方向导数与最速下降方向

设有单位向量 $h=(h_1,h_2,\cdots,h_n)^T\in R^n$ 可微函数 $f(x)$

在 x 点沿 h 方向的方向导数定义为

$$\frac{\partial f(x)}{\partial h}=\lim_{\alpha\to0^+}\frac{f(x+\alpha h)-f(x)}{\alpha}$$
$$=\lim_{\alpha\to0^+}\frac{\nabla f(x)^T(\alpha h)+o(\|\alpha h\|)}{\alpha}$$
$$=\nabla f(x)^T h+\lim_{\alpha\to0^+}\frac{o(\|\alpha h\|)}{\alpha}$$
$$=\nabla f(x)^T h$$
$$=\|\nabla f(x)\|\cos(\nabla f(x),h)$$

凸函数及其性质

设有单位向量 $h=(h_1,h_2,\cdots,h_n)^T\in R^n$ 可微函数 $f(x)$

在 x 点沿 h 方向的方向导数定义为

$$\frac{\partial f(x)}{\partial h}=\lim_{\alpha\to0^+}\frac{f(x+\alpha h)-f(x)}{\alpha}$$
$$=\lim_{\alpha\to0^+}\frac{\nabla f(x)^T(\alpha h)+o(\|\alpha h\|)}{\alpha}$$
$$=\nabla f(x)^T h+\lim_{\alpha\to0^+}\frac{o(\|\alpha h\|)}{\alpha}$$
$$=\nabla f(x)^T h$$
$$=\|\nabla f(x)\|\cos(\nabla f(x),h)$$

性质 3

设有单位向量 $h=(h_1,h_2,\cdots,h_n)^T\in R^n$ 可微函数 $f(x)$

在 x 点沿 h 方向的方向导数定义为

$$\frac{\partial f(x)}{\partial h}=\lim_{\alpha\to0^+}\frac{f(x+\alpha h)-f(x)}{\alpha}$$
$$=\lim_{\alpha\to0^+}\frac{\nabla f(x)^T(\alpha h)+o(\|\alpha h\|)}{\alpha}$$
$$=\nabla f(x)^T h+\lim_{\alpha\to0^+}\frac{o(\|\alpha h\|)}{\alpha}$$
$$=\nabla f(x)^T h$$
$$=\|\nabla f(x)\|\cos(\nabla f(x),h)$$

性质 4

设 $f(x)$ 是二阶可微的，则 $f(x)$ 在开凸集 R 上为凸函数的充分必要条件是：对一切 $x\in R$，$Hesse$ 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。

最优解的判定

充分：梯度=0，海赛矩阵>0

必要：梯度=0，海赛矩阵>=0

满足充要为非奇异点，必要不充分为非奇异点

二分法

$x^*\in[a_n,b_n]$ $\quad c_n=(a_n+b_n)/2$

$x_1^n=c_n-\frac{\varepsilon}{2}$ $\quad x_2^n=c_n-\frac{\varepsilon}{2}$

$if\ f(x_1^n)<f(x_2^n)$ → $a_{n+1}=a_n,b_{n+1}=x_2^n.$

$if\ f(x_1^n)>f(x_2^n)$ → $a_{n+1}=x_1^n,b_{n+1}=b_n$

获得新的搜索区间 $[a_{n+1},b_{n+1}]$ 且 $x^*\in[a_{n+1},b_{n+1}]$

例题

$$f(x)=8x^3-2x^2-7x+3$$

$x\in[0,1]$; $x^*=0.63$; $\varepsilon=0.1$; $a^0=0$, $b^0=1$

解：

$c^0=0.5$, $x_1^0=0.45$ $\quad x_2^0=0.55$

$f(0.45)=0.52$ $\quad f(0.55)=-0.124$

Then $a^1=x_1^0=0.45$, $b^1=b^0=1$

$c^1=0.725$, $x_1^1=0.675$ $\quad x_2^1=0.775$

$f(0.675)=-0.17$ $\quad f(0.775)=0.076$

Then $a^2=a^1=0.45$, $b^2=x_2^1=0.775$

区间等分法

2 点等分法

$$x_1^n=a_n+\frac{1}{3}(b_n-a_n)$$

$$x_2^n=a_n+\frac{2}{3}(b_n-a_n)$$

If $f(x_1^n)<f(x_2^n)$ → $a_{n+1}=a_n,b_{n+1}=x_2^n.$

If $f(x_1^n)>f(x_2^n)$ → $a_{n+1}=x_1^n,b_{n+1}=b_n$

$L_{n+1}=\frac{2}{3}L_n$,每次检查 2 个点

3 点等分法

$$x_1^n=a_n+\frac{1}{4}(b_n-a_n)$$

$$x_2^n=a_n+\frac{2}{4}(b_n-a_n)$$

$$x_3^n=a_n+\frac{3}{4}(b_n-a_n)$$

$if\ f(x_1^n)=minf(x_i^n)$ → $a_{n+1}=a_n,b_{n+1}=x_2^n.$

$if\ f(x_2^n)=minf(x_i^n)$ → $a_{n+1}=x_1^n,b_{n+1}=x_3^n$

$if\ f(x_3^n)=minf(x_i^n)$ → $a_{n+1}=x_2^n,b_{n+1}=b_n$

斐波那契法

1、 初始搜索区间 $L_1=b_0-a_0$

2、计算下一（第二）搜索区间 $L_2=\frac{F_{n-1}L_1}{F_n}+\frac{(-1)^n\varepsilon}{F_n}$

3、更新 x_1 与 x_2

4、计算两点函数值，比较后确定新区间 a_1 与 b_1

5、是否迭代到 n

精度关系： $L_n=\frac{L_1+F_{n-2}\varepsilon}{F_n}$

例题

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 $t^*=0.5$，$f(t^*)=1.75$

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

$L_n=\frac{L_1}{F_n}\le0.5$ → $F_n\ge\frac{3-(-1)}{0.5}=8$ → 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

$x_1^1=b_0-L_2=3-2.5=0.5$

$x_2^1=a_0+L_2=-1+2.5=1.5$

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

更新区间 $a_1=a_0=-1$, $b_1=x_2^1=1.5$

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

…………

0618 法

渐近收敛率：$\lim_{n\to\infty}\frac{L_n}{L_{n-1}}$

$$F_n=\frac{1}{\sqrt5}\left[\left(\frac{1+\sqrt5}{2}\right)^{n+1}-\left(\frac{1-\sqrt5}{2}\right)^{n+1}\right]$$

证明：对于黄金分割法：$\frac{L_n}{L_{n-1}}=\lambda=0.618$ → $\lim_{n\to\infty}\frac{L_n}{L_{n-1}}=0.618$

对于 fibonacci：$\frac{L_n}{L_{n-1}}=\frac{F_{n-1}}{F_n}=\frac{\left(\frac{1+\sqrt5}{2}\right)^n-\left(\frac{1-\sqrt5}{2}\right)^n}{\left(\frac{1+\sqrt5}{2}\right)^{n+1}-\left(\frac{1-\sqrt5}{2}\right)^{n+1}}$

$\lim_{n\to\infty}\frac{L_n}{L_{n-1}}=\frac{\left(\frac{1+\sqrt5}{2}\right)^n}{\left(\frac{1+\sqrt5}{2}\right)^{n+1}}=0.618$

例题

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.

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

$x_2^1=a_0+0.618\times(b_0-a_0)=1.472$

$f(x_1^1)=1.7508$, $f(x_2^1)=2.6948$ → $f(x_2^1)>f(x_1^1)$

更新区间 $a_1=a_0=-1$

$b_1=x_2^1=1.472$

更新点 $\quad x_1^2=a_1+0.382\times(b_1-a_1)=-0.0557$

$x_2^2=x_1^1=0.528$

……………………

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

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

进退法

1、给定初始点 x^0，初始步长 h^0

2、考察 $f(x^0)$，$f(x^0+h)$

3、 $\qquad if\quad f(x^0+h)>f(x^0)$

后退一步计算 $f(x^0-\lambda h)$，$0<\lambda<1$

Untl $f(x^0-\hat\lambda h))>f(x^0)$

→ $x^*\in[x^0-\hat\lambda h,x^0+h]$

$\qquad if\quad f(x^0+h)<f(x^0)$

前进一步计算 $f(x^0+\lambda h)$，$\lambda>1$

Until $\quad f(x^0+\hat\lambda h))>f(x^0+h)$

→ $x^*\in[x^0,x^0+\hat\lambda h]$

多项式插值法

假设搜索起点 $x^0=0$,方向 $d=1$

计算 $g(0),g(1)$；

$if\ g(1)>g(0)$ 计算 $g(\lambda)$,

$$\lambda=\frac{1}{2},\frac{1}{4},\frac{1}{8}\dots until\ g(\lambda)<g(0)$$

设 $a=0,b=\lambda,c=2\lambda$, 根据二次式插值计算：

$$\hat\lambda=(手写)$$

$if\ g(1)<g(0)$ 计算 $g(\lambda),\lambda=2,4,8\dots a,b,c,until\ g(c)>g(b)$，根据二次式插值计算：

$if\ g(\hat\lambda)<g(b)$ → $\lambda_1=\hat\lambda$

$if\ g(\hat\lambda)\ge g(b)$ → $\lambda_1=b$

例题

求 $f(x)=8x^3-2x^2-7x+3$ 的近似极小点，$x^0=0$，$d=1$

$$g(\lambda)=f(x^0+\lambda d)=f(\lambda)=8\lambda^3-2\lambda^2-7\lambda+3$$

$$g(0)=3,g(1)=2$$

$$\because g(0)>g(1)$$

$\therefore\ \lambda=2$, $g(2)=45>g(1)$

$$a=0,b=1,c=2$$

$$\hat\lambda=0.52$$

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 $d^p=-\nabla f(x^p)$，$\overline{d^p}=-\frac{\nabla f(x^p)}{\|\nabla f(x^p)\|}$

进行一维搜索 $\min_k f(x^p+kd^p)$

$$x^{p+1}=x^p+kd^p$$

精度判断为 $\|d^p\|\le\varepsilon$

例题

求 $minf(x)=(x_1-2)^4+(x_1-2x_2)^2$,

$x^0=(0.00,3.00)^T$,$\varepsilon=0.1$

解：

$$\nabla f(x)=\begin{pmatrix}4(x_1-2)^3+2(x_1-2x_2)\\-4(x_1-2x_2)\end{pmatrix}$$

$$\nabla f(x^0)=\begin{pmatrix}-44\\24\end{pmatrix}$$

$$d^0=-\nabla f(x^0)=\begin{pmatrix}44\\-24\end{pmatrix}$$

$$x^1=x^0+k^0d^0=\begin{pmatrix}0\\3\end{pmatrix}+k^0\begin{pmatrix}44\\-24\end{pmatrix}=\begin{pmatrix}44k^0\\-24k^0+3\end{pmatrix}$$

$$minf(x^1)=(44k^0-2)^4+(44k^0-2(-24k^0+3))^2$$

$$\widehat{k^0}=0.062$$

$$x^1=\begin{pmatrix}44\times0.062\\-24\times0.062+3\end{pmatrix}=\begin{pmatrix}2.728\\1.512\end{pmatrix}$$

$$\vdots$$

当进行第七次迭代时，$x^7=\begin{pmatrix}2.28\\1.15\end{pmatrix}$，此时 $\|d^7\|=0.09<0.1$ $\quad$ 满足要求。

梯度法对于二次函数的讨论

例题 求 $minf(x)=\frac{1}{2}(x_1^2+2x_2^2)$，$x^0=(4,4)^T$

$$Q=\begin{pmatrix}1&0\\0&2\end{pmatrix}$$

$$k^*=-[\frac{x^TQ^2x}{x^TQ^3x}]_{x^p}$$

$$k^*=-\frac{5}{9}$$

$$x^1=x^0+k\nabla f(x^0)=\begin{pmatrix}4\\4\end{pmatrix}-\frac{5}{9}\begin{pmatrix}4\\8\end{pmatrix}=\begin{pmatrix}\frac{16}{9}\\-\frac{4}{9}\end{pmatrix}$$

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 $x^{k+1}=x^k-\nabla^2f(x^k)^{-1}\nabla f(x^k)$ ，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。

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

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

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

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

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

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 $d^p=-[\nabla^2f(x^p)]^{-1}\nabla f(x^p)$

进行一维搜索 $\min_k f(x^p+kd^p)$

$$x^{p+1}=x^p+kd^p$$

精度判断为 $\|d^p\|\le\varepsilon$

例题

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

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

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

$\nabla f(x^0)=\begin{pmatrix}4\\100\end{pmatrix}$，$[\nabla^2f(x)]^{-1}=\begin{pmatrix}\frac12&0\\1&\frac{1}{50}\end{pmatrix}$

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

$$d^0=-[\nabla^2f(x^0)]^{-1}\nabla f(x^0)=-\begin{pmatrix}\frac12&0\\0&\frac{1}{50}\end{pmatrix}\begin{pmatrix}4\\100\end{pmatrix}$$

$$=-\begin{pmatrix}2\\2\end{pmatrix}$$

$\min_k f(x^0+kd^0)=(2-2k)^2+25\times(2\text{-}2k)^2=26\times(2-2k)^2$

$\frac{df}{dk}=-104(2-2k)=0$ → $k=1$

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

$\|\nabla f(x^1)\|=0$，故达最优。

FR 共轭梯度法

1：选取初始点 x^0，初始方向 $v^0=-\nabla f(x^0)$

2：for i=1,⋯,n-1

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

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

3：x^0=x^n ，回 step 1.

例题

$minf(x_1,x_2)=\frac32x_1^2+\frac12x_2^2-x_1x_2-2x_1$，$x^0=(-2,4)$

解：第 1 次迭代

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

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

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

$$=(-2+12\lambda,4-6\lambda)^T$$

$f(x^1)=\frac32(-2+12\lambda)^2+\frac12(4-6\lambda)^2-(-2+12\lambda)(4-6\lambda)-2(-2+12\lambda)$

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

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

第 2 次迭代

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

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

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

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

$$=(-\frac{90}{289},\frac{210}{289})^T$$

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

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

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

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

已达最优。

变尺度法（DFP 拟牛顿法的一种）

$Step1$,给定初始点 $x^{(k)}\in E^n$,允许误差 $\varepsilon>0$,

$Step2$,置 $H_1=I_n$,计算出在 $x^{(1)}$ 处的梯度

$$g_1=\nabla f(x^{(1)})$$

置 $k=1$

$Step3$,令 $\quad d^{(k)}=-H_kg_k$

$Step4$,从 $x^{(k)}$ 出发,沿 $d^{(k)}$ 进行一维搜索,求 λ_k,使

$$f(x^{(k)}+\lambda_kd^{(k)})=\min_{\lambda\ge0}f(x^{(k)}+\lambda d^{(k)})$$

令 $x^{(k+1)}=x^{(k)}+\lambda_kd^{(k)}$,

$Step5$,检验是否满足收敛准则,若

$$\|\nabla f(x^{(k+1)})\|\le\varepsilon,$$

停止,得 $\bar x=x^{(k+1)}$,否则,转 $Step6$

$Step6$,若 $k=n$,则令 $x^{(1)}=x^{(k+1)}$,转 $Step2$;否则,转 $Step7$

$Step7$,令 $g_{k+1}=\nabla f(x^{(k+1)}),p^{(k)}=x^{(k+1)}-x^{(k)},q^{(k)}=g_{k+1}-g_k$,

由公式(10.4.18)计算 H_{k+1},置 $k:=k+1$,转 $Step3$

例题

例1用DFP方法求解下列问题

$$\min\ \ 2x_1^2+x_2^2-4x_1+2$$

初始点及初始矩阵分别为

$$x^{(1)}=\begin{pmatrix}2\\1\end{pmatrix},H_1=\begin{pmatrix}1&0\\0&1\end{pmatrix}$$

在点 $x=(x_1,x_2)^T$ 的梯度

$$g=\begin{pmatrix}4(x_1-1)\\2x_2\end{pmatrix}$$

第1次迭代

在点 $x^{(1)}$ 处的梯度 $g_1=\begin{pmatrix}4\\2\end{pmatrix}$

令搜索方向

$$d^{(1)}=-H_1g_1=\begin{pmatrix}-4\\-2\end{pmatrix}$$

从 $x^{(1)}$ 出发,沿方向 $d^{(1)}$ 进行一维搜索,求步长 λ_1:

$$\min_{\lambda\ge0}f(x^{(1)}+\lambda d^{(1)})$$

得到 $\quad \lambda_1=\frac{5}{18}$

令

$$x^{(2)}=x^{(1)}+\lambda_1d^{(1)}=\begin{bmatrix}2\\1\end{bmatrix}+\frac{5}{18}\begin{bmatrix}-4\\-2\end{bmatrix}=\begin{bmatrix}8/9\\4/9\end{bmatrix}$$

$$g_2=\begin{bmatrix}4(\frac89-1\\2\frac49\end{bmatrix}=\begin{bmatrix}-4/9\\8/9\end{bmatrix}$$

第2次迭代 $\quad \Delta x^1=\lambda_1d^{(1)}=\begin{pmatrix}-10/9\\-5/9\end{pmatrix}$

$$\Delta G^1=g_2-g_1=\begin{pmatrix}-40/9\\-10/9\end{pmatrix}$$

计算 H_2

$$H_2=H_1+\Delta H_1$$

$$=H_1+\frac{\Delta X^1\cdot(\Delta X^1)^T}{(\Delta G^1)^T\cdot\Delta X^1}-\frac{H_1\cdot\Delta G^1\cdot(\Delta G^1)^T\cdot H_1}{(\Delta G^1)^T\cdot H_1\cdot\Delta G^1}$$

$$=\begin{pmatrix}1&0\\0&1\end{pmatrix}+\frac{\begin{pmatrix}-10/9\\-5/9\end{pmatrix}\begin{pmatrix}-10/9&-5/9\end{pmatrix}}{\begin{pmatrix}-10/9&-5/9\end{pmatrix}\begin{pmatrix}-40/9\\-10/9\end{pmatrix}}+\frac{\begin{pmatrix}1&0\\0&1\end{pmatrix}\begin{pmatrix}-40/9\\-10/9\end{pmatrix}\begin{pmatrix}-40/9&-10/9\end{pmatrix}\begin{pmatrix}1&0\\0&1\end{pmatrix}}{\begin{pmatrix}-40/9&-10/9\end{pmatrix}\begin{pmatrix}1&0\\0&1\end{pmatrix}\begin{pmatrix}-40/9\\-10/9\end{pmatrix}}$$

$$=\begin{pmatrix}1&0\\0&1\end{pmatrix}+\frac{1}{18}\begin{pmatrix}4&2\\2&1\end{pmatrix}-\frac{1}{17}\begin{pmatrix}16&4\\4&1\end{pmatrix}$$

$$=\frac{1}{306}\begin{pmatrix}86&-38\\-38&305\end{pmatrix}$$

令

$$d^{(2)}=-H_2g_2=-\frac{1}{306}\begin{pmatrix}86&-38\\-38&305\end{pmatrix}\begin{pmatrix}-4/9\\8/9\end{pmatrix}=\frac{12}{51}\begin{pmatrix}1\\-4\end{pmatrix}$$

从$x^{(2)}$出发,沿方向$d^{(2)}$进行一维搜索,求步长λ_2:

$$\min_{\lambda\ge 0} f(x^{(2)}+\lambda d^{(2)})$$

得到

$$\lambda_2=\frac{17}{36}$$

令

$$x^{(3)}=x^{(3)}+\lambda_2 d^{(2)}=\begin{bmatrix}1\\0\end{bmatrix}$$

$$g_3=\nabla f(x^{(3)})=\begin{bmatrix}0\\0\end{bmatrix}$$

于是得最优解

$$(x_1,x_2)=(1,0)$$

坐标轮换法

例题

用坐标轮换法求$\boldsymbol{minf(x)=(x_1-2)^4+(x_1-2x_2)^2}$

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

解：取向量$\boldsymbol{e_1}=(1,0)^{\boldsymbol{T}}$，$\boldsymbol{e_2}=(0,1)^{\boldsymbol{T}}$

$$\boldsymbol{x^0+\lambda_1e_1=(\lambda_1,3)^T}$$

$$\boldsymbol{f(x^0+\lambda_1e_1)}=[\boldsymbol{\lambda_1}-2]^4+[\boldsymbol{\lambda_1}-6]^2$$

求 $\boldsymbol{\min_{\lambda} f(x^0+\lambda e_1)}$

$\frac{df(x^0+\lambda_1e_1)}{d\lambda_1}=0 \quad \rightarrow \quad \boldsymbol{\lambda_1}=3.13$

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

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

$$\boldsymbol{x_1^{(2)}=x_1^{(1)}+\lambda_2e_2}=\begin{bmatrix}3.13\\3+\boldsymbol{\lambda_2}\end{bmatrix}$$

$$\boldsymbol{f\left(x_2^{(1)}+\lambda_2e_2\right)}=(1.13)^4+(3.13-6-2\boldsymbol{\lambda_2})^2$$

求 $\boldsymbol{\min_{\lambda} f(x_2^{(1)}+\lambda_2e_2)}$

$\frac{df\left(x_2^{(1)}+\lambda_2e_2\right)}{d\lambda_2}=0 \quad \rightarrow \quad \boldsymbol{\lambda_2}=-1.44$

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

步长加速法

1、给定初始步长$\boldsymbol{\delta}>0$，加速因子$\boldsymbol{\alpha}\ge 1$，缩减率$\boldsymbol{\beta}\in(0,1)$，允许误差$\boldsymbol{\varepsilon}>0$及初始迭代点$\boldsymbol{x^1}$，置$\boldsymbol{y^1=x^1},\boldsymbol{k}=1,\boldsymbol{j}=1$；

2、如果$\boldsymbol{f\left(y^j+\delta e_j\right)<f(y^j)}$，令$\boldsymbol{y^{j+1}=y^j+\delta e_j}$，转第 4 步； 否则转第 3 步。

3、如果$\boldsymbol{f\left(y^j-\delta e_j\right)<f(y^j)}$，令$\boldsymbol{y^{j+1}=y^j-\delta e_j}$，转第 4 步；否则，令$\boldsymbol{y^{j+1}=y^j}$，转第 4 步。

4、如果 $\boldsymbol{j<n}$，则置$\boldsymbol{j=j+1}$，转第 2 步；否则，转第 5 步；

5、如果$\boldsymbol{f(y^{n+1})<f(x^k)}$，则转第 6 步；否则，转第 7 步；

6、令$\boldsymbol{x^{k+1}=y^{n+1}}$，$\boldsymbol{y^1=x^{k+1}+\alpha(x^{k+1}-x^k)}=(1+\boldsymbol{\alpha})\boldsymbol{x^{k+1}-x^k}$，置$\boldsymbol{k=k}+1,\mathbf{j}=1$,转第 2 步；

7、$\boldsymbol{\delta=\beta\delta}$，$\boldsymbol{y^1=x^k,x^{k+1}=x^k}$，置$\boldsymbol{k=k}+1,\boldsymbol{j}=1$，转第 2 步。

例题

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

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

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

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

$\boldsymbol{f(y^1+\delta e_1)}=197\frac{9}{16}>\boldsymbol{f(y^1)}$,故失败。

$$\boldsymbol{y^1-\delta e_1}=\begin{bmatrix}2\\0\end{bmatrix}+\frac{1}{2}\begin{bmatrix}1\\0\end{bmatrix}=\begin{bmatrix}\frac{3}{2}\\0\end{bmatrix}$$

$\boldsymbol{f(y^1-\delta e_1)}=25\frac{9}{16}<\boldsymbol{f(y^1)}$,成功。

因此，令$\boldsymbol{y^2=y^1-\delta e_1}=\begin{bmatrix}\frac{3}{2}\\0\end{bmatrix}$，从$\boldsymbol{y^2}$出发，沿$\boldsymbol{e_2}$探测情况如下：

$\boldsymbol{y^2+\delta e_2}=\begin{bmatrix}\frac{3}{2}\\0\end{bmatrix}+\frac{1}{2}\begin{bmatrix}0\\1\end{bmatrix}=\begin{bmatrix}\frac{3}{2}\\\frac{1}{2}\end{bmatrix}$,

$\boldsymbol{f(y^2+\delta e_2)}=15\frac{9}{16}<\boldsymbol{f(y^2)}$,成功。

因此，令$\boldsymbol{y^3=y^2+\delta e_2}=\begin{bmatrix}\frac{3}{2}\\\frac{1}{2}\end{bmatrix}$.

第一轮探测完成后，由于$\boldsymbol{f(y^3)<f(y^1)}$,因此得到第 2 个基点$\boldsymbol{x^2=y^3}=\begin{bmatrix}\frac{3}{2}\\\frac{1}{2}\end{bmatrix}$,再沿方向$\boldsymbol{x^2-x^1}$进行模式移动，令：

$$\boldsymbol{y^1=x^2+\alpha(x^2-x^1)}=2\boldsymbol{x^2-x^1}=2\begin{bmatrix}\frac{3}{12}\\\frac{1}{2}\end{bmatrix}-\begin{bmatrix}2\\0\end{bmatrix}=\begin{bmatrix}1\\1\end{bmatrix}$$

模式移动后，立即从得到的点$\boldsymbol{y^1}$出发，进行第 2 轮探测移动。探测情况如下：

……………………

Powell 算法

设有一组线性独立的向量$\{\boldsymbol{v^i,i=1,\dots,n}\}$，$\boldsymbol{x^0}$为初始点。算法步骤为：

1：for i=1,...,n, 找到 $\boldsymbol{\min_{\lambda} f(x^{i-1}+\lambda v^i)}$,的 $\boldsymbol{\lambda}$值，递推下一点$\boldsymbol{x^i=x^{i-1}+\lambda v^i}$

2：for i=1,...,n-1, 使 $\boldsymbol{v^i=v^{i+1}}$

3： $\boldsymbol{v^n=x^n-x^0}$

4： 找到 $\boldsymbol{\min_{\lambda} f(x^n+\lambda(x^n-x^0))}$,的 $\boldsymbol{\lambda}$值，替换 $\boldsymbol{x^0=x^n+\lambda(x^n-x^0)}$

5：达不到精度回到 step1

例题

$f(x)=x_1^2+2x_2^2 \quad v^1=[1,-1]^T, \quad v^2=[1,1]^T, \quad x^0=[20,20]^T$

$\lambda_1=6.66, x^1=x^0+\lambda_1v^1=[26.66,13.34]^T$

$\lambda_2=-17.8, x^2=x^1+\lambda_2v^2=[8.86,-4.46]^T$

$u^1=x^2-x^0=[-11.14,-24.46]^T$

$\lambda_3=-0.15, x^3=x^2+\lambda_3u^1=[10.48,-0.92]^T$

$\lambda_4=-3.46, x^4=x^3+\lambda_4v^2=[7.02,-4.18]^T$

$\lambda_5=-0.15, x^5=x^4+\lambda_5u^1=[8.68,-0.64]^T$

$u^2=x^5-x^3=[-1.8,0.28]^T$

$$(u^1,Au^2)=[-11.14,-24.46]\begin{bmatrix}2&0\\0&4\end{bmatrix}\begin{bmatrix}-1.8\\0.28\end{bmatrix}=3\neq 0$$

拉格朗日函数法（等式约束）

原问题模型

$$\boldsymbol{minf(x)}$$

$$\boldsymbol{s.t.} \quad \boldsymbol{g_i(x)}=0 \quad \forall\boldsymbol{i}=1,2,\dots,\boldsymbol{m<n}$$

拉格朗日函数

$$\boldsymbol{L(x,\lambda)=f(x)+\sum_{i=1}^{m}\lambda_ig_i(x)}$$

必要性条件

$$\boldsymbol{L(x,\lambda)=f(x)+\sum_{i=1}^{m}\lambda_ig_i(x)}$$

$$\frac{\partial L}{\partial x_j}=\frac{\partial L}{\partial x_j}+\sum_{i=1}^{m}\lambda_i\frac{\partial g_i}{\partial x_j}=0 \qquad \boldsymbol{j}=1,2,\dots,\boldsymbol{n}$$

$$\frac{\partial L}{\partial \lambda_i}=\boldsymbol{g_i(x)}=0 \qquad \boldsymbol{i}=1,2,\dots,\boldsymbol{m}$$

以上条件成立基础是 **无约束问题存在最优解**。

存在性条件

此解存在的条件是：$\left[\frac{\partial g}{\partial x}\right]$的秩为 m

拉格朗日函数法（不等式约束）

$$\boldsymbol{minf(x)}$$

$$\boldsymbol{s.t.} \quad \boldsymbol{h_j(x)}\ge 0 \quad \boldsymbol{j}=1,2,\dots,\boldsymbol{m<n}$$

基本思路 :将不等式约束问题转化为等式约束问题。

使用松驰变量$\boldsymbol{\theta_j^2=h_j(x)}\ge 0$

$$\boldsymbol{L(x,\lambda,\theta)=f(x)+\sum_{j=1}^{m}\lambda_j(h_j(x)-\theta_j^2)}$$

$$\frac{\partial L}{\partial x_i}=\frac{\partial f}{\partial x_i}+\sum_{j=1}^{m}\lambda_j\frac{\partial h_j(x)}{\partial x_i}=0 \qquad i=1,\dots,n$$

$$\frac{\partial L}{\partial \lambda_j}=\boldsymbol{h_j(x)}-\boldsymbol{\theta_j^2}=0 \qquad j=1,\dots,m$$

$$\frac{\partial L}{\partial \theta_j}=-2\lambda_j\theta_j=0 \qquad j=1,\dots,m$$

三种情况

情形 I：$\boldsymbol{\lambda_j^*}=0$且 $\boldsymbol{\theta_j^*}\neq 0$，故$\boldsymbol{h_j(x^*)=(\theta_j^*)^2}>0$,故最优解在可行域内部。若所有 $\boldsymbol{\lambda_j^*}=0$，等价于无约束情形。

情形 II：$\boldsymbol{\lambda_j^*}\neq 0$且 $\boldsymbol{\theta_j^*}=0$，故 $\boldsymbol{h_j(x^*)}$=0， 故最优解在可行域边界。由于 $\boldsymbol{\lambda_j^*}\neq 0$，故不满足$\boldsymbol{\nabla f(x^*)}=0$.

情形 III：$\boldsymbol{\lambda_j^*}=0$且 $\boldsymbol{\theta_j^*}=0$，故 $\boldsymbol{h_j(x^*)}$=0,且$\boldsymbol{\nabla f(x^*)}=0$，故约束条件的边界穿过无约束问题的最优解。

例题 $\quad \boldsymbol{min \;\; f(x)=(x-a)^2+b}$

$$\boldsymbol{s.t \quad x\ge c}$$

令 $\boldsymbol{\theta^2=x-c\ge 0}$,得到：

$$\boldsymbol{L=(x-a)^2+b+\lambda(x-c-\theta^2)}$$

$$\frac{\partial L}{\partial x}=2(\boldsymbol{x-a})+\boldsymbol{\lambda}=\mathbf{0}$$

$$\frac{\partial L}{\partial \lambda}=\boldsymbol{x-c-\theta^2}=\mathbf{0}$$

$$\frac{\partial L}{\partial \theta}=-2\boldsymbol{\lambda\theta}=\mathbf{0}$$

$$\mathbf{2}(\boldsymbol{x-a})+\boldsymbol{\lambda}=\mathbf{0}$$

$$\boldsymbol{x-c-\theta^2}=\mathbf{0}$$

情形 I :$\boldsymbol{\lambda=0}$，$\boldsymbol{x^*=a}$，$\boldsymbol{\theta^2=a-c}$.若 $\boldsymbol{c<a}$，$\boldsymbol{\theta}$为实值.

情形 II： $\boldsymbol{\theta=0}$，$\boldsymbol{x^*=c}$，$\boldsymbol{\lambda^*=-2(c-a)}$

情形 III： $\boldsymbol{\lambda=0}$， $\boldsymbol{\theta=0}$， $\boldsymbol{x^*=a=c}$

库恩-塔克条件（KT）（等式约束）

首先定义一个 类拉格朗日函数$\boldsymbol{L(x,\lambda)=f(x)+\sum_{i=1}^{r}\lambda_ig_i(x)}$

以下四组条件在x^*点需满足：

$$\begin{cases}\boldsymbol{\frac{\partial f}{\partial x_j}(x^*)+\sum_{i=1}^{r}\lambda_i\frac{\partial g_i}{\partial x_j}(x^*)=0}\\ \boldsymbol{g_i(x^*)\le 0}\\ \boldsymbol{(\lambda_i^*)g_i(x^*)=0}\\ \boldsymbol{\lambda_i^*\ge 0}\end{cases}$$

简洁的表示：
$$\begin{cases}\boldsymbol{\nabla_xL(x^*,\lambda^*)=0}\\ \boldsymbol{\nabla_\lambda L(x^*,\lambda^*)\le 0}\\ \boldsymbol{(\lambda^*)^Tg(x^*)=0}\\ \boldsymbol{\lambda^*\ \ge 0}\end{cases}$$

KT 条件为必要条件，当目标函数为凸时，为充要条件。

例题 利用 KT 条件求：

$$\boldsymbol{\min_x(x-a)^2+b}$$

$$\boldsymbol{s.t.} \quad \boldsymbol{x\ge c}$$

$$\boldsymbol{L(x,\lambda)=(x-a)^2+b+\lambda(c-x)}$$

$$\frac{\partial L}{\partial x}=\mathbf{2}(\boldsymbol{x-a})-\boldsymbol{\lambda}=\mathbf{0}$$

$$\frac{\partial L}{\partial \lambda}=\boldsymbol{c-x\le 0}$$

$$\boldsymbol{\lambda g(x^*)=\lambda(c-x)=0}$$

$$\boldsymbol{\lambda\ge 0}$$

情形 I：$\boldsymbol{\lambda=0}$，→ $\boldsymbol{x=a}$,且 $\boldsymbol{c\le a}$

情形 II： $\boldsymbol{x=c}$，→ $\boldsymbol{c>a}$

情形 III： $\boldsymbol{x=c=a}$

库恩-塔克条件（KT）（不等式约束）

$$\boldsymbol{\min_x f(x)}$$

$$\boldsymbol{s.t.} \quad \boldsymbol{h_i(x)=0} \quad \boldsymbol{i=1,2,\dots,r}$$

$$\boldsymbol{g_j(x)\le 0} \quad \boldsymbol{i=1,2,\dots,n}$$

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

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

刺绣法

$minf(x)$, $\quad s.t.\ g(x)\ge 0.$ 　　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：$x^{p+1}=x^p-1.0\nabla f(x^p)$

if x^p 在可行域外部，则：$x^{p+1}=x^p+1.0\nabla g(x^p)$

例题 $\qquad \boldsymbol{minf(x)=x_1^2+2x_2^2}$

$$\boldsymbol{s.t.} \quad \boldsymbol{x_1+x_2\ge 4}$$

设初始点为 （1,4.5)，固定步长为 1.0

设初始点为 （1,4.5)，固定步长为 1.0

归一化后的梯度为：

$$\boldsymbol{\nabla f(x)}=\frac{1}{\sqrt{4x_1^2+16x_2^2}}\begin{bmatrix}2x_1\\4x_2\end{bmatrix},\boldsymbol{\nabla g(x)}=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$$

if $\boldsymbol{x^p}$ 在可行域内部，则： $\boldsymbol{x^{p+1}=x^p}-1.0\boldsymbol{\nabla f(x^p)}$

if $\boldsymbol{x^p}$ 在可行域外部，则： $\boldsymbol{x^{p+1}=x^p}+1.0\boldsymbol{\nabla g(x^p)}$

(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)

改进的刺绣法

$$\boldsymbol{x^{p+1}=x^p+k_pd^p}$$

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

if $\boldsymbol{x^p}$ 在可行域外部，则：

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

例题 $\qquad \boldsymbol{x^{p+1}=x^p+k_pd^p}$

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

if $\boldsymbol{x^p}$ 在可行域外部，则：

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

$$\boldsymbol{\nabla f(x)}=\begin{bmatrix}\boldsymbol{2x_1}\\\boldsymbol{2x_2}\end{bmatrix},\ \boldsymbol{\nabla g(x)}=\begin{bmatrix}\mathbf{3}\\\mathbf{2}\end{bmatrix}$$

$$\boldsymbol{x^{p+1}}=\begin{cases}\boldsymbol{x^p}-\dfrac{\boldsymbol{k_p}}{\sqrt{\boldsymbol{4x_1^2+4x_2^2}}}\begin{bmatrix}\boldsymbol{2x_1}\\\boldsymbol{2x_2}\end{bmatrix}\\ \boldsymbol{x^p}+\dfrac{\boldsymbol{k_p}}{\sqrt{\mathbf{13}}}\begin{bmatrix}\mathbf{3}\\\mathbf{2}\end{bmatrix}\end{cases}$$

$$\boldsymbol{x^*}=(\frac{\mathbf{18}}{\mathbf{13}},\frac{\mathbf{12}}{\mathbf{13}})$$

坐标轮换法

1.给定初始点$\boldsymbol{x^0}$，初始步长$\boldsymbol{\delta}$，缩放因子$\boldsymbol{u}>1$及精度$\boldsymbol{\varepsilon_1,\varepsilon_2}$，置$\boldsymbol{m}=0$ 　　2.令$\boldsymbol{y=x^k}$

3.从$\boldsymbol{y}$出发，按步长$\boldsymbol{t=\delta_1^0}$ 沿坐标轴$\boldsymbol{e^1}$的正方向搜索，令$\boldsymbol{y^1=y+te^1}$，如果$\boldsymbol{y^1}$在可行域内，即满足$\boldsymbol{g(y^1)\ge 0}$，且有 $\boldsymbol{f(y^1)<f(y)}$,则取$\boldsymbol{t=2t}$，加速向前搜索直到不满足可行性条件或函数值下降的条件，转 5.

如果沿坐标轴 $\boldsymbol{e^1}$的正方向找不到同时满足两条件的点，则转向负方向搜索，再转 5.

4.如果沿坐标轴 $\boldsymbol{e^1}$的正方向和负方向都找不到同时满足可行性条件和函数值下降的点，则减小搜索步长 $\boldsymbol{\delta^0=\delta^0/u}$

5.依次沿其他坐标轴进行同样的搜索，最终求得此轮搜索的最优点$\boldsymbol{y^*}$

6.如果$\|\boldsymbol{y^*-x^k}\|\le\boldsymbol{\varepsilon_2}$, 转 7, 否则$\boldsymbol{x^{k+1}=y^*,k=k+1}$,转 2.

7.如果步长$\boldsymbol{max(\delta^0)\le\varepsilon_1}$,则$\boldsymbol{x^*=x^k}$,否则$\boldsymbol{\delta^0=\delta^0/u}$，转 2.

有约束问题梯度算法（可行方向法）

1：计算$\boldsymbol{\nabla f(x^p),\nabla g_i(x^p),i}=1,\dots,\boldsymbol{r}$

2：解线性规划问题$\boldsymbol{\max_{d^p} x_0}$

$$\boldsymbol{s.t.} \quad \boldsymbol{x_0+(\nabla f(x^p),d^p)}\le 0$$

$$\boldsymbol{x_0+(\nabla g_i(x^p),d^p)\le -g_i(x^p)} \quad \boldsymbol{i}=1,\dots,\boldsymbol{r}$$

$$\left|\boldsymbol{d_j^p}\right|\le 1$$

3：当$\boldsymbol{x_0}=0$时停止

4：求得合适步长$\boldsymbol{k_p}>0$,并保证 $\mathbf{min}\boldsymbol{f\left(x^p+k_pd^p\right)}$

例题 $\qquad \boldsymbol{minf(x)=x_1^2+2x_2^2}$

$$\boldsymbol{s.t.} \quad \boldsymbol{x_1+x_2\ge 4}$$

$$\boldsymbol{\max_{d^p} x_0}$$

$$\boldsymbol{s.t.} \quad \boldsymbol{x_0+(\nabla f(\overline{x}),d)}\le 0$$

$$(\boldsymbol{\nabla g(\overline{x}),d})\le 0$$

$$\left|\boldsymbol{d_1}\right|\le 1,\left|\boldsymbol{d_2}\right|\le 1$$

$\boldsymbol{\nabla f(x)}=\begin{pmatrix}\boldsymbol{2x_1}\\\boldsymbol{4x_2}\end{pmatrix}=\begin{pmatrix}\boldsymbol{x_1}\\\boldsymbol{2x_2}\end{pmatrix}$，$\boldsymbol{\nabla f(\overline{x})}=\begin{pmatrix}0.85\\6.30\end{pmatrix}$,归一化为：$\begin{pmatrix}0.134\\0.993\end{pmatrix}$

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

$$\boldsymbol{s.t.} \quad \boldsymbol{x_0}+0.134\boldsymbol{d_1}+0.993\boldsymbol{d_2}\le 0$$

$$-0.707\boldsymbol{d_1}-0.707\boldsymbol{d_2}\le 0$$

$$\left|\boldsymbol{d_1}\right|\le 1,\left|\boldsymbol{d_2}\right|\le 1$$

$$\boldsymbol{x_0}\le -0.134(\boldsymbol{d_1+d_2})-0.859\boldsymbol{d_2}$$

观察可得： $\boldsymbol{d_2}=-1$，$\boldsymbol{d_1}=1.$ $\boldsymbol{x_0}=0.859$

外点罚函数算法

1.给定初始点$\boldsymbol{x^0}$，罚参数$\{\boldsymbol{K_i}\}$，精度$\boldsymbol{\varepsilon>0}$，置$\boldsymbol{m}=1$

2.构造罚函数$\boldsymbol{F(x,K)=f(x)+K\|g(x)\|^2}$

3.采用无约束非线性规划，以 $\boldsymbol{x^{m-1}}$为初始点求解$\boldsymbol{F(x,K)}$

4.设最优解为 $\boldsymbol{x^m}$，若 $\boldsymbol{x^m}$满足终止条件，结束，否则$\boldsymbol{m++}$，转 2.

$\{\boldsymbol{K_i}\}$的一般选法：先选定一个初始常数$\boldsymbol{K_1}$和一个比例系数$\boldsymbol{\delta}\ge 2$,则 $\boldsymbol{K_i=K_1\delta^{i-1}}$

终止条件可选$\|\boldsymbol{x^{m+1}-x^m}\|^2$

内点罚函数法

1.给定初始点$\boldsymbol{x^0}$，罚参数$\{\boldsymbol{K_i}\}$，缩小系数$\boldsymbol{\delta}$，精度$\boldsymbol{\varepsilon}>0$，置$\boldsymbol{m}=1$

2.构造罚函数$\boldsymbol{F(x,K)=f(x)+K\sum_{i=1}^{m}\frac{1}{g_i\ (x)}}$

3.用某种无约束非线性规划，以 $\boldsymbol{x^{m-1}}$为初始点求解$\boldsymbol{F(x,K)}$

4.设最优解为 $\boldsymbol{x^m}$，若 $\boldsymbol{x^m}$满足终止条件，结束，否则$\boldsymbol{m}$++，转 2.

$\{\boldsymbol{K_i}\}$的一般选法：先选定一个初始常数$\boldsymbol{K_1}$和一个比例系数$\boldsymbol{\delta}<1$,则 $\boldsymbol{K_i=K_1\delta^{i-1}}$

多项式插值法
```
syms x;
fx=8*x^3-2*x^2-7*x+3;
fx=exp(x)-5*x;

x0=1;
x1=x0+1;

fx0=subs(fx,findsym(fx),x0);
fx1=subs(fx,findsym(fx),x1);

lamder=1;
if fx1<fx0   %%double lamder
    fxh=fx1;
  while fxh<fx0
    lamder=2*lamder;
    xh=x0+lamder;
    fxh=subs(fx,findsym(fx),xh);
  end

else %%divide lamder
    fxh=fx1;
    while      fxh>fx0
    lamder=lamder/2;
    xh=x0+lamder;
    fxh=subs(fx,findsym(fx),xh);
    end
end

x1=x0;
x2=(x0+xh)/2;
x3=xh;
y1=eval(subs(fx,findsym(fx),x1));
y2=eval(subs(fx,findsym(fx),x2));
y3=eval(subs(fx,findsym(fx),x3));

%%%%
b=((x2^2-x3^2)*(y1-y2)-(x1^2-x2^2)*(y2-y3))/((x2^2-x3^2)*(x1-x2)-(x1^2-x2^2)*(x2-x3));
a=(y1-y2-b*(x1-x2))/(x1^2-x2^2);
c=y1-a*x1^2-b*x1;

x_opt=-b/(2*a);
```

三点等分法
```
syms x;
fx=8*x^3-2*x^2-7*x+3;
a=0;b=1;
delta=0.0001;

x0=a;
f_x0=subs(fx,findsym(fx),x0);
pre_f_x0=999;
k=0;
disp(sprintf('k=%d,x=%.4f,fx=%.4f\n',k,x0,eval(f_x0)));

while       abs(f_x0-pre_f_x0)>delta
      k=k+1;
      x1=a+(b-a)/4;
      x2=a+2*(b-a)/4;
      x3=a+3*(b-a)/4;
      f_x1=subs(fx,findsym(fx),x1);
      f_x2=subs(fx,findsym(fx),x2);
      f_x3=subs(fx,findsym(fx),x3);
      if  f_x1<=f_x2  & f_x1<=f_x3
            b=x2;x0=x1;
      elseif  f_x2<=f_x1 & f_x2<=f_x3
            a=x1;b=x3;x0=x2;
      elseif  f_x3<=f_x1 & f_x3<=f_x2
            a=x2;x0=x3;
      end

pre_f_x0=f_x0;
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
```

两点等分法
```
syms x;
fx=8*x^3-2*x^2-7*x+3;
a=0;b=1;
delta=0.0001;

x0=(a+b)/2;
pre_f_x0=100;
f_x0=subs(fx,findsym(fx),x0);
k=0;
disp(sprintf('k=%d,x0=%.4f,f(x0)=%.4f\n',k,x0,eval(f_x0)));

while         abs(f_x0-pre_f_x0)>=delta
      k=k+1;
      x1=a+(b-a)/3;
      x2=a+2*(b-a)/3;
      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;
      pre_f_x0=f_x0;
      f_x0=subs(fx,findsym(fx),x0);
      disp(sprintf('k=%d,x0=%.4f,f(x0)=%.4f\n',k,x0,eval(f_x0)));
end
```

进退法
```
syms x;
fx=x^4-x^2-2*x+5;
h=0.1;
x0=0;

k=0;
x1=x0;
while 1
      k=k+1;
      x4=x1+h;
      f4=subs(fx,findsym(fx),x4);
      f1=subs(fx,findsym(fx),x1);

      if f4<f1
            x2=x1;f2=f1;
            x1=x4;f1=f4;
            h=2*h;
      else%%fxh>=fx0
            if k==1
                  h=-h;
                  x2=x4;
                  f2=f4;
            else
                  x3=x2;
                  x2=x1;
                  x1=x4;
                  h=h/2;
                  break;
            end
      end
end
a=min(x1,x3)
b=x1+x3-a
```

快速下降法
```
format long;
% syms x1 x2;
% syms k;
%fx=(x1-2).^4+(x1-2*x2).^2;
% var=[x1,x2];
% x0=[0,3];
% delta=0.1;

syms x1 x2 x3 x4;
syms k;
fx=x1^2+x2^2+x3^2+x4^2;
var=[x1,x2,x3,x4];
x0=[2,-2,1,-1];
delta=0.1;

df=-jacobian(fx,var);
d_x0=subs(df,var,x0);
tol=1;
s=0;
disp(sprintf('s=%d,x=(%.4f,%.4f),fx=%.4f\n',s,x0(1),x0(2),eval(Funval(fx,var,x0))));

while    tol>delta    && s<15
      s=s+1;
      d_x0=Funval(df,var,x0);
      tol=vpa(norm(d_x0));
      x1=x0+k*d_x0;

fk=subs(fx,var,x1);
[a,b]=model_fb_search(fk,0,0.1);
k_min=model_golden_search(fk,a,b);

x1=x0+k_min*d_x0;
x0=x1;
%disp(sprintf('norm(d_x0)=%.4f',eval(tol)));
%disp(sprintf('s=%d,x=(%.4f,%.4f),fx=%.8f\n',s,eval(vpa(x0(1))),eval(vpa(x0(2))),eval(vpa(Funval(fx,var,x0)))));

disp(sprintf('s=%d,x=(%.4f,%.4f,%.4f,%.4f),fx=%.8f\n',s,eval(vpa(x0(1))),eval(vpa(x0(2))),eval(vpa(x0(3))),eval(vpa(x0(4))),eval(vpa(Funval(fx,var,x0)))));
      end
```

外点法:
```
syms x1 x2;
fx=x1^2-x1*x2+x2-x1+1;
hx=2*x1+3*x2-9;
gx=[x1^2+x2^2-6,x1,x2];
m=length(gx);
var=[x1,x2];
x0=[2,2];
k1=0.05;
lambda=2;

s=0;
fx_x0=subs(fx,var,x0);
display(sprintf('s=%d,x=[%.4f,%.4f],f(x)=%.4f',s,x0(1),x0(2),eval(fx_x0)));

tol=1;
delta=0.01;
while tol>delta
   pk=fx+k1*(hx)^2;
   for i=1:m
      gx_x0=subs(gx(i),var,x0);
      ifgx_x0<0
         pk=pk+k1*(gx(i))^2;
         end
      end

[xp,minf]=minNewton_Model(pk,x0,var);
      k1=k1*lambda;
      tol=norm(xp-x0);
      x0=xp;
      s=s+1;
  fx_x0=subs(fx,var,x0);
display(sprintf('s=%d,x=[%.4f,%.4f],f(x)=%.4f',s,eval(x0(1)),eval(x0(2)),eval(fx_x0)));
end
```

刺绣法:
```
syms x1 x2;
var=[x1,x2];
f=x1^2+2*x2^2;
g=x1+x2-4;
x0=[1,4.5];
h=1;
delta=0.5;
df=jacobian(f,var);
dg=jacobian(g,var);

tol=1;
s=1;
f_x0=Funval(f,var,x0);
disp(sprintf('s=%d,x=[%.4f,%.4f],fx=%.4f\n',s,x0(1),x0(2),eval(f_x0)));
while    tol>delta    && s<=10
      s=s+1;
g_x0=Funval(g,var,x0);
df_x0=Funval(df,var,x0);
dg_x0=Funval(dg,var,x0);
  if g_x0>=0
      x1=x0-h*df_x0/norm(df_x0);
   else
x1=x0+h*dg_x0/norm(dg_x0);
      end
  tol=norm(x1-x0);
      x0=x1;
f_x0=Funval(f,var,x0);
disp(sprintf('s=%d,x=[%.4f,%.4f],fx=%.4f\n',s,eval(x0(1)),eval(x0(2)),eval(f_x0)));
end
```

改进刺绣法:
```
syms x1 x2;
var=[x1,x2];
f=x1^2+x2^2;
g1=3*x1+2*x2-6;
g2=x1+0*x2;
g3=0*x1+x2;
x0=[2,3];
h=1;
delta=0.1;
df=jacobian(f,var);
dg1=jacobian(g1,var);
dg2=jacobian(g2,var);
dg3=jacobian(g3,var);
s=0;
f_x0=Funval(f,var,x0);
disp(sprintf('s=%d,x=[%.4f,%.4f],fx=%.4f\n',s,x0(1),x0(2),eval(f_x0)));
tol=1;
while tol>delta && s<6
     s=s+1;
     df_x0=Funval(df,var,x0);
     dg1_x0=Funval(dg1,var,x0);
     dg2_x0=Funval(dg2,var,x0);
     dg3_x0=Funval(dg3,var,x0);
     g1_x0=Funval(g1,var,x0);
     g2_x0=subs(g2,findsym(g2),x0(1));
     g3_x0=subs(g3,findsym(g3),x0(2));
     d_x0=-df_x0/norm(df_x0);
     if g1_x0<0
        d_x0=d_x0+g1_x0/norm(g1_x0);
     end
     ifg2_x0<0
        d_x0=d_x0+g2_x0/norm(g2_x0);
     end
     if g3_x0<0
        d_x0=d_x0+g3_x0/norm(g3_x0);
     end
     x1=x0+h*d_x0;
     tol=norm(x1-x0);
     x0=x1;
  f_x0=Funval(f,var,x0);
disp(sprintf('s=%d,x=[%.4f,%.4f],fx=%.4f\n',s,eval(x0(1)),eval(x0(2)),eval(f_x0)));
end
```

有约束问题的梯度算法（可行方向法）
```
syms x1 x2;
syms k;
var=[x1,x2];
n=length(var);
f=x1^2+2*x2^2;
g=4-x1-x2;
df=jacobian(f,var);
dg=jacobian(g,var);
x0=[0.85,3.15];
delta=0.01;
f_x0=Funval(f,var,x0);
s=1;
tol=1;
disp(sprintf('s=%d,x=[%.4f,%.4f],fx=%.4f\n',s,x0(1),x0(2),eval(f_x0)));
while tol>delta
df_x0=Funval(df,var,x0);
dg_x0=Funval(dg,var,x0);
df_x0=df_x0/norm(df_x0);
dg_x0=dg_x0/norm(dg_x0);
C=[-1,zeros(1:n)];
a1=[1,eval(df_x0(1:n))];
a2=[1,eval(dg_x0(1:n))];
a3=[0,1,0];
a4=[0,0,1];
A=[a1;a2;a3;a4];
B=[0;0;1;1];
[x,min]=linprog(C,A,B);
d=x(2:3,:);
tol=abs(min);

x1=x0+k*d';
fk=Funval(f,var,x1);
[a0,b0]=model_fb_search(fk,0,0.1);
[k_min]=model_golden_search(fk,a0,b0);
x0=x0+k_min*d';
f_x0=Funval(f,var,x0);
disp(sprintf('s=%d,x=[%.4f,%.4f],fx=%.4f\n',s,eval(vpa(x0(1))),eval(vpa(x0(2))),eval(vpa(f_x0))));
end
```

变尺度法（DFP）
```
syms x1 x2;
syms k;
var=[x1;x2];
x0=[10;10];
A=[3,-1;-1,1];
fx=0.5*[x1,x2]*A*[x1;x2];

H0=eye(2,2);
df=jacobian(fx,var);
df=df';
df_x0=subs(df,var,x0);

delta=0.1;tol=1;s=0;
display(sprintf('s=%d,x=[%.4f,%.4f]',s,x0(1),x0(2)));
while tol>delta
      s=s+1;
      v0=-H0*df_x0;
      xp=x0+k*v0;
      fk=subs(fx,var,xp);
k_opt=solve(diff(fk),k);
      xp=x0+k_opt*v0;
df_xp=subs(df,var,xp);
      y=df_xp-df_x0;
      u=xp-x0;
      A=(u*u')/(u'*y);
B=(H0*y*(H0*y)')/(y'*H0*y);
H0=H0+A-B;
tol=norm(xp-x0);
df_x0=df_xp;
x0=xp;
display(sprintf('s=%d,x=[%.4f,%.4f]',s,eval(x0(1)),eval(x0(2))));
end
```

FR 共轭梯度法
```
syms x1 x2;
fx=1.5*x1^2+0.5*x2^2-x1*x2-2*x1;
var=[x1,x2];
x0=[-2,4];
delta=0.1;
df=jacobian(fx,var);
d_x0=Funval(df,var,x0);
f_x0=Funval(fx,var,x0);
tol=norm(d_x0);
s=0;
disp(sprintf('s=%d,x=[%.4f,%.4f],tol=%.4f,fx=%.4f\n',s,eval(vpa(x0(1))),eval(vpa(x0(2))),eval(tol),eval(vpa(f_x0))));

syms k;
v0=-d_x0;
while tol>delta
      s=s+1;
      x1=x0+k*v0;
fk=Funval(fx,var,x1);
[a,b]=model_fb_search(fk,0,0.1);
k_min=model_golden_search(fk,a,b);
      x1=x0+k_min*v0;
d_x0=Funval(df,var,x0);
d_x1=Funval(df,var,x1);
tol=norm(d_x1);
v1=-dx1+v0*norm(d_x1)^2/(norm(d_x0)^2);
      v0=v1;
      x0=x1;
f_x0=Funval(fx,var,x0);
disp(sprintf('s=%d,x=[%.4f,%.4f],tol=%.4f,fx=%.4f',s,eval(vpa(x0(1))),eval(vpa(x0(2))),eval(tol),eval(vpa(f_x0))));
end
```

POWell 算法
```
syms k;
syms x1 x2;
var=[x1;x2];

%fx=(x1+x2)^2+(x1-1)^2;
%x0=[2;1];
%delta=0.1;
%d=[1,0;
0,1];

fx=(x1)^2+2*(x2)^2;
x0=[20;20];
delta=0.1;
d=[1,-1;
      1,1];

[m,n]=size(d);
tol=1;
s=0;
fx_x0=subs(fx,var,x0);
display(sprintf('s=%d,x=[%.4f,%.4f],f(x)=%.4f',s,x0(1),x0(2),eval(fx_x0)));
while tol>delta
      s=s+1;
      y=zeros(m,n);
      y(:,1)=x0;
   for i=1:n
      yp=y(:,i)+k*d(:,i);
      fy=subs(fx,var,yp);
      k_opt=solve(diff(fy),k);
      y(:,i+1)=y(:,i)+k_opt*d(:,i);
   end
      d(:,n+1)=y(:,n+1)-y(:,1);
      xp=y(:,n+1)+k*d(:,n+1);
      fy=subs(fx,var,xp);
      k_opt=solve(diff(fy),k);
      xp=y(:,n+1)+k_opt*d(:,n+1);

      tol=norm(xp-x0);
      x0=xp;
      for i=1:n
          d(:,i)=d(:,i+1);
      end

fx_x0=subs(fx,var,x0);
display(sprintf('s=%d,x=[%.4f,%.4f],f(x)=%.4f',s,eval(x0(1)),eval(x0(2)),eval(fx_x0)));
end
```

步长加速法:
```
syms x1 x2;
syms k;
fx=(1-x1)^2+5*(x2-x1^2)^2;
x0=[2;0];
lambda=[0.5,0.5];
alpha=1;
beta=0.5;
var=[x1;x2];
tol=1;
delta=0.01;
d=eye(2,2);
n=length(d);

s=0;
fx_x0=subs(fx,var,x0);
display(sprintf('s=%d,x=[%.4f,%.4f],f(x)=%.4f',s,x0(1),x0(2),eval(fx_x0)));

while tol>delta
   s=s+1;
   %%%%%探测
   y1=x0;
   fx_y1=subs(fx,var,y1);
   for i=1:n
     %正向探测
     di=d(:,i)                ;
     yp=y1+lambda(i)*di;
     fx_yp=subs(fx,var,yp);
     if fx_yp<fx_y1
         y1=yp;
     else %负向探测
   yp=y1-lambda(i)*di;
   fx_yp=subs(fx,var,yp);
   if fx_yp<fx_y1
       y1=yp;
   else
lambda(i)=lambda(i)/2;
     end
   end
end
      %%%%模式移动
xp=y1;
fx_xp=subs(fx,var,xp);
   if      fx_xp<fx_y1
   x0=y1+alpha*(y1-x0);
   else
lambda=beta*lambda;
      end
tol=norm(lambda);
fx_x0=subs(fx,var,x0);
display(sprintf('s=%d,x=[%.4f,%.4f],f(x)=%.4f',s,x0(1),x0(2),eval(fx_x0)));
end
```

坐标轮换法
```
syms k;
syms x1 x1;
var=[x1;x2];
fx=(x1-2)^4+(x1-2*x2)^2;
x0=[0;3];
delta=0.04;
fx_x0=subs(fx,var,x0);
d=eye(2,2);
n=length(d);
s=0;
display(sprintf('s=%d,x=[%.4f,%.4f],f(x)=%.4f',s,x0(1),x0(2),eval(fx_x0)));

tol=1;
while tol>delta
      s=s+1;
      y=x0;
      for i=1:n
            d0=d(:,i);
            yp=y+k*d0;
            fk=subs(fx,var,yp);
            [a,b]=model_fb_search(fk,0,0.1);
            k_opt=model_golden_search(fk,a,b);
            %k_opt=solve(diff(fk),k);
         yp=y+k_opt*d0;
         y=yp;
      end
      tol=norm(y-x0);
      x0=y;
      fx_x0=subs(fx,var,x0);
display(sprintf('s=%d,x=[%.4f,%.4f],f(x)=%.4f',s,x0(1),x0(2),eval(fx_x0)));
end
```

二分法
```
%   获取左右取点的偏差值  eps
%   精度：   delta
%   最优值时的变量：x0
%   最优值：   minf
syms x;
fx=8*x^3-2*x^2-7*x+3;
a=0;b=1;
eps=0.1;
delta=0.0001;

x0=(a+b)/2;
pre_f_x0=100;
f_x0=subs(fx,findsym(fx),x0);
k=0;
disp(sprintf('k=%d,x0=%.4f,f(x0)=%.4f\n',k,x0,eval(f_x0)));

 while         abs(f_x0-pre_f_x0)>=delta
      k=k+1;
      c=(a+b)/2;
      x1=c-eps/2;
      x2=c+eps/2;
      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;
pre_f_x0=f_x0;
f_x0=subs(fx,findsym(fx),x0);
disp(sprintf('k=%d,x0=%.4f,f(x0)=%.4f\n',k,x0,eval(f_x0)));
end
```

fibonacci 法
```
syms x;
fx=x^2-x+2;
a=-1;b=3;
delta=0.5;

x0=(a+b)/2;
f_x0=subs(fx,findsym(fx),x0);
k=0;
disp(sprintf('k=%d,x=%.4f,fx=%.4f\n',k,x0,eval(f_x0)));
Fn=ceil((b-a)/delta);
%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
```

黄金分割法
```
syms x;
fx=x^2-6*x+2;
a=0;b=10;
delta=0.3;

x0=(a+b)/2;
f_x0=subs(fx,findsym(fx),x0);
k=0;
disp(sprintf('k=%d,[%.4f-%.4f],x=%.4f,fx=%.4f\n',k,a,b,x0,eval(f_x0)));

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
```

牛顿法
```
syms x;
fx=8*x^3-2*x^2-7*x+3;
delta=0.001;
x0=1;
fx0=subs(fx,findsym(fx),x0);
df=diff(fx);
d2f=diff(df);
k=0;
v_df=subs(df,findsym(df),x0);
disp(sprintf('k=%d,x=%.4f,fx=%.4f\n',k,x0,eval(fx0)));

while v_df>delta
       k=k+1;
       v_df=subs(df,findsym(df),x0);
       v_d2f=subs(d2f,findsym(d2f),x0);
       x0=x0-v_df/v_d2f;
       fx0=subs(fx,findsym(fx),x0);
       disp(sprintf('k=%d,x=%.4f,fx=%.4f\n',k,eval(x0),eval(fx0)));
end
```