Laboration 3 – Optimeringsproblem

Inledning

Vi skall söka minsta eller största värdet hos en funktion på en mängd, dvs. vi skall lösa s.k. optimeringsproblem

$$\min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}) \quad \text{eller} \quad \max_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x})$$

där $f : \mathbb{R}^n \to \mathbb{R}$ är en funktion och Ω är en mängd som utgörs av hela eller en del av definitionsmängden D_f.

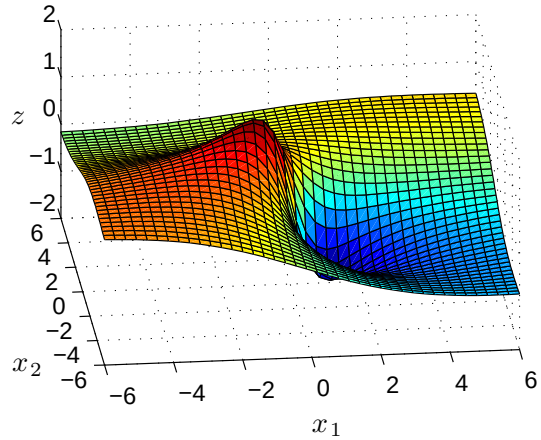
Om Ω är hela definitionsmängden säger vi att vi har ett problem *utan bivillkor* och om Ω är är en del av definitionsmängden har vi ett problem *med bivillkor*.

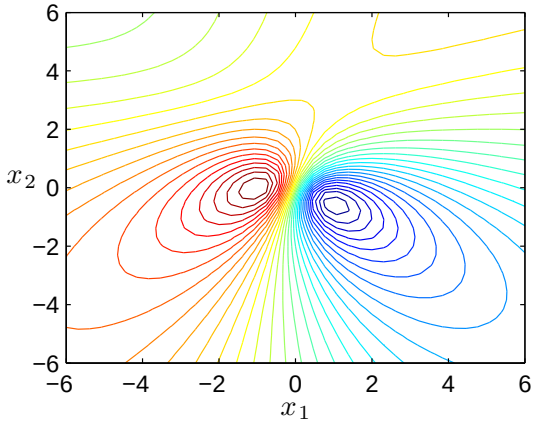
Optimering utan bivillkor

Vi skall bestämma maximi- och minimipunkter samt sadelpunkter till en funktion $f : \mathbb{R}^2 \to \mathbb{R}$. Som exempel tar vi funktionen

$$f(\boldsymbol{x}) = f(x_1, x_2) = \frac{x_2 - 3\,x_1 + x_1\,x_2}{1 + x_1^2 + x_2^2}$$

Vi ritar upp funktionsytan samt nivåkurvor för att få en känsla för var de stationära punkterna finns och av vilken typ de är.





Det ser ut som vi har tre stationära punkter, en lokal minimipunkt, en lokal maximipunkt och en sadelpunkt.

En stationär punkt till $f(\boldsymbol{x})$ är en punkt $\boldsymbol{a}$ för vilken gradientvektorn $\nabla f(\boldsymbol{a}) = \boldsymbol{0}$. Bestämningen av vilken typ av stationär punkt det är kan vi göra med hjälp av egenvärdena till Hessian-matrisen.

I den förra laborationen linjäriserade vi en funktion $f(\boldsymbol{x})$ runt $\boldsymbol{a}$ för att beskriva funktionsytans lutning, dvs. vi gjorde en linjär modell av funktionen runt $\boldsymbol{a}$ med

$$f(\boldsymbol{x}) \approx L(\boldsymbol{x}) = f(\boldsymbol{a}) + \boldsymbol{\nabla} f(\boldsymbol{a})^T(\boldsymbol{x} - \boldsymbol{a})$$

Nu vill vi ha en modell av $f(\boldsymbol{x})$ runt den stationära punkten $\boldsymbol{a}$ som beskriver hur funktionsytan buktar (har vi en kulle, en svacka eller ett pass på ytan), och då ger den linjära modellen inte tillräckligt med information.

En kvadratisk modell av funktionen $f(\boldsymbol{x})$ runt $\boldsymbol{a}$ kommer däremot att beskriva hur funktionsytan buktar och den modellen ges av (Taylorutveckling runt $\boldsymbol{a}$)

$$f(\boldsymbol{x}) \approx f(\boldsymbol{a}) + \boldsymbol{\nabla} f(\boldsymbol{a})^T(\boldsymbol{x} - \boldsymbol{a}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{a})^T \boldsymbol{H}(\boldsymbol{a})(\boldsymbol{x} - \boldsymbol{a})$$

där $\boldsymbol{H}(\boldsymbol{x})$ är Hessianmatrisen av andra derivator

$$\boldsymbol{H}(\boldsymbol{x}) = \begin{bmatrix} f''_{x_1x_1}(\boldsymbol{x}) & f''_{x_1x_2}(\boldsymbol{x}) \\ \\ f''_{x_2x_1}(\boldsymbol{x}) & f''_{x_2x_2}(\boldsymbol{x}) \end{bmatrix}$$

Eftersom $\boldsymbol{a}$ en stationär punkt så är $\boldsymbol{\nabla} f(\boldsymbol{a}) = \boldsymbol{0}$ och vi får att

$$f(\boldsymbol{x}) \approx f(\boldsymbol{a}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{a})^T \boldsymbol{H}(\boldsymbol{a})(\boldsymbol{x} - \boldsymbol{a})$$

Genom att beräkna egenvärdena till $\boldsymbol{H}(\boldsymbol{a})$ kan vi avgöra vilken typ av stationär punkt vi har (se Adams s. 746). Är egenvärdena positiva har vi en minimipunkt, är de negativa har vi en maximipunkt och har de olika tecken har vi en sadelpunkt.

Uppgift 1. Betrakta funktionen $f(x_1, x_2) = 2x_1^3 - 3x_1^2 - 6x_1x_2(x_1 - x_2 - 1)$. Rita upp funktionsytan och nivåkurvor, så att eventuella lokala maximi- och minimipunkter samt sadelpunkter blir synliga.

Newtons metod

I förra laborationen generaliserade vi Newtons metod för att lösa system av icke-linjära ekvationer $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$. Villkoret för en stationär punkt $\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{0}$ är ett sådant ekvationssystem.

Vi kan alltså beräkna stationär punkt till en funktion $f(\boldsymbol{x})$ genom att försöka lösa ekvationen $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}$, där $\boldsymbol{g} : \mathbb{R}^2 \to \mathbb{R}^2$ med

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{\nabla} f(\boldsymbol{x}) = \begin{bmatrix} f'_{x_1}(\boldsymbol{x}) \\ f'_{x_2}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} f'_{x_1}(x_1, x_2) \\ f'_{x_2}(x_1, x_2) \end{bmatrix}$$

med Newtons metod: Antag att vi har en approximation $\boldsymbol{x}_k$ av en stationär punkt, dvs. en approximation av lösningen till $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}$. Vi bildar nästa approximation av lösningen:

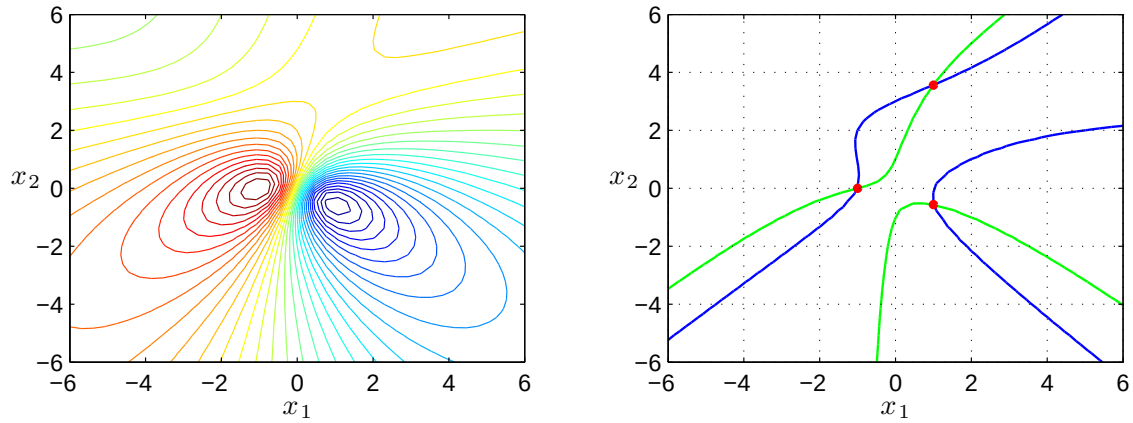
$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{h},$$

där $\boldsymbol{h}$ är lösning till det linjära ekvationssystemet

$$\boldsymbol{D}\boldsymbol{g}(\boldsymbol{x}_k)\boldsymbol{h} = -\boldsymbol{g}(\boldsymbol{x}_k).$$

Här är Jacobimatrisen $\boldsymbol{Dg}(\boldsymbol{x}_k)$ är en $n \times n$-matris. (Jacobimatrisen till $\boldsymbol{g}$ är faktiskt Hessianmatrisen till f.)

Åter till vårt exempel. Nu måste vi finna bra startapproximationer. Vi har redan en graf av nivåkurvorna till funktionen, men för att lite noggrannare se var de stationära punkterna ligger ritar vi också noll-nivåkurvor till de två komponenterna i gradientvektorn. Sedan kan vi läsa av startapproximationer av de stationära punkterna och bestämma dem noggrannt med Newtons metod.

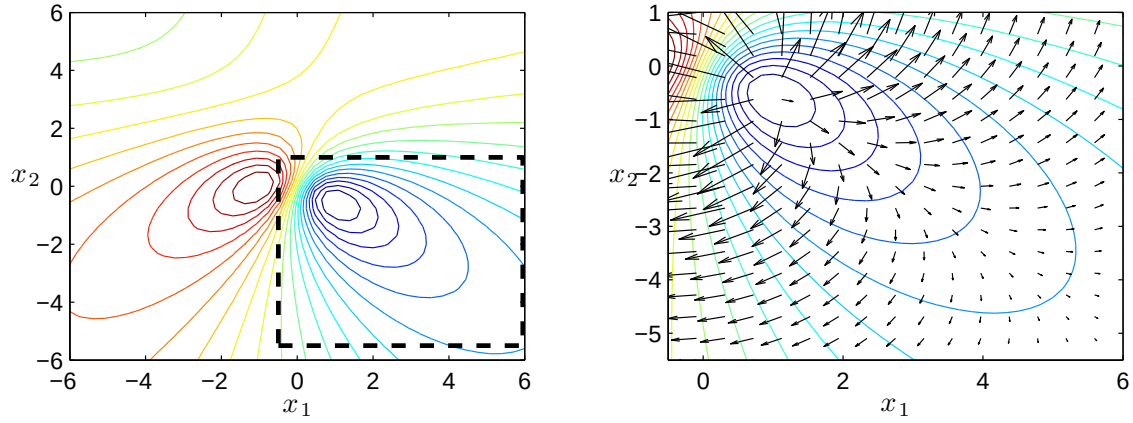


Uppgift 2. Vi ser åter på funktionen $f(x_1, x_2) = 2x_1^3 - 3x_1^2 - 6x_1x_2(x_1 - x_2 - 1)$. Beräkna nu alla stationära punkter noggrannt med Newtons metod, dvs. lös ekvationen $\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{0}$. Därefter bestämmer du vilken typ av stationära punkter vi har genom att studera egenvärdena till Hessianmatrisen $\boldsymbol{H}(\boldsymbol{x})$.

Steepest descentmetoden

En funktion växer som snabbast i gradientens riktning och minskar snabbast i negativa gradientens riktning. Vi skall söka efter minimipunkter genom att utgående från en startapproximation röra oss i negativa gradientens riktning för att komma ned så snabbt som möjligt längs funktionsytan ungefär som vi kan låta en snöboll rulla ut för en sluttning.

Vi ser nedan till vänster nivåkurvorna till vår funktion. Området runt minimipunkten ser vi uppförstorat till höger. Där har vi även ritat ut gradienterna på några ställen.



Vi ser att de pekar mot det håll funktionen växer som snabbast och vi skall alltså gå i motsatta riktningarna.

Vi beskriver nu **Steepest descentmetoden:** Antag att vi har en approximation $\boldsymbol{x}_k$ av en lokal minimipunkt bilda nästa approximation:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{\nabla} f(\boldsymbol{x}_k),$$

där α_k en konstant som avgör hur långt vi går i riktningen $-\boldsymbol{\nabla} \boldsymbol{f}(\boldsymbol{x}_k)$.

Ett bästa val av α_k är sådant att

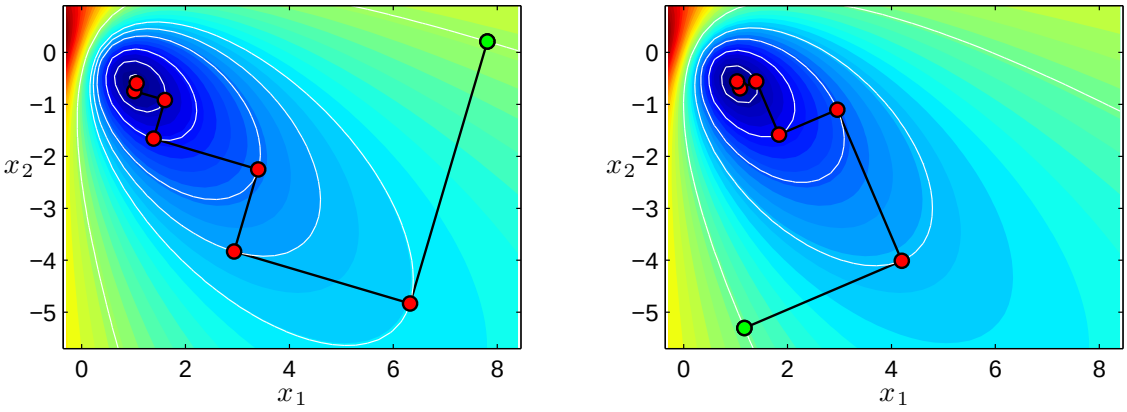
$$g(s) = f(\boldsymbol{x}_k - s\boldsymbol{\nabla} f(\boldsymbol{x}_k))$$

minimeras, vilket leder till ett en-dimensionellt optimeringsproblem. Man kan också välja ett α_k så att

$$f(\boldsymbol{x}_k - \alpha_k \boldsymbol{\nabla} f(\boldsymbol{x}_k)) < f(\boldsymbol{x}_k),$$

vilket garanterar en minskning av funktionsvärdet.

Vi ser hur det går då vi använder Steepest descentmetoden med optimalt α_k-värde för två olika startapproximationer av minimipunkten. Riktigt dåliga approximationer så att vi skall få se hur metoden stegar sig fram.



Startpunkten $\boldsymbol{x}_0$ är markerad med grönt och följande punkter $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots$ med rött. Vi lämnar $\boldsymbol{x}_k$ med rät vinkel mot nivåkurvan genom punkten (varför?) och tar ett steg längs $-\boldsymbol{\nabla} f(\boldsymbol{x}_k)$ tills vi tangerar en nivåkurva (varför?), där har vi nästa approximation $\boldsymbol{x}_{k+1}$. Detta får du reda ut i uppgift 4(e) nedan.

Vill man istället söka en lokal maximipunkt går man antingen i positiv gradientriktning (steepest ascent) eller tillämpar steepest descent på $-f(\boldsymbol{x})$.

Uppgift 3. Betrakta funktionen $f(x_1, x_2) = x_1^2 x_2 e^{-(x_1^2+x_2^2)}$. Rita upp funktionsytan och nivåkurvor, så att eventuella lokala maximi- och minimipunkter samt sadelpunkter blir synliga. Använd sedan Steepest descentmetoden för att beräkna lokala maximi- och minimipunkter.

Färdiga program i MATLAB

I OPTIMIZATION TOOLBOX i MATLAB finns `fminunc` för minimering av funktioner i flera variabler. För funktioner vi vill minimera men som inte är derivarbara finns `fminsearch`. Vill vi minimera en funktion i en enda variabel använder vi `fminbnd`.

Ett anrop av `fminunc` kan se ut så här:

```
>> x=fminunc(@(x)x(1)^2*x(2)*exp(-(x(1)^2+x(2)^2)),[1;-1])
```

vilket minimerar funktionen i uppgift 3 från punkten $(1, -1)$, eller

```
>> x0=ginput(1)
>> x=fminunc(@fun,x0')
```

som minimerar funktionen `fun` från en punkt som ges av ett klick från musen. Här är `fun.m` en funktion i MATLAB, till exempel

```
function f=fun(x)
f=x(1)^2*x(2)*exp(-(x(1)^2+x(2)^2));
```

Uppgift 4. Vi gör här en utförligare undersökning av funktionen

$$f(x, y) = x^3 + 3xy^2 - 15x + y^2 - 15y.$$

(a). Bestäm (för hand om du kan) alla stationära punkter till f och klassificera punkterna som maximi-, minimi- eller sadelpunkt med hjälp av Hessianmatrisen. Bestäm gradientvektorn och Hessianmatrisen för hand. Gör de återstående beräkningarna med MATLAB.

(b). Rita upp funktionsytan och nivåkurvor, så att eventuella lokala maximi- och minimipunkter samt sadelpunkter blir synliga. Jämför med vad du kom fram till i (a).

(c). Funktionen har en minimipunkt. Bestäm denna med Steepest descentmetoden. Klicka in en startpunkt med MATLABs kommando `ginput`.

(d). Bestäm minimipunkten med `fminunc`. Jämför med resultatet i (c).

(e). Beskriv med penna och papper hur Steepest descentmetoden fungerar. Använd dig av gradientvektorer och nivåkurvor.

Optimering med bivillkor

Vi skall se på lösning av optimeringsproblem med bivillkor

$$\min_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x}) \quad \text{eller} \quad \max_{\boldsymbol{x}\in\Omega} f(\boldsymbol{x})$$

där är Ω en mängd som begränsas av bivillkor.

Vi antar att mängden Ω kan beskrivas på följande sätt.

$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0},\ \boldsymbol{h}(\boldsymbol{x}) \leq \boldsymbol{0}\}$$

där $\boldsymbol{g} : \mathbb{R}^n \mapsto \mathbb{R}^m$ och $\boldsymbol{h} : \mathbb{R}^n \mapsto \mathbb{R}^q$.

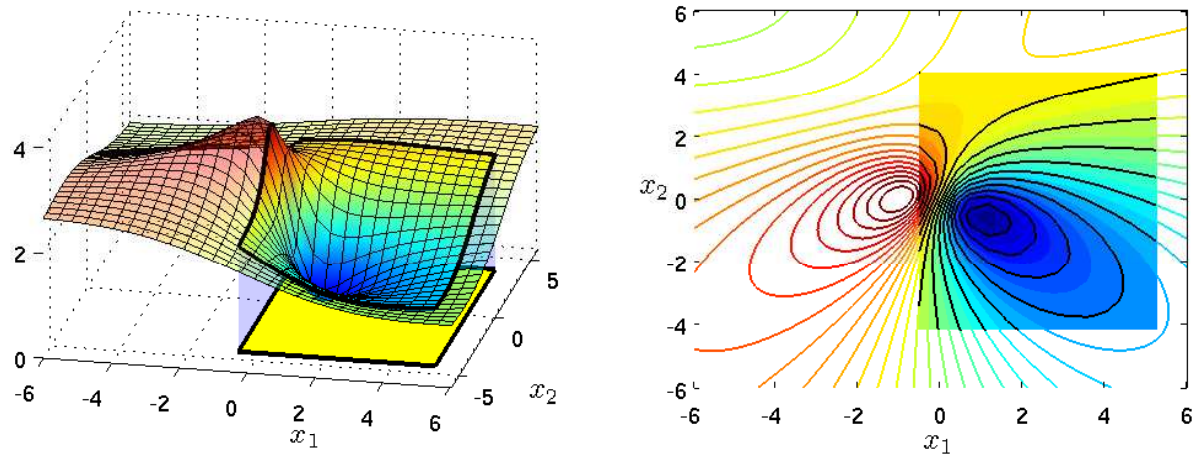
Här kallas f ofta för *objektfunktion* och Ω för *tillåtna mängden*. Bivillkoret och $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}$ kallas *likhetsbivillkor* och $\boldsymbol{h}(\boldsymbol{x}) \leq \boldsymbol{0}$ kallas *olikhetsbivillkor*.

Som exempel ser vi på minimering eller maximering av

$$f(\boldsymbol{x}) = \frac{x_2 - 3\,x_1 + x_1\,x_2}{1 + x_1^2 + x_2^2}$$

då Ω är ett rektangulärt område.

Vi ritar en figur där vi ser funktionsyta och nivåkurvor samt tillåtna mängden.



Vi ser att vi måste undersöka stationära punkter, men också titta på randen av området. Det ser ut som vi har en minimipunkt i det inre av Ω och en maximipunkt på randen av Ω.

Vi kommer inte göra så mycket mer åt denna problemställning, utan nöjer oss med att se vad vi kan göra med färdiga program genom att se på ett exempel.

Färdiga program i MATLAB

I OPTIMIZATION TOOLBOX finns också funktionen `fmincon` som är avsedd för minimering av en funktion under bivillkor.

Objektfunktionen måste nu beskrivas som en funktion, samma gäller de icke-linjära bivillkoren.

Linjära bivillkor $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{c}$ och $\boldsymbol{B}\boldsymbol{x} = \boldsymbol{d}$ samt enkla gränser $\boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}$ på variablerna, ges med hjälp av matriserna $\boldsymbol{A}$ och $\boldsymbol{B}$ samt vektorerna $\boldsymbol{c}$, $\boldsymbol{d}$, $\boldsymbol{l}$ och $\boldsymbol{u}$.

Vi måste också ge en startapproximation av lösningen. Användning enligt

```
>> x=fmincon(@fun,x0,A,c,B,d,l,u,@nonlcon)
```

Bivillkor som inte finns med i vårt aktuella problem ersätts med tomma mängden.

Exempel. Antag att vi vill minimera plåtåtgången vid tillverkningen av en burk med radien x_1 och höjden x_2, givet att den skall rymma volymen $V = 1$.

Arean av burkens yta blir $A = 2\pi x_1 x_2 + 2\pi x_1^2$ och dess volym blir $V = \pi x_1^2 x_2$. Vi kommer alltså ha

$$f(\boldsymbol{x}) = 2\pi x_1 x_2 + 2\pi x_1^2 \quad \text{och} \quad g(\boldsymbol{x}) = \pi x_1^2 x_2 - 1$$

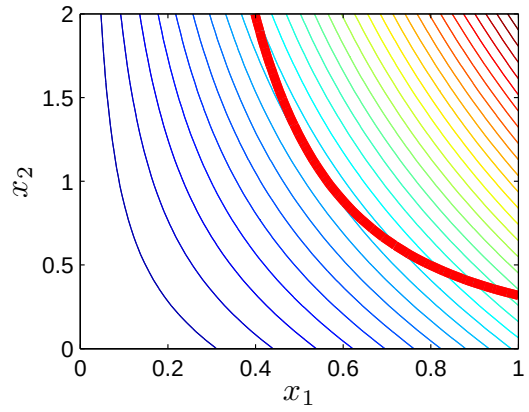
och skall lösa

$$\min_{g(\boldsymbol{x})=0} f(\boldsymbol{x})$$

Vi ritar en bild med nivåkurvor till objektfunktionen samt noll-nivåkurvan till bivillkoret, så att vi ser var bivillkoret är uppfyllt.

```
>> x1=linspace(0,1,40); x2=linspace(0,2,40);
>> [X1,X2]=meshgrid(x1,x2);
>> F=2*pi*X1.*(X1+X2);
```

```
>> H=pi*X1.^2.*X2-1;
>> contour(x1,x2,F,30), hold on
>> contour(x1,x2,H,[0 0],'r','linewidth',4), hold off
>> xlabel('x_1'), ylabel('x_2')
```



Vi ser att $0.5 \leq x_1 \leq 0.6$ och $0.8 \leq x_2 \leq 1.2$, det blir våra enkla gränser.

Nu beskriver vi objektfunktion och bivillkor enligt

```
function f=fun_burk(x)
f=2*pi*x(1)*(x(1)+x(2));

function [g,h]=con_burk(x)
g=[];                   % Vi har inget olikhetsbivillkor
h=pi*x(1)^2*x(2)-1;
```

Läser av en approximation av de optimala värdena på radien x_1 och höjden x_2 och löser sedan problemet noggrannt med `fmincon`

```
>> x0=ginput(1);
>> l=[0.5;0.8]; u=[0.6;1.2];      % Enkla gränser för x
>> x=fmincon(@fun_burk,x0,[],[],[],[],l,u,@con_burk)
x =
    0.5419    1.0839
```