1. (20%) Consider the problem

$$\min_{x_1,x_2} \quad 0.1*(x_1-3)^2+x_2^2 \qquad \text{s.t.} \quad x_1^2+x_2^2-1\le 0 \tag{1}$$

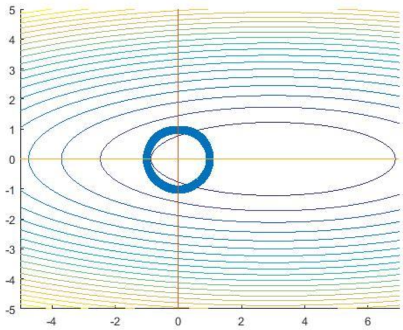
(a) Write down the KKT conditions for (1).

(b) Solve the KKT conditions and find the optimal solutions, including the Lagrangian parameters.

(c) Compute the reduced Hessian and check the second order conditions for the solution.

A1.(a):

- $\nabla L(\vec{x},\lambda)=\nabla[0.1*(x_1-3)^2+x_2^2]+\lambda\nabla[x_1^2+x_2^2-1]=0$
- $x_1^2+x_2^2-1\le 0$
- $\lambda*[x_1^2+x_2^2-1]=0$
- $\lambda\ge 0$



A1.(b) 由圖可知 min at (1,0)

$d_{x_1}L(\vec{x},\lambda)=0.2*(x_1-3)+2*\lambda*x_1=0$ -> $x_1*(0.2+2*\lambda)=0.6$

$d_{x_2}L(\vec{x},\lambda)=2*x_2+2*\lambda*x_2=0$ -> $x_2=0$

$d_{\lambda}L(\vec{x},\lambda)=x_1^2+x_2^2-1=0$ -> $(\frac{0.6}{0.2+2*\lambda})^2=1\to 50\lambda^2+5\lambda-4=0$

$($*無整數解, 以matlab求近似解到小數點第一位*$)\cong(\lambda+0.3)(\lambda-0.2)=0$ 取$\lambda=0.2$

$\mathbf{x_1=\left(\frac{0.6}{0.2+2*0.2}\right)=1\ ,x_2=0\quad \lambda=0.2>=0}$

A1.(c)

$$\nabla L=\begin{bmatrix}0.2*(x_1-3)+2*\lambda*x_1\\ 2*x_2+2*\lambda*x_2\end{bmatrix}=0$$

$$\nabla\nabla L=\begin{bmatrix}0.2+2*\lambda & 0\\ 0 & 2+2*\lambda\end{bmatrix}\to\nabla\nabla L(\lambda=0.2)=\begin{bmatrix}0.6 & 0\\ 0 & 2.4\end{bmatrix}$$

$$\nabla C_1(x^*)=\begin{bmatrix}2*x_1\\ 2*x_2\end{bmatrix}\to\nabla C_1(1,0)=\begin{bmatrix}2\\0\end{bmatrix}\ \because\lambda>0\ \ \therefore\nabla C_1(x,\lambda)w=0$$

$$\therefore W=\left(\begin{matrix}0\\ w_2\end{matrix}\middle|\ w_2\in R\ \ and\ \ w_2\ne 0\right)\ne\vec{0}$$

$$\therefore w*\nabla\nabla L*w=\begin{pmatrix}0\\ w_2\end{pmatrix}^T*\begin{bmatrix}0.6 & 0\\ 0 & 2.4\end{bmatrix}*\begin{pmatrix}0\\ w_2\end{pmatrix}=2.4*(w_2)^2>0$$

$\because$ the sufficient conditions staified $\therefore x^*=\begin{pmatrix}1\\0\end{pmatrix}$ is stric local soultion

2. (20%) The trust region method (for unconstrained optimization problem) needs to solve a local model in each step

$$\min_{\vec{p}} \quad m(\vec{p}) = \frac{1}{2}\vec{p}^T A\vec{p} + \vec{g}^T\vec{p}$$
$$\text{s.t.} \quad \vec{p}^T\vec{p} \le \Delta^2.$$

Prove that the optimal solution $\vec{p}^*$ of the local model satisfies

$$\begin{aligned} (A+\lambda I)\vec{p}^* &= -\vec{g} \\ \lambda(\Delta - \|\vec{p}^*\|) &= 0 \\ (A+\lambda I) \quad &\text{is} \quad \text{positive semidefinite.} \end{aligned}$$

(Hint: to prove the last statement, you only need to consider the directions in the *critical cone*.)

A2:

假設p^**為最佳解*(使 $m(p^*)<=m(p)$)

$$L = \frac{1}{2}p^T Ap + g^T p - u*(\Delta^2 - p^T p)$$

$\because$ KKT $\therefore \nabla_x L = g + A*p^* + 2*u*p^* = 0$

令$\lambda = 2u \to \nabla_x L = g + A*p^* + \lambda * p^* = 0 \to (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I})p^* = -\mathbf{g}$

$\because$complementarity condition $\therefore \lambda * \left(\Delta^2 - {p^*}^T p^*\right) = 0 \to \lambda * (\Delta - ||p^*||) = 0$

if $\lambda = 0 \to$ inactive $\to \Delta^2 - {p^*}^T p^* \ge 0$ *此限制式無存在意義.*

$\to$ 轉為無限制式求極值問題.$\to$ $\mathbf{A}+\boldsymbol{\lambda}\mathbf{I} = \mathbf{A}$ **為半正定矩陣**(否則無解)

If $\lambda > 0 \to$ active $\to \Delta^2 - {p^*}^T p^* = 0 \to \Delta^2 = {p^*}^T p^* \to \Delta = ||p^*||$

由假設 $m(p^*) \le m(p) \to g^T * p^* + \frac{1}{2}{p^*}^T * A * p^* \le g^T * p + \frac{1}{2}p^T * A * p$---1

$\because (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I})p^* = -\mathbf{g} \to \quad -(\mathbf{A}+\boldsymbol{\lambda}\mathbf{I})p^* = \mathbf{g}$

$$\therefore -{p^*}^T * (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I}) * p^* + \frac{1}{2}{p^*}^T * A * p^* \le -{p^*}^T * (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I}) * p + \frac{1}{2}p^T * A * p$$

式子兩邊同加上$\frac{1}{2}\lambda\,\Delta^2 = \frac{1}{2}\lambda({p^*}^T p^*) = \frac{1}{2}\lambda(p^T p)$

$$-{p^*}^T * (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I}) * p^* + \frac{1}{2}{p^*}^T * (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I}) * p^* \le -{p^*}^T * (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I}) * p + \frac{1}{2}p^T * (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I}) * p$$

上式整理後$\to \frac{1}{2}\left({p^*}^T * (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I}) * p\right) - {p^*}^T * (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I}) * p + \frac{1}{2}p^T * (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I}) * p \ge 0$

$\to \left({p^*}^T * (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I}) * p\right) - 2 * {p^*}^T * (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I}) * p + p^T * (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I}) * p \ge 0$

$\to \left((p^* - p)^T * (\mathbf{A}+\boldsymbol{\lambda}\mathbf{I}) * (p^* - p)\right) \ge 0$

$\to$ $\mathbf{A}+\boldsymbol{\lambda}\mathbf{I} = \mathbf{A}$ **為半正定矩陣**

3. (60%) Implement the Interior Point Method (IPM), as shown in Figure 1, to solve linear programming problem.

$$\begin{aligned} \min_{\vec{x}} \quad & \vec{c}^T \vec{x} \\ \text{s.t.} \quad & A\vec{x} - \vec{s} = \vec{b} \\ & \vec{s} \geq 0 \end{aligned}$$

You can assume $\vec{x}_0 = 0$ is a feasible interior point.

Used Example

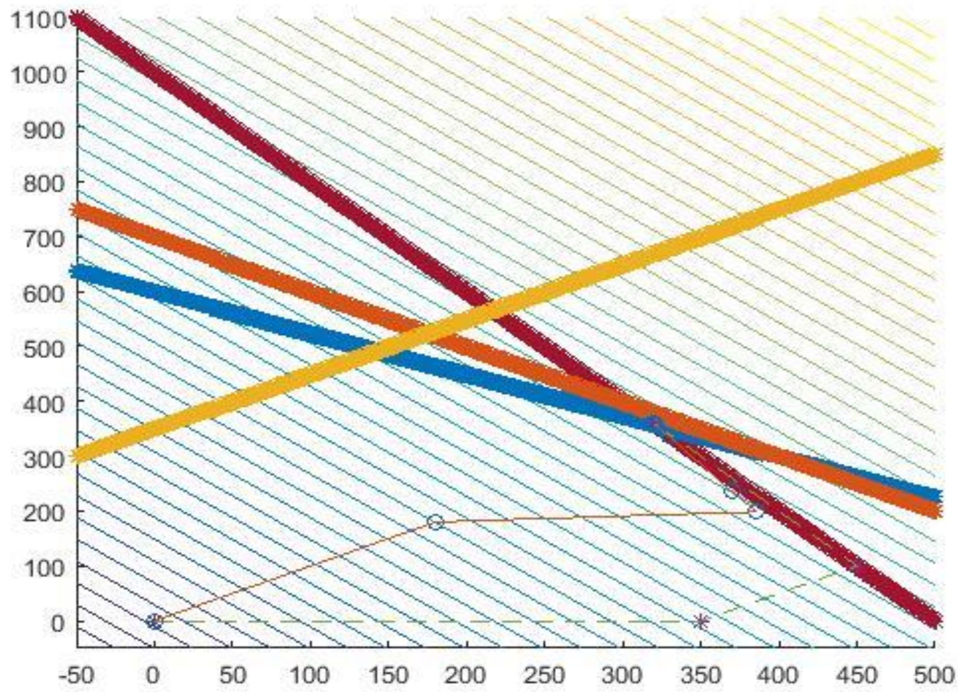
$$\begin{aligned} \max_{x_1,x_2} \quad & z = 8x_1 + 5x_2 \\ \text{s.t.} \quad & 2x_1 + x_2 \leq 1000 \\ & 3x_1 + 4x_2 \leq 2400 \\ & x_1 + x_2 \leq 700 \\ & x_1 - x_2 \leq 350 \\ & x_1, x_2 \geq 0 \end{aligned}$$

轉換問題為$\min\limits_{x_1,x_2} z = -8x_1 - 5x_2, \quad \sigma = 0.4$ (*測試過,步數最少*)

$$A = \begin{pmatrix} -2 & -1 \\ -3 & -4 \\ -1 & -1 \\ -1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad b = \begin{pmatrix} -1000 \\ -2400 \\ -700 \\ -350 \\ 0 \\ 0 \end{pmatrix} \quad lambda = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} s_0 = \begin{pmatrix} 1000 \\ 2400 \\ 700 \\ 350 \\ 0 \\ 0 \end{pmatrix}$$

$c = \begin{pmatrix} -8 \\ -5 \end{pmatrix}, x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ 收斂條件改 (u <=0.001)而非u==0 不然要等好久...

IPM(圈圈)與 simplex method(*)比較



IPM 撞牆壁之後,似乎就很難移動了.每次移動均很小步

```
function [Z X]=interior_point_method(A, b, c, x0, lambda0, s0)
  X=[0;0]; %init returnX record path
  %init 相關變數
  x = x0;
  lambda=lambda0;
  s = s0;
  F = [x;lambda;s];
  [m,n] = size(A);
  sigma = 0.4;
  alpha = 100;
  u = (lambda'*s)/m;
  r=0.001;

  while(u >0.001 )   % 收斂條件 u==0
     alpha = 100;
     H=[zeros(n,n),-A',zeros(n,m);
     A,zeros(m,m),eye(m);zeros(m,n),diag(s),diag(lambda)];
     u = (lambda'*s)/m;
     g = [A'*lambda-c;A*x-s-b;sigma*u*ones(m,1)-
     diag(lambda)*diag(s)*ones(m,1)];
     d=H\g;
     %compute appropriate alpha
     try_point= F+alpha*d; %先走alpah步看看
     while(min(try_point(n+1:n+m).*try_point(n+m+1:n+2*m))<r*u)
     %檢查是否越界
        alpha=alpha/2;  %改走小步一點
        try_point= F+alpha*d; %在走走看
     end
     %update
     F = F+alpha*d;  %正式走 updata
     x = F(1:n);
     lambda = F(n+1:n+m);
     s = F(n+m+1:n+2*m);
     X=[X x];  %record path
  end
   x
   Z=c'*x;
end
```