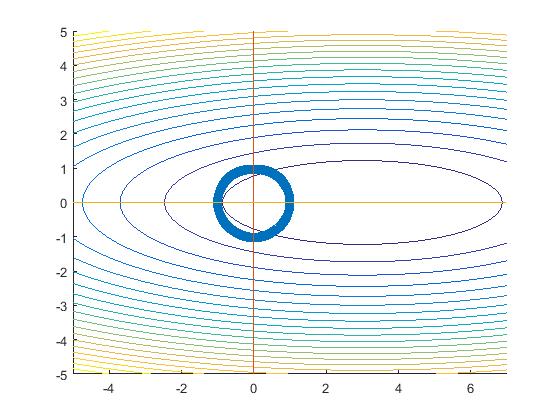


**A1.(a):**

* **=0**



**A1.(b)** **由圖可知min at (1,0)**

)+2=0 ->

->

->

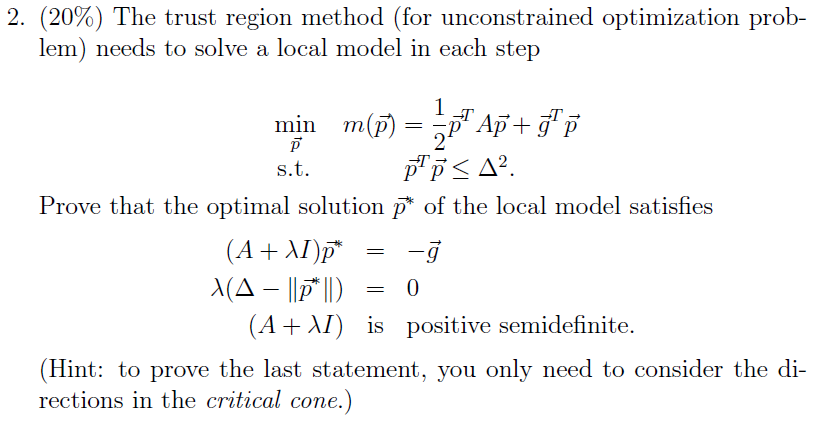
取

**>=0**

**A1.(c)**

=0

W=



**A2:**

**假設**(使m()<=m())

=0

令

complementarity condition

if

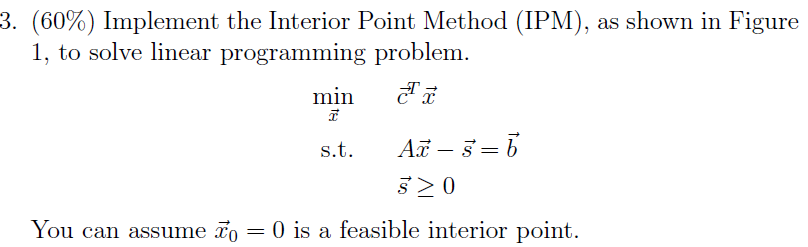
(否則無解)

If active

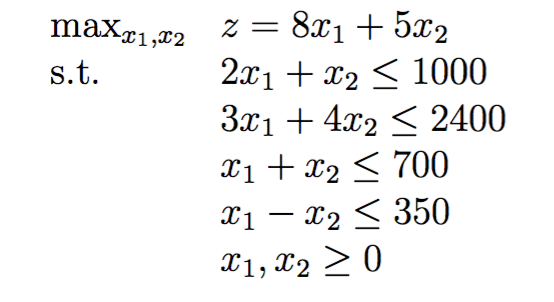
---1

式子兩邊同加上

上式整理後

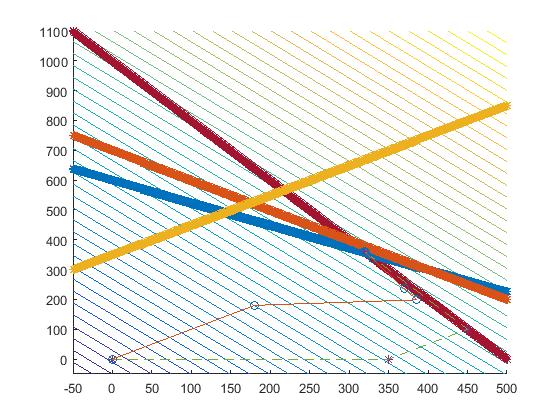


Used Example



轉換問題為,

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