$$=\begin{pmatrix}8x_1+2x_2+1\\2x_1+4x_2+1\end{pmatrix}, H=\begin{pmatrix}8&2\\2&4\end{pmatrix}$$

$$H^{-1}=\frac{1}{14}\begin{pmatrix}2&-1\\-1&4\end{pmatrix}$$

$$x_{n+1}=x_n-H^{-1}(x_n)\nabla f(x_n)$$

$$=\begin{pmatrix}x_1^n\\x_2^n\end{pmatrix}-\frac{1}{14}\begin{pmatrix}2&-1\\-1&4\end{pmatrix}\begin{pmatrix}8x_1+2x_2+1\\2x_1+4x_2+1\end{pmatrix}$$

$$=\begin{pmatrix}-\frac{1}{14}\\-\frac{3}{14}\end{pmatrix}$$

广义牛顿法－二阶导数法

- 二阶导数法（广义牛顿法）
  - 取初始点$x^0\in E^n$，允许误差$\varepsilon>0$.
  - 计算梯度方向 $d^p=-[\nabla^2 f(x^p)]^{-1}\nabla f(x^p)$
  - 进行一维搜索$\min\limits_k f(x^p+kd^p)$
  - $x^{p+1}=x^p+kd^p$
  - 精度判断为$\|d^p\|\le\varepsilon$

- 证明：d^p为函数下降方向。即证$\nabla f(x^p)\cdot d^p<0$
- $\nabla^T f(x^p)\cdot d^p=\nabla^T f(x^p)\cdot-[\nabla^2 f(x^p)]^{-1}\nabla f(x^p)$
  $=-\nabla^T f(x^p)\cdot[\nabla^2 f(x^p)]^{-1}\nabla f(x^p)$

可知当$[\nabla^2 f(x^p)]^{-1}$正定时，$\nabla^T f(x^p)\cdot d^p<0$

故只需$f(x)$为凸函数，即为下降方向。

二阶导数法对于二次函数可以一步达优。

- 例4_5

$\min f(x)=x_1^2+25x_2^2, x^0=(2,2)^T$,精度为0.01

- 解：$\nabla f(x)=\begin{pmatrix}2x_1\\50x_2\end{pmatrix}$

$\nabla^2 f(x)=\begin{pmatrix}2&0\\0&50\end{pmatrix}>0$

$\nabla f(x^0)=\begin{pmatrix}4\\100\end{pmatrix}$, $[\nabla^2 f(x)]^{-1}=\begin{pmatrix}\frac{1}{2}&0\\1&\frac{1}{50}\end{pmatrix}$

$$\|\nabla f(x^0)\|=50.04>0.01$$

$d^0=-[\nabla^2 f(x^0)]^{-1}\nabla f(x^0)=-\begin{pmatrix}\frac{1}{2}&0\\0&\frac{1}{50}\end{pmatrix}\begin{pmatrix}4\\100\end{pmatrix}=-\begin{pmatrix}2\\2\end{pmatrix}$

$\min\limits_k f(x^0+kd^0)=(2-2k)^2+25\times(2-2k)^2=26\times(2-2k)^2$

$\frac{df}{dk}=-104(2-2k)=0\rightarrow k=1$

$x^1=x^0+kd^0=(2,2)^T+(-2,-2)^T=(0,0)^T$

$\|\nabla f(x^1)\|=0$，故达最优。

问题：1)局部收敛，初始点选择不当，可能会导致不收敛；2)牛顿法不是下降算法，当二阶 Hesse 阵非正定时，不能保证是下降方向；3)二阶 Hesse 阵必须可逆，否则算法将无法进行下去；4)对函数分析性质要求苛刻，计算量大，仅适合小规模优化问题。5)该方法需要求二阶导数，以及矩阵的逆，计算速度会很慢

二分法

优点：计算简单，方法可靠；对 f (x) 要求不高(只要连续即可) ；收敛性总能得到保证；

缺点：可在大范围内求根，但收敛较慢,且不能求重根和复根，其收敛速度仅与一个以 1/2 为比值的等比级数相同，通常用于求根的初始近似值,而后在使用其它的求根方法。

FR 共轭梯度法

- **Fletcher-Reeves共轭梯度算法**
  - 1：选取初始点x^0，初始方向$v^0=-\nabla f(x^0)$
  - 2：$for\ i=1,\dots,n$

    2.1：$x^i=x^{i-1}+\lambda_{i-1}v^{i-1}$，$\lambda_{i-1}$取$minf(x^{i-1}+\lambda v^{i-1})$

    2.2：$v^i=-\nabla f(x^i)+\boxed{\frac{\|\nabla f(x^i)\|^2}{\|\nabla f(x^{i-1})\|^2}v^{i-1}}$

    3：$x^0=x^n$.

- **例 4_7**

  用FR共轭梯度法求：

  $minf(x_1,x_2)=\frac{3}{2}x_1^2+\frac{1}{2}x_2^2-x_1x_2-2x_1,x^0=(-2,4)$

  解：第1次迭代

  $$\nabla f(x^0)=(3x_1-x_2-2,x_2-x_1)^T=(-12,6)^T$$

  $$v^0=-\nabla f(x^0)=(12,-6)^T$$

  $x^1=x^0+\lambda v^0=(-2,4)^T+(12\lambda,-6\lambda)^T=(-2+12\lambda,4-6\lambda)^T$

  $f(x^1)=\frac{3}{2}(-2+12\lambda)^2+\frac{1}{2}(4-6\lambda)^2-(-2+12\lambda)(4-6\lambda)-2(-2+12\lambda)$

  $$f'_\lambda(x^1)=612\lambda-180=0$$

  $\lambda=\frac{5}{17},x^1=(\frac{26}{17},\frac{38}{17})^T$

$$\nabla f(x^1)=(3x_1-x_2-2,x_2-x_1)^T=(\frac{6}{17},\frac{12}{17})^T$$

$$d^1=-\nabla f(x^1)+\frac{\|\nabla f(x^1)\|^2}{\|\nabla f(x^0)\|^2}d^0$$

$$=-\left(\frac{6}{17},\frac{12}{17}\right)^T+\frac{\left\|\left(\frac{6}{17},\frac{12}{17}\right)^T\right\|^2}{\|(-12,6)^T\|^2}\cdot\left(\frac{6}{17},\frac{12}{17}\right)^T=(-\frac{90}{289},\frac{210}{289})^T$$

$$x^2=x^1+\lambda d^1=(\frac{26}{17},\frac{38}{17})^T+(-\frac{90\lambda}{289},\frac{210\lambda}{289})^T$$

$$\lambda=\frac{17}{10}$$

$$x^2=(1,1)^T$$

$$\nabla f(x^2)=0$$

共轭梯度法特点：具有全局收敛性；收敛速度快（二次收敛）。对于正定二次函数，具有二次终结性；算法结构简单，计算量小，存储量小。（如 FR 法只要 3 个 n 维向量的存储空间）适用性：特别适用于大规模优化问题的求解。共轭梯度法需要计算梯度，这个过程很耗时

变尺度法-DFP 算子—拟牛顿

- **变尺度算法（DFP）**
  - x^i为搜索点，H_i为H阵的近似。$g^i=\nabla f(x^i)$
  - 1：选取 x^0, H_0
  - 2：for i=0,1,…n-1

    $x^{i+1}=x^i+\lambda_iv^i$,其中$v^i=-H_ig^i$，λ_i为一维搜索最优步长，$\min_\lambda f(x^i+\lambda v^i)$
  - 3：令 $u^i=x^{i+1}-x^i$，$y^i=g^{i+1}-g^i$，

    $H_{i+1}=H_i+A_i-B_i,A_i=\frac{u^i(u^i)^T}{(u^i)^Ty^i},B_i=\frac{H_iy^i(H_iy^i)^T}{(y^i)^TH_iy^i}$
  - 4：判断精度，不满足则$x^0=x^n$,转2.

- **例 4_11**

  用DFP方法求$\min f(x)=\frac{1}{2}x^TAx$, $A=\begin{bmatrix}3&-1\\-1&1\end{bmatrix}$, $x^0=\begin{pmatrix}10\\10\end{pmatrix}$,

  $H_0=\begin{bmatrix}1&0\\0&1\end{bmatrix}$

  解：$\nabla f(x)=\begin{pmatrix}3x_1-x_2\\-x_1+x_2\end{pmatrix}$,

  $\nabla f(x^0)=\begin{pmatrix}20\\0\end{pmatrix}$, $v^0=-H_0\nabla f(x^0)=-\begin{pmatrix}20\\0\end{pmatrix}$

  从x^0出发沿着方向v^0作一维搜索。

  $$\min_\lambda f(x^0+\lambda v^0)$$

  $$\varphi(\lambda)=f(x^0+\lambda v^0)=100-400\lambda+600\lambda^2$$

  $$\frac{d\varphi}{d\lambda}=-400+1200\lambda=0,\lambda_0=0.33$$

  $$x^1=x^0+\lambda_0v^0=(3.34,10)^T$$

$\nabla f(x^1)=\begin{pmatrix}0.00\\6.66\end{pmatrix}$, $u^0=\lambda_0v^0=-\begin{pmatrix}6.6\\0\end{pmatrix}$

$$y^0=\nabla f(x^1)-\nabla f(x^0)=-\begin{pmatrix}20\\-6.66\end{pmatrix}$$

$A_i=\frac{u^i(u^i)^T}{(u^i)^Ty^i},B_i=\frac{H_iy^i(H_iy^i)^T}{(y^i)^TH_iy^i}$

$A_0=\begin{bmatrix}0.33&0\\0&0\end{bmatrix},B_0=-\begin{bmatrix}0.9&0.3\\-0.3&0.01\end{bmatrix}$

$H_{i+1}=H_i+A_i-B_i$

$$H_1=H_0+A_0-B_0=\begin{bmatrix}0.43&0.3\\0.3&0.1\end{bmatrix}$$

- $v^1=-H_1\nabla f(x^1)=-\begin{pmatrix}2\\6\end{pmatrix}$

从x^1出发沿着方向v^1作一维搜索。

$$\min_\lambda f(x^1+\lambda v^1)$$

$$\lambda_1=5/3$$

$$x^2=x^1+\lambda_1v^1=(-0,0)^T$$

$$H_2=H_1+A_1+B_1=\begin{bmatrix}0.5&0.5\\0.5&1.5\end{bmatrix}$$

Powell 算法

- 设有一组线性独立的向量$\{v^i,i=1,\dots,n\}$, x^0为初始点。算法步骤为:
- 1：$for\ i$=1,…,n，找到 $\min_\lambda f(y^{i-1}+\lambda v^i)$,的 λ值，递推下一点 $y^i=y^{i-1}+\lambda v^i$
- 2：$for\ i$=1,…,n，使 $v^i=v^{i+1}$
- 3：$v^n=y^n-y^0$
- 4：找到 $\min_\lambda f(x^n+\lambda(x^n-x^0))$,的 λ值，替换$x^0=y^n+\lambda(y^n-y^0)$
- 5：达不到精度回到step1

- **例 4_9**

  用powell方法求 $minf(x)=(x_1+x_2)^2+(x_1-1)^2$,初始点

  $x^0=\begin{bmatrix}2\\1\end{bmatrix}$，初始搜索方向 $d^{(1,1)}=(1,0)^T$, $d^{(1,2)}=(0,1)^T$

  解：第一轮迭代:

  令 $y^{(1,1)}=x^0$。从$y^{(1,1)}$出发沿着方向$d^{(1,1)}$作一维搜索。

  $$\min_\lambda f(y^{(1,1)}+\lambda d^{(1,1)})$$

  $$y^{(1,1)}+\lambda d^{(1,1)}=(2,1)^T+\lambda(1,0)^T=(2+\lambda,1)^T$$

  $$\varphi(\lambda)=f(y^{(1,1)}+\lambda d^{(1,1)})=(3+\lambda)^2+(1+\lambda)^2$$

  $$\frac{d\varphi}{d\lambda}=2(3+\lambda)+2(1+\lambda)=0,\lambda_1=-2$$

  $$y^{(1,2)}=x^{(1,1)}+\lambda_1d^{(1,1)}=(0,1)^T$$

从$y^{(1,2)}$出发沿着方向$d^{(1,2)}$作一维搜索。

$$\min_\lambda f(y^{(1,2)}+\lambda d^{(1,2)})$$

$\lambda_2=-1$，$y^{(1,3)}=y^{(1,2)}+\lambda_2d^{(1,2)}=(0,0)^T$　(坐标轮换结束)

创建方向$d^{(1,3)}=y^{(1,3)}-y^{(1,1)}=(-2,-1)^T$

求解　$\min_\lambda f(x^{(1,2)}+\lambda d^{(1,3)})$

$\lambda_3=-\frac{2}{13}$，$x^1=y^{(1,3)}+\lambda_3d^{(1,3)}=(\frac{4}{13},\frac{2}{13})^T$

进入第2轮搜索：

初始点 $y^{(2,1)}=x^1=(\frac{4}{13},\frac{2}{13})^T$，

搜索方向为：

$d^{(2,1)}=d^{(1,2)}=(0,1)^T$, $d^{(2,2)}=d^{(1,3)}=(-2,-1)^T$

求解　$\min_\lambda f(y^{(2,1)}+\lambda d^{(2,1)})$

$\lambda_1=\frac{-6}{13}$，$y^{(2,2)}=y^{(2,1)}+\lambda_1d^{(2,1)}=(\frac{4}{13},\frac{-4}{13})^T$

求解　$\min_\lambda f(y^{(2,2)}+\lambda d^{(2,2)})$

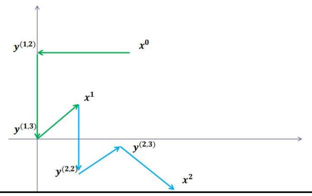
$\lambda_2=\frac{-18}{169}$，$y^{(2,3)}=y^{(2,2)}+\lambda_2d^{(2,2)}=(\frac{88}{169},\frac{-34}{169})^T$

创建方向$d^{(2,3)}=y^{(2,3)}-y^{(2,1)}=(\frac{36}{169},\frac{-60}{169})^T$

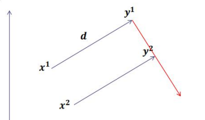
求解　$\min_\lambda f(y^{(2,3)}+\lambda d^{(2,3)})$

$\lambda_3=\frac{9}{4}$，$x^2=y^{(2,3)}+\lambda_3d^{(2,3)}=(1,-1)^T$

- **例 4_9　迭代过程**



- **关于powell算法的共轭方向**
  - 定理：设$f(x)=\frac{1}{2}x^TAx+b^Tx+c$，$A$为$n$阶正定阵。任取方向$d$和点$x^1$，$x^2$。从$x^1$出发沿方向$d$作一维搜索得极小点 y^1，从x^2出发沿方向d作一维搜索得极小点 y^2，则有y^2-y^1与方向d关于A共轭。



- 第一轮搜索方向:

  $d^{(1,1)}=(1,0)^T$, $d^{(1,2)}=(0,1)^T$,　$d^{(1,3)}=(-2,-1)^T$
- 第二轮搜索方向:

  $d^{(2,1)}=(0,1)^T$, $d^{(2,2)}=(-2,-1)^T$,　$d^{(2,3)}=(\frac{36}{169},\frac{-60}{169})^T$

  沿方向$d^{(1,3)}$得 x^1,沿方向$d^{(2,2)}$得 $x^{(2,2)}$,而$d^{(2,3)}=x^{(2,2)}-x^{(2,0)}$，故和方向 $d^{(2,2)}$共轭。

  而x^2是沿共轭方向搜索得到的，因此必为极小点。

K-T 条件

- $\min_x f(x)$

  $s.t.$　$h_i(x)=0$　$i=1,2,\dots,r$

  　　$g_j(x)\le0$　$i=1,2,\dots,n$

  $$L(x,\lambda)=f(x)+\sum_{i=1}^r\mu_ih_i(x)+\sum_{i=1}^n\lambda_ig_i(x)$$

  $$\begin{cases}\frac{\partial f}{\partial x_j}(x^*)+\sum_{i=1}^r\mu_i\frac{\partial h_i}{\partial x_j}(x^*)+\sum_{i=1}^n\lambda_i\frac{\partial g_i}{\partial x_j}(x^*)=0\\h_i(x)=0\\g_i(x^*)\le0\\(\lambda_i^*)g_i(x^*)=0\\\lambda_i^*\ge0\end{cases}$$

- **例 6_8**
  - 用KT条件解NLP问题
  - $\min_x f(x)=(x_1-1)^2+(x_2-2)^2$

    $s.t.$　$x_2-x_1=1$

    　　$x_1+x_2\le2$

    　　$x_1,x_2\ge0$

  解：$h_1(x)=-x_1+x_2-1=0$

  　$g_1(x)=x_1+x_2-2\le0$

  　$g_2(x)=-x_1\le0$

  　$g_3(x)=-x_2\le0$

  $$L(x,\lambda)=f(x)+\sum_{i=1}^r\mu_ih_i(x)+\sum_{i=1}^n\lambda_ig_i(x)$$

$L(x,\mu,\lambda)=(x_1-1)^2+(x_2-2)^2+\mu(-x_1+x_2-1)+\lambda_1(x_1+x_2-2)+\lambda_2(-x_1)+\lambda_3(-x_2)$

根据KT条件有：

$$\frac{\partial L}{\partial x_1}=2(x_1-1)-\mu+\lambda_1-\lambda_2=0$$

$$\frac{\partial L}{\partial x_2}=2(x_2-2)+\mu+\lambda_1-\lambda_3=0$$

$$\lambda_1(x_1+x_2-2)=0$$

$$\lambda_2x_1=0$$

$$\lambda_3x_2=0$$

$$\lambda_1,\lambda_2,\lambda_3\ge0$$

讨论解的情况：

由$h_1(x)=-x_1+x_2-1=0$可设

情形I：$x_1\ne0,x_2\ne0$，则：$\lambda_2=0,\lambda_3=0$

$$2(x_1-1)-\mu+\lambda_1=0$$

$$2(x_2-2)+\mu+\lambda_1=0$$

$$\lambda_1(x_1+x_2-2)=0$$

$$x_2-x_1=1$$

$$x_1+x_2\le2$$

$$x_1,x_2\ge0$$

若$\lambda_1=0$，可得解 $\mu=0$，$x_1=1$，$x_2=2$，不满足$x_1+x_2\le2$，即非鞍点

若$\lambda_1\ne0$，可得解 $\mu=0$，$\lambda_1=1$，$x_1=\frac{1}{2}$，$x_2=\frac{3}{2}$，满足各类条件。

情形II：$x_1\ne0,x_2=0$，则：$\lambda_2=0,\lambda_3\ne0$，得联立方程组：

$$2(x_1-1)-\mu+\lambda_1=0$$

$$2(0-2)+\mu+\lambda_1=0$$

$$\lambda_1(x_1-2)=0$$

$$-x_1=1$$

可得解 $\mu=0$，$\lambda_1=0$，$\lambda_3=-4$，$x_1=-1$，$x_2=0$，不满足约束。

情形III：$x_1=0,x_2\ne0$，则：$\lambda_2\ne0$, $\lambda_3=0$，得联立方程组：

$$2(0-1)-\mu+\lambda_1-\lambda_2=0$$

$$2(x_2-2)+\mu+\lambda_1=0$$

$$\lambda_1(x_1-2)=0$$

$$x_2-1=0$$

可得解 $\mu=0$，$\lambda_1=0$，$\lambda_2=-4$，$x_1=0$，$x_2=1$，满足约束。

坐标轮换法

- **坐标轮换法算法步骤**
  - 1：取$x_1^1\in E^n$，置坐标轴方向：$e_1=[1,0,0,\cdots,0]^T$，$e_2=[0,1,0,\cdots,0]^T$，$e_n=[0,0,0,\cdots,1]^T$
  - 令 $j=1,k=1,x_k^j=x_1^1$；
  - 2：求单变量极值问题的最优解

    $$\min_\lambda f(x_k^j+\lambda e_j)=f(x_k^j+\lambda_je_j)$$

    $$x_k^{j+1}=x_k^j+\lambda_je_j$$
  - 3：判断是否满足$j=n$，若$j=n$，转下一步，否j++，转2.
  - 4：检验是否满足收敛性判别准则

    $\|x_k^{n+1}-x_k^1\|<\varepsilon$,否则令 $x_{k+1}^1=x_k^{n+1}$，$k++,j=1$,转2.

- **例5_1**
  - 用坐标轮换法求$minf(x)=(x_1-2)^4+(x_1-2x_2)^2$

    $x^0=(0,3)^T,\varepsilon=0.04$.

    解：取向量$e_1=(1,0)^T$, $e_2=(0,1)^T$

    $$x^0+\lambda_1e_1=(\lambda_1,3)^T$$
  - $f(x^0+\lambda_1e_1)=[\lambda_1-2]^4+[\lambda_1-6]^2$
  - 求 $\min_\lambda f(x^0+\lambda e_1)$
  - $\frac{df(x^0+\lambda_1e_1)}{d\lambda_1}=0$　→　$\lambda_1=3.13$

    $$x_1^{(1)}=x^0+\lambda_1e_1=\begin{bmatrix}0\\3\end{bmatrix}+3.13\begin{bmatrix}1\\0\end{bmatrix}=\begin{bmatrix}3.13\\3\end{bmatrix}$$

$x_1^{(1)}=\begin{bmatrix}3.13\\3\end{bmatrix}$出发，沿$e_2$方向搜索，求得$x_1^{(2)}$

- $x_1^{(2)}=x_1^{(1)}+\lambda_2e_2=\begin{bmatrix}3.13\\3+\lambda_2\end{bmatrix}$
- $f\left(x_2^{(1)}+\lambda_2e_2\right)=(1.13)^4+(3.13-6-2\lambda_2)^2$
- 求 $\min_\lambda f(x_2^{(1)}+\lambda_2e_2)$
- $\frac{df(x_2^{(1)}+\lambda_2e_2)}{d\lambda_2}=0$　→　$\lambda_2=-1.44$

  $$x_1^{(2)}=x_1^{(1)}+\lambda_2e_2=\begin{bmatrix}3.13\\1.56\end{bmatrix}$$

优点是算法简单，计算量小。其***缺点***是计算效率低，对高维问题尤为突出．因此，坐标轮换法通常用于维数较低的优化问题（一般 n＜10）．对于可微函数，可能收敛于梯度为零的局部最优点。

步长加速发

- 1、给定初始步长$\delta>0$，加速因子$\alpha\ge1$，缩减率$\beta\in(0,1)$，允许误差$\varepsilon>0$及初始迭代点x^1，置$y^1=x^1,k=1,j=1$；
- 2、如果$f(y^j+\delta e_j)<f(y^j)$，令$y^{j+1}=y^j+\delta e_j$，转第4步；否则转第3步。
- 3、如果$f(y^j-\delta e_j)<f(y^j)$，令$y^{j+1}=y^j-\delta e_j$，转第4步；否则，令$y^{j+1}=y^j$，转第4步。
- 4、如果$j<n$，则置$j=j+1$，转第2步；否则，转第5步；
- 5、如果$f(y^{n+1})<f(x^k)$，则转第6步；否则，转第7步；
- 6、令$x^{k+1}=y^{n+1}$，$y^1=x^{k+1}+\alpha(x^{k+1}-x^k)=(1+\alpha)x^{k+1}-x^k$，置$k=k+1,j=1$,转第2步；
- 7、$\delta=\beta\delta,y^1=x^k,x^{k+1}=x^k$，置$k=k+1,j=1$,转第2步。

例5_2

- 用步长加速法求$\min f(x) = (1-x_1)^2+5(x_2-x_1^2)^2$

$x^1=(2,0)^T$,初始步长$\delta=0.5$，加速因子$\alpha=1$，缩减率$\beta=0.5$，坐标方向$e_1=(1,0)^T, e_2=(0,1)^T$

解：先在x^1周围进行探测移动，令 $y^1=(2,0)^T$,探测情况如下：

$f(y^1)=81,\quad y^1+\delta e_1=\begin{bmatrix}2\\0\end{bmatrix}+\frac{1}{2}\begin{bmatrix}1\\0\end{bmatrix}=\begin{bmatrix}\frac{5}{2}\\0\end{bmatrix}$

- $f(y^1+\delta e_1)=197\frac{9}{16}>f(y^1)$,故失败。
- $y^1-\delta e_1=\begin{bmatrix}2\\0\end{bmatrix}+\frac{1}{2}\begin{bmatrix}1\\0\end{bmatrix}=\begin{bmatrix}\frac{3}{2}\\0\end{bmatrix}$
- $f(y^1-\delta e_1)=25\frac{9}{16}<f(y^1)$,成功。

- 因此，令$y^2=y^1-\delta e_1=\begin{bmatrix}\frac{3}{2}\\0\end{bmatrix}$，从$y^2$出发，沿$e_2$探测情况如下：
- $y^2+\delta e_2=\begin{bmatrix}\frac{3}{2}\\0\end{bmatrix}+\frac{1}{2}\begin{bmatrix}0\\1\end{bmatrix}=\begin{bmatrix}\frac{3}{2}\\\frac{1}{2}\end{bmatrix}$,
- $f(y^2+\delta e_2)=15\frac{9}{16}<f(y^2)$,成功。
- 因此，令$y^3=y^2+\delta e_2=\begin{bmatrix}\frac{3}{2}\\\frac{1}{2}\end{bmatrix}$.

- 第一轮探测完成后，由于$f(y^3)<f(y^1)$,因此得到第2个基点

$x^2=y^3=\begin{bmatrix}\frac{3}{2}\\\frac{1}{2}\end{bmatrix}$,再沿方向$x^2-x^1$进行模式移动，令：

- $y^1=x^2+\alpha(x^2-x^1)=2x^2-x^1=2\begin{bmatrix}\frac{3}{12}\\\frac{1}{2}\end{bmatrix}-\begin{bmatrix}2\\0\end{bmatrix}=\begin{bmatrix}1\\1\end{bmatrix}$
- 模式移动后，立即从得到的点 y^1 出发，进行第2轮探测移动。探测情况如下：

- 先沿e_1探测，这时有：
- $f(y^1)=0,\ y^1+\delta e_1=\begin{bmatrix}1\\1\end{bmatrix}+\frac{1}{2}\begin{bmatrix}1\\0\end{bmatrix}=\begin{bmatrix}\frac{3}{2}\\1\end{bmatrix}$,
- $f(y^1+\delta e_1)=8\frac{1}{16}>f(y^1)$,失败
- $y^1-\delta e_1=\begin{bmatrix}1\\1\end{bmatrix}-\frac{1}{2}\begin{bmatrix}1\\0\end{bmatrix}=\begin{bmatrix}\frac{1}{2}\\0\end{bmatrix}$,
- $f(y^1-\delta e_1)=3\frac{1}{16}>f(y^1)$,失败；
- 由于沿e_1的正反向探测均失败，故令$y^2=y^1=\begin{bmatrix}1\\1\end{bmatrix}$

- 从y^2出发沿e_2探测的情况是：
- $f(y^2+\delta e_2)=\frac{5}{4}>f(y^2)$,失败
- $f(y^2-\delta e_2)=\frac{5}{4}>f(y^2)$,失败；
- 由于沿e_2的正反向探测均失败，故令$y^3=y^2=\begin{bmatrix}1\\1\end{bmatrix}$
- 比较在y^3和基点 x^2处的函数值，由于$f(y^3)=0<f(x^2)=15\frac{9}{16}$,表明此次模式移动是成功的，因此得到新的基点：

$x^3=y^3=\begin{bmatrix}1\\1\end{bmatrix}$

- 从x^3出发沿方向x^3-x^2进行模式移动。令：
- $y^1=x^3+a(x^3-x^2)=2x^3-x^2=(\frac{1}{2},\frac{3}{2})^T$
- 然后，从y^1出发，进行探测移动。做下去就会发现此次模式移动仍失败。因此退回到基点x^3。减小步长，令：$\delta=\beta\delta=\frac{1}{4}$。
- 再从$y^1=x^3$开始，依次沿e_1,e_2探测。我们还会发现，在x^3周围的探测移动也是失败的，必须继续缩减步长。继续下去，必能得出结论，x^3是局部最优解。
- 事实上，用解析方法容易求得x^3是此问题的最优解。

可行方向法

- 1：计算$\nabla f(x^p),\nabla g_i(x^p), i=1,\dots,m$
- 2：解线性规划问题$\max\limits_{d^p} x_0$

$$\begin{aligned}s.t.\quad & x_0+(\nabla f(x^p),d^p)\le 0\\ & x_0+(\nabla g_i(x^p),d^p)\le -g_i(x^p)\quad i=1,\dots,m\\ & (\nabla g_i(x^p),d^p)\le 0\quad i=1,\dots,m\ (\text{可如此简化})\\ & |d_j^p|\le 1\end{aligned}$$

- 3：当$x_0=0$时停止
- 4：求得合适步长$k_p>0$,并保证 $\min f(x^p+k_pd^p)$

例 7_5

- $\min f(x)=x_1^2+2x_2^2$
- $s.t.\quad x_1+x_2\ge 4$

Xo=(0.85,3.15)

$$\max_{d^p} x_0$$
$$\begin{aligned}s.t.\quad & x_0+(\nabla f(\bar{x}),d)\le 0\\ & (\nabla g(\bar{x}),d)\le 0\\ & |d_1|\le 1,|d_2|\le 1\end{aligned}$$

$\nabla f(x)=\begin{pmatrix}2x_1\\4x_2\end{pmatrix}=\begin{pmatrix}x_1\\2x_2\end{pmatrix}, \nabla f(\bar{x})=\begin{pmatrix}0.85\\6.30\end{pmatrix}$,归一化为：$\begin{pmatrix}0.134\\0.991\end{pmatrix}$

- $\nabla g(\bar{x})=\begin{pmatrix}-1\\-1\end{pmatrix}$,归一化为：$\begin{pmatrix}-0.707\\-0.707\end{pmatrix}$

$$\max_{d^p} x_0$$
$$\begin{aligned}s.t.\quad & x_0+0.134d_1+0.991d_2\le 0\\ & -0.707d_1-0.707d_2\le 0\\ & |d_1|\le 1,|d_2|\le 1\end{aligned}$$

可得：$d_2=-0.5$，$d_1=1.\ x_0=0.357$

- 再考查一个点：$\bar{x}=\begin{pmatrix}2.667\\1.333\end{pmatrix}$

$\nabla f(x)=\begin{pmatrix}2x_1\\4x_2\end{pmatrix}=\begin{pmatrix}x_1\\2x_2\end{pmatrix}, \nabla f(\bar{x})=\begin{pmatrix}2.667\\2.666\end{pmatrix}$,归一化为$\begin{pmatrix}0.707\\0.707\end{pmatrix}$

- $\nabla g(\bar{x})=\begin{pmatrix}-1\\-1\end{pmatrix}$,归一化为：$\begin{pmatrix}-0.707\\-0.707\end{pmatrix}$
- $\max\limits_{d^p} x_0$

$$\begin{aligned}s.t.\quad & x_0+0.707d_1+0.707d_2\le 0\\ & -0.707d_1-0.707d_2\le 0\\ & |d_1|\le 1,|d_2|\le 1\\ & d_1=1,\ d_2=-1, x_0=0\end{aligned}$$

刺绣发

"刺绣法"（hemstitching）算法

- $\min f(x),\quad s.t.\quad g(x)\ge 0.$ 已知：x^0,h,δ，求 x^*
- step1.求归一化的 $\nabla f(x),\nabla g(x)$
- step2. $while(\|x^{p+1}-x^p\|>\delta)$

  $if\ x^p$ 在可行域内部，则：$x^{p+1}=x^p-h\nabla f(x^p)$

  $if\ x^p$ 在可行域外部，则：$x^{p+1}=x^p+h\nabla g(x^p)$

例 7_1

- $\min f(x)=x_1^2+2x_2^2$
- $s.t.\quad x_1+x_2\ge 4$
- 设初始点为 （1,4.5），固定步长为1.0
- 归一化后的梯度为：
- $\nabla f(x)=\frac{1}{\sqrt{4x_1^2+16x_2^2}}\begin{bmatrix}2x_1\\4x_2\end{bmatrix}, \nabla g(x)=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$
- $if\ x^p$ 在可行域内部，则：$x^{p+1}=x^p-1.0\nabla f(x^p)$
- $if\ x^p$ 在可行域外部，则：$x^{p+1}=x^p+1.0\nabla g(x^p)$
- (1,4.5)→(0.89,3.5)→(0.76,2.50)→(1.47,3.21)→(1.05,2.30)→(1.76,3.01)→(1.37,2.10)→(2.08,2.81)

改进刺绣发

改进的刺绣法

- 算法在处理出可行域时，同时考虑目标函数优化和可行域偏差。具体实现是：
- $x^{p+1}=x^p+k_pd^p$
- $if\ x^p$ 在可行域内部，则：$d^p=-\frac{\nabla f(x^p)}{\|\nabla f(x^p)\|}$
- $if\ x^p$ 在可行域外部，则：

$$d^p=-\frac{\nabla f(x^p)}{\|\nabla f(x^p)\|}+\sum\nolimits_{g_i(x)\ violate\ at\ x^p}\frac{\nabla g_i(x^p)}{\|\nabla g_i(x^p)\|}$$

- 当两个梯度方向正好相反时，效率会低。

例 7_2

- 写出下面规划问题的刺绣法的递推迭代公式：
- $\min f(x)=x_1^2+x_2^2$
- $s.t.\quad 3x_1+2x_2\ge 6$
- $\quad x_1,x_2\ge 0$
- 起始点：(2,3), 步长为k_p

- $\nabla f(x)=\begin{bmatrix}2x_1\\2x_2\end{bmatrix}, \nabla g(x)=\begin{bmatrix}3\\2\end{bmatrix}$
- $x^{p+1}=\begin{cases}x^p-\frac{k_p}{\sqrt{4x_1^2+4x_2^2}}\begin{bmatrix}2x_1\\2x_2\end{bmatrix}\\ x^p+\frac{k_p}{\sqrt{13}}\begin{bmatrix}3\\2\end{bmatrix}\end{cases}$
- $x^*=\left(\frac{18}{13},\frac{12}{13}\right)$

例 7_3

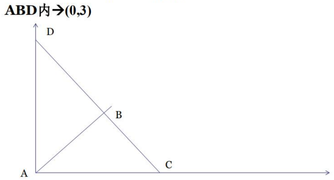
- $\max f(x)=x_1^2+x_2^2$

$s.t.\quad 6-3x_1-2x_2\ge 0$

$\quad x_1,x_2\ge 0$

- 起始点：(0,0)

- $\nabla f(x)=\begin{bmatrix}2x_1\\2x_2\end{bmatrix}$,→$\nabla f(x_0)=\begin{bmatrix}0\\0\end{bmatrix}$
- 若出发点在$x_2=\frac{2}{3}x_1$→$\left(\frac{18}{13},\frac{12}{13}\right)$; 若在ABC内→(2,0); 在ABD内→(0,3)



罚函数算法-内点法

内点罚函数算法步骤

- 1.给定初始点x^0，罚参数$\{K_i\}$，缩小系数δ，精度$\varepsilon>0$，置$\varepsilon_0=1$
- 2.构造罚函数$F(x,K)=f(x)+K\sum_{i=1}^m\frac{1}{g_i\ (x)}$
- 3.用某种无约束非线性规划，以 x^{m-1}为初始点求解 $F(x,K)$
- 4.设最优解为 x^m，若 x^m满足终止条件，结束，否则m++，转2.
- $\{K_i\}$的一般选法：先选定一个初始常数K_1和一个比例系数$\delta<1$,则 $K_i=K_1\delta^{i-1}$

例 7_10

- 用内点法求 $\min f(x)=x_1^2+x_2^2$
- $s.t.\quad x_1+x_2-1\ge 0$

  $2x_1-x_2-2\ge 0$

  $x_1,x_2>0$

- 初始点为[3,1], $K_1=8$，δ=0.5
- 解：$P(x,K)=f(x)+\sum_{i=1}^m K_i/[g_i\ (x)]$

$=x_1^2+x_2^2+k_1/(x_1+x_2-1)+k_1/(2x_1-x_2-2)+k_1/x_1+k_1/x_2$

使用坐标轮换法求解此无约问题得到$x'=\begin{pmatrix}2.76\\1.43\end{pmatrix}$

精度为0.49，继续迭代，令$K_1=K_1\times\delta=4$, 进入下一轮…

罚函数算法-外点法

外点罚函数算法步骤

- 1.给定初始点x^0，罚参数$\{K_i\}$，精度$\varepsilon>0$，置$m=1$
- 2.构造罚函数$F(x,K)=f(x)+K\|g(x)\|^2$
- 3.采用无约束非线性规划，以 x^{m-1}为初始点求解 $F(x,K)$
- 4.设最优解为 x^m，若 x^m满足终止条件，结束，否则$m++$，转2.
- $\{K_i\}$的一般选法：先选定一个初始常数K_1和一个比例系数$\delta\ge 2$,则 $K_i=K_1\delta^{i-1}$
- 终止条件可选$\|x^{m+1}-x^m\|^2$

例 7_8

- 用外点法求 $\min f(x)=\frac{1}{2}x_1^2+\frac{1}{4}x_2^2$
- $s.t.\quad x_1+x_2=1$
- 初始点为[0,0], $K_1=0.05$，δ=2
- 解：$P(x,K)=f(x)+\sum_{i=1}^m K_i[h_i\ (x)]^2$

$$=\frac{1}{2}x_1^2+\frac{1}{4}x_2^2+K_1[x_1+x_2-1]^2$$

使用牛顿法求此无约多维优化问题：$x'=\begin{pmatrix}1/13\\2/13\end{pmatrix}$

$\|x'-x^m\|=0.17$ 循环继续…

$K_1=K_1\delta=0.05\times 2=0.1$ ….

$x^*=\begin{pmatrix}0.329\\0.658\end{pmatrix}$

例 7_9

- 用外点法求 $\min f(x)=x_1^2-x_1x_2+x_2-x_1+1$

$s.t.\quad 2x_1+3x_2-9=1$

$\quad x_1^2+x_2^2-6\ge 0$

$\quad x_1,x_2>0$

- 初始点为[2,2], $K_1=0.05$，δ=2

$$P(x,K)=f(x)+\sum_{i=1}^m K_i[h_i\ (x)]^2+\sum_{i=1}^m K_i[\min(0,g_i\ (x)]^2$$

代码一黄金分割法

```
while b-a>delta
    k=k+1;
    L=0.618*(b-a);
    x1=b-L;
    x2=a+L;
    f_x1=subs(fx,findsym(fx),x1);
    f_x2=subs(fx,findsym(fx),x2);
    if f_x1>f_x2
        a=x1;
    else
        b=x2;
    end
    x0=(a+b)/2;
    f_x0=subs(fx,findsym(fx),x0);
disp(sprintf('k=%d,[%.4f-%.4f],x1=%.2f,x2=%.2f,fx1=%.2f,fx2=%.2f,delta=%.2f,x=%.4f,fx=%.4f\n',k,a,b,x1,x2,eval(f_x1),eval(f_x2),abs(b-a),x0,eval(f_x0)));
end
```

代码一牛顿法

```
while norm(g1) > e
    x0=x0-inv(g2)*g1;
    g1=subs(df,{x1,x2},{x0(1,1),x0(2,1)});
    g2=subs(ddf,{x1,x2},{x0(1,1),x0(2,1)});
    k=k+1;
    fprintf('迭代次数 :%d,x1=%f,x2=%f\n',k,eval(x0(1,1)),eval(x0(2,1)));
end
```

代码一坐标轮换法

```
while norm(x)>e
    x00=x0;
x01=x0+t1*e1;
fx1=(x01(1,1)-2)^4+(x01(1,1)-2*x01(2,1))^2;
df1=diff(fx1,'t1');
t10=vpa(solve(df1),2);
t10=t10(t10==real(t10));
x0=x0+t10*e1;
x02=x0+t2*e2;
fx2=(x02(1,1)-2)^4+(x02(1,1)-2*x02(2,1))^2;
df2=diff(fx2,'t2');
t20=vpa(solve(df2),2);
t20=t20(t20==real(t20));
x0=x0+t20*e2;
x=x0-x00;
k=k+1;
fprintf('迭代次数 :%d,x1=%f,x2=%f\n',k,eval(x0(1,1)),eval(x0(2,1)));
End
```

代码一步长加速法

```
for i=1:maxIter
    if i==1
        X(i,:)=[0,3]; y1=X(i,:);
    end
    fy1=subs(f,{x1,x2},{y1(1),y1(2)});
    y1pe1=y1+delta*e1;
    if subs(f,{x1,x2},{y1pe1(1),y1pe1(2)})<fy1
        y2=y1pe1;
      fy2=subs(f,{x1,x2},{y1pe1(1),y1pe1(2)});
    else
        y2=y1;fy2=fy1;
    end
    y1me1=y2-delta*e1;
    if subs(f,{x1,x2},{y1me1(1),y1me1(2)})<fy2
        y3=y1me1;
fy3=subs(f,{x1,x2},{y1me1(1),y1me1(2)});
    else
        y3=y2;fy3=fy2;
    end
    y1pe2=y3+delta*e2;
    if subs(f,{x1,x2},{y1pe2(1),y1pe2(2)})<fy3
        y4=y1pe2;
      fy4=subs(f,{x1,x2},{y1pe2(1),y1pe2(2)});
    else
        y4=y3;fy4=fy3;
    end
    y1me2=y4-delta*e2;
    if subs(f,{x1,x2},{y1me2(1),y1me2(2)})<fy4
        y5=y1me2;
fy5=subs(f,{x1,x2},{y1me2(1),y1me2(2)});
    else
        y5=y4;fy5=fy4;
    end
    X(i+1,:)=y5;
    y1=X(i+1,:)+alpha*(X(i+1,:)-X(i,:));
    if X(i+1,:)==X(i,:)
        delta=delta*beta;
    end
    k=k+1;
    disp(['探测次数 :',num2str(k),',x1=',num2str(X(i,1)),',x2=',num2str(X(i,2))]);
end
```

代码一斐波那契

```
%先确定 N
F=ones(1,2);
n=2;
while 1
   n=n+1;
   F(n)=F(n-1)+F(n-2);
   if F(n)>=Fn
       break;
   end
end
L1=b-a;
L2=L1*F(n-1)/F(n);
x1=b-L2;
x2=a+L2;
for k=1:n-1
    fx1=subs(fx,findsym(fx),x1);
    fx2=subs(fx,findsym(fx),x2);
    if fx1<fx2
        b=x2;x2=x1;x1=a+b-x2;
    elseif fx1==fx2
        a=x1;b=x2;
    else
        a=x1;x1=x2;x2=b-(x1-a);
    end
     x0=(x1+x2)/2;
    f_x0=subs(fx,findsym(fx),x0);
disp(sprintf('k=%d,[%.2f,%.2f],x=%.4f,fx=%.4f\n',k,a,b,x0,eval(f_x0)));
end
```