Sök ett X^* sådant att $F(X^*) = 0$ med

$$F = \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix} \text{ och } X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

dvs

$$\begin{cases} f_1(x_1^*, x_2^*) = 0 \\ f_2(x_1^*, x_2^*) = 0 \end{cases}$$

Newtons metod:

$$X^{(k+1)} = X^{(k)} + d^{(k)}$$

där

$$J(X^{(k)})d^{(k)} = -F(X^{(k)})$$

J, Jacobianan eller derivatamatrisen:

$$J(X) = \begin{bmatrix} \frac{\partial f_1(x_1, x_2)}{\partial x_1} & \frac{\partial f_1(x_1, x_2)}{\partial x_2} \\ \frac{\partial f_2(x_1, x_2)}{\partial x_1} & \frac{\partial f_2(x_1, x_2)}{\partial x_2} \end{bmatrix}$$

som kan ersättas med tex. :

$$\begin{bmatrix} \frac{f_1(x_1 + h, x_2) - f_1(x_1 - h, x_2)}{2h} & \frac{f_1(x_1, x_2 + h) - f_1(x_1, x_2 - h)}{2h} \\ \frac{f_2(x_1 + h, x_2) - f_2(x_1 - h, x_2)}{2h} & \frac{f_2(x_1, x_2 + h) - f_2(x_1, x_2 - h)}{2h} \end{bmatrix}$$

där derivatorna ersatts med differenser.

Betrakta det ickelinjära systemet:

$$\begin{cases} f_1(x,y) = (x+3)(y^3-7)+18 \\ f_2(x,y) = \sin(ye^x - 1) \end{cases}$$

```
f1=inline('(x+3).*(y.^3-7)+18','x','y');
f2=inline('sin(y.*exp(x)-1)','x','y');
```

Inline – funktioner.

```
xled=-1:0.1:2; yled=0:0.1:2;
[X,Y]=meshgrid(xled,yled);
```

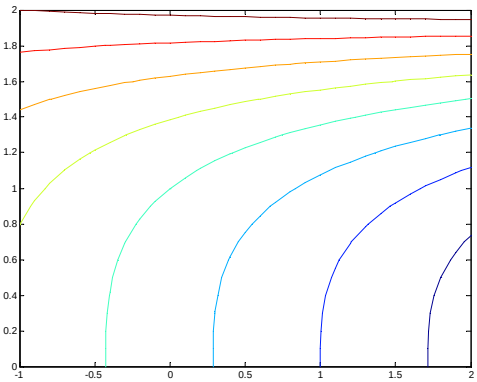
Def. ett rutnät i x- och y-led.

```
F1=f1(X,Y);
```

Beräkna f1 över rutnätet.

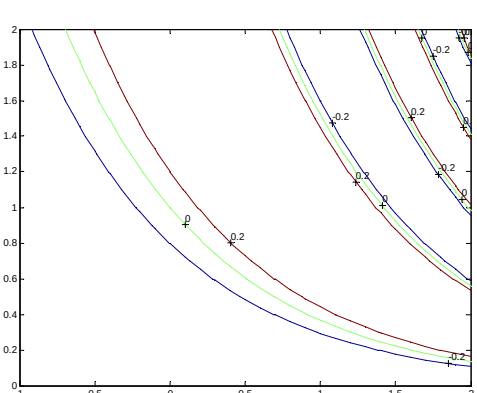
```
figure(1)
contour(X,Y,F1)
figure(2)
```

f1:s nivåkurvor, valda av MATLAB



```
F2=f2(X,Y);
C=contour(X,Y,F2,[-0.2,0,0.2]);
clabel(C)
```

nivåkurvorna –0.2, 0 och 0.2 för f2, med värdena utskrivna.



```
figure(3)
contour(X,Y,F1,[0,0],'r')
hold on
contour(X,Y,F2,[0,0],'b')
grid on
startx=ginput(1)'
plot(startx(1),startx(2),'*k')
xlabel('x');ylabel('y');title('Newtondemo')
```

Rita upp var funktionerna är 0, lägg på ett rutnät.
Klicka in ett startvärden med musen och rita ut punkten.

```
function f=F(x)

  f=[(x(1)+3).*(x(2).^3-7)+18 ; sin(x(2).*exp(x(1))-1)];
```

```
function j=Jdiff(F,x)

  f1=F(x+[1e-8;0]);
  f2=F(x+[0;1e-8]);
  f3=F(x-[1e-8;0]);
  f4=F(x-[0;1e-8]);

  j11=f1(1)-f3(1); j12=f2(1)-f4(1);
  j21=f1(2)-f3(2); j22=f2(2)-f4(2);

  j=[j11 , j12 ; j21 , j22]/2e-8;
```

Anm.:
Om man anropar funktionen med **@** så behövs inte feval.

```
function j=J(x)

  j=[  x(2).^3-7                                  , …
       (x(1)+3)*3.*x(2).^2                        ; ...
       exp(x(1)).*x(2).*cos(exp(x(1)).*x(2)-1)    ,…
       exp(x(1)).*cos(exp(x(1)).*x(2)-1)          ] ;
```

```
function nollst =Newtonsyst( f , x )

while 1

p = menu('Vill Du ta ett Newtonsteg?','Ja','Nej');

if p = = 2
   break
end

Fk=f(x);
Jk=Jdiff(f,x);

% Jk=J(x);

dk=-Jk\Fk;

nyttx=x+dk

plot(nyttx(1),nyttx(2),'*k');plot([x(1);nyttx(1)],[x(2);nyttx(2)],'g')

if norm(x-nyttx)/norm(x)<1e-5
   break
end

x = nyttx;

end

nollst = x;
```

1 = TRUE

Lämna repetitionen om p = 2, om Du klickat på Nej.

Beräkna funktion och derivatamatris i x.

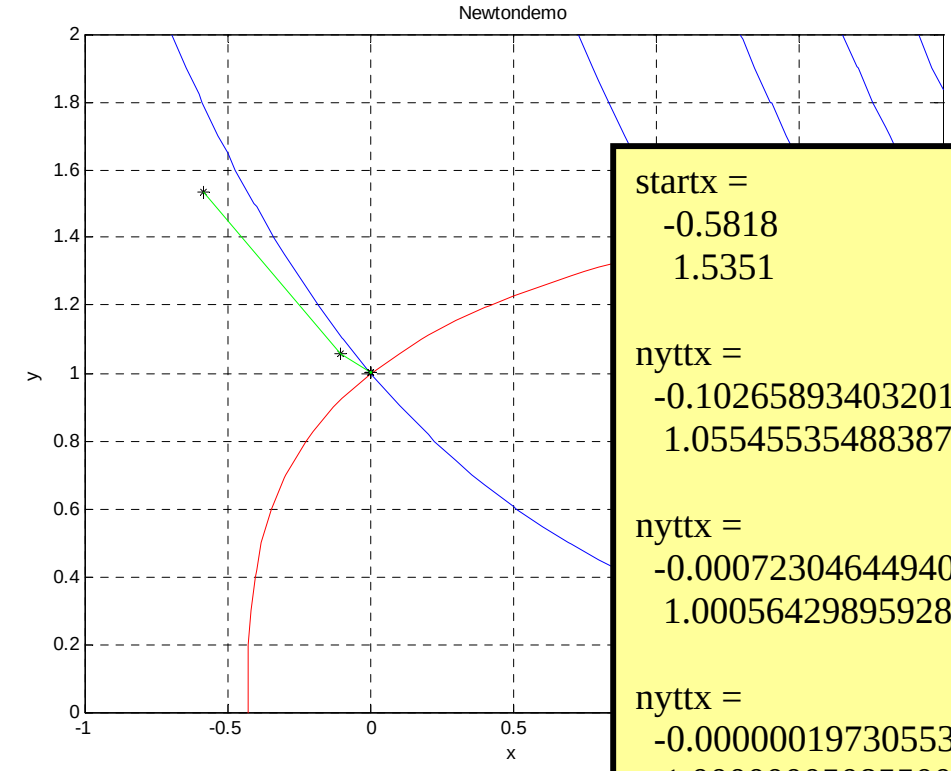
Anm,: feval ej nödv. vid anrop med @.

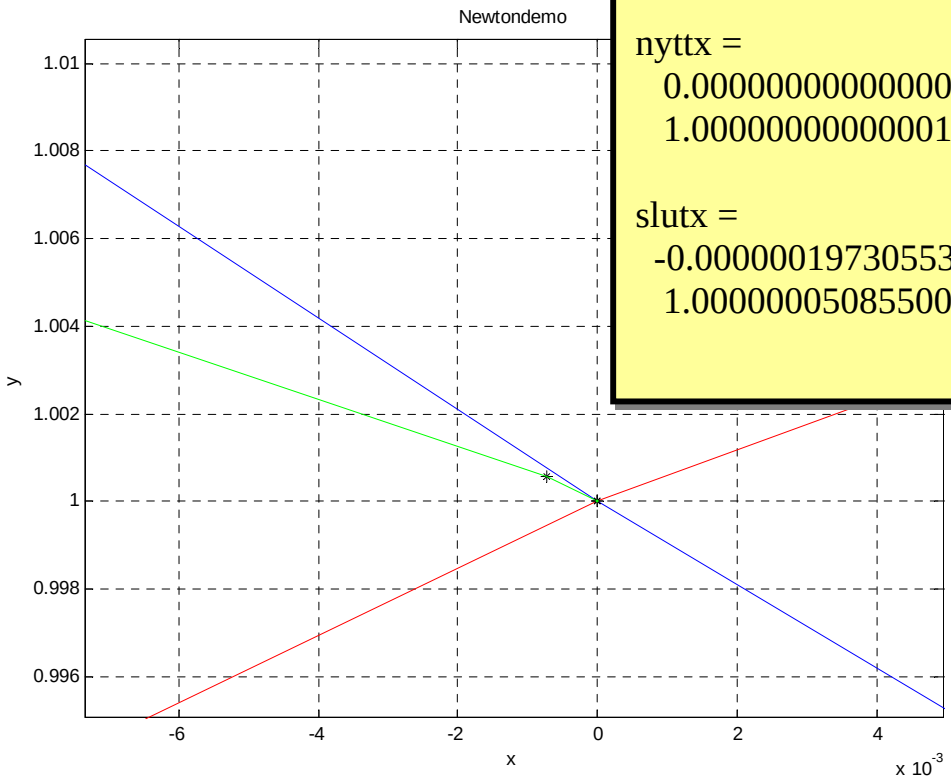
Ett steg Newton.

Avbrott.
" om x nära xnytt "

Uppdatera x inför nytt steg.

Funktionsvärdet.





startx =
-0.5818
1.5351

nyttx =
-0.10265893403201
1.05545535488387

nyttx =
-0.00072304644940
1.00056429895928

nyttx =
-0.00000019730553
1.00000005085500

nyttx =
0.00000000000000
1.00000000000001

slutx =
-0.00000019730553
1.00000005085500