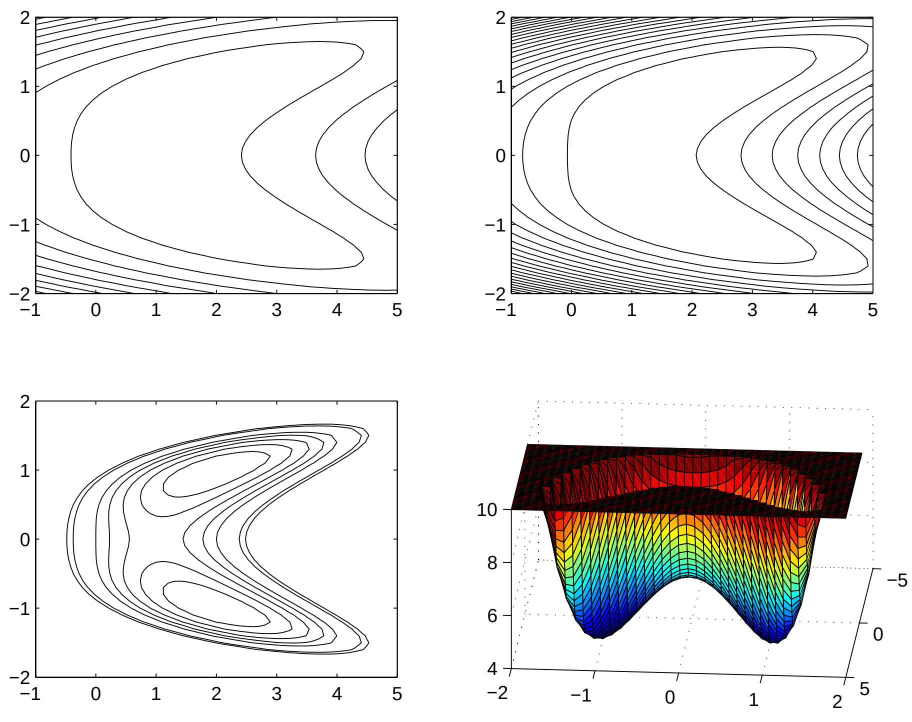
Fö 4, DN1241,1243, Nivåkurvor och annan 3D-grafik

Ref: MATLAB 7 i korthet, sec 14.14.

```
x=-1:0.1:5; y=-2:0.1:2;          % skapa punkternas koordinater
[X,Y]=meshgrid(x,y);             % skapa nätmatriser
Z=2*X.^2+4*Y.^4-4*X.*Y.^2-4*X+8;

subplot(2,2,1)
contour(X,Y,Z)
subplot(2,2,2)
contour(X,Y,Z,20)  %20 st nivåkurvor
subplot(2,2,3)
contour(X,Y,Z,[4.8 5.6 6.4 7.2 8.0 9.8 10.4])
%nivåkurvor enl sista argumentet
subplot(2,2,4)
ind=find(Z>10); Z(ind)=10;
surf(X,Y,Z); rotate3d on;  %bilden roteras
                           %med hjälp av musen
%testa surfc  och surfl  samt shading och colormap
```



Lös systemet

$$10x_1 - x_2 - x_1^3 = 0$$
$$x_1 + 10x_2 - x_3 + x_2^3 = 2$$
$$x_1 + 2x_3 + x_3^3 = 1$$

```
%Newton-Raphssons metod för ekvationssystem
%filnamn: snewton.m  FINNS på kursbiblioteket
format short e, format compact
disp('      x          f(x)              h=J\f(x)');
x=[0.1 0.2 0.5]';    %Startvektor
h=x;
iter=1;
while ( (norm(h,inf) > 1.0e-10*norm(x,inf)) & (iter < 20)),
  f =[10*x(1)-x(2)-x(1)^3
      x(1)+10*x(2)-x(3)+x(2)^3-2
      x(1)+2*x(3)+x(3)^3-1];
  %Jacobian
  J=[10-3*x(1)^2   -1    0
       1   10+3*x(2)^2  -1
       1           0   2+3*x(3)^2];
  h =-J\f;
  disp(iter)
  disp([x f h])
  x=x+h; iter=iter+1;
end
format long
x
------------------------------------------------------------------------

snewton
       x          f(x)              h=J\f(x)
     1
   1.0000e-01   7.9900e-01  -7.6038e-02
   2.0000e-01  -3.9200e-01   4.0896e-02
   5.0000e-01   2.2500e-01  -5.4168e-02
     2
   2.3962e-02  -1.2949e-03   1.0153e-04
   2.4090e-01   1.0719e-03  -2.7978e-04
   4.4583e-01   4.2423e-03  -1.6731e-03
     3
   2.4063e-02  -7.4207e-10  -1.4468e-08
   2.4062e-01   5.6548e-08  -1.4539e-07
   4.4416e-01   3.7392e-06  -1.4371e-06
     4
   2.4063e-02  -4.6563e-17  -1.0440e-14
   2.4062e-01   1.5099e-14  -1.0443e-13
   4.4416e-01   2.7520e-12  -1.0578e-12
x =
   0.02406303171468
   0.24061638394166
   0.44415765605955
```

Följande icke-linjära ekvationssystem skall lösas med Newtons metod:

$$\begin{aligned} x^2 + y^2 &= 2.12 \\ y^2 - x^2y &= 0.04 \end{aligned}$$

För att bestämma startgissningar till flera lösningar till systemet ovan skall vi studera funktionen som i Matlab definieras enligt

```
f=inline('(x.^2 + y.^2-2.12).^2+(y.^2  -x.^2.*y-0.04).^2','x','y');
```

Observera att denna funktion är summan av kvadraterna på skillnaden mellan vänster och höger led i ekvationerna ovan. Då ekvationssystemet har en lösning är $f(x, y) = 0$.

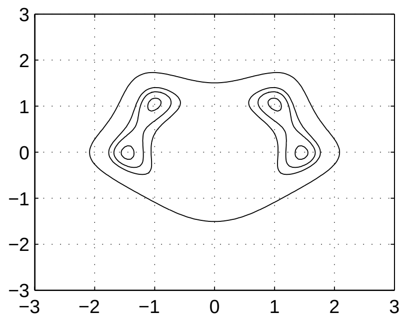
Följande program (`hem34.m` i kursbiblioteket) ritar nivåkurvor för vår funktion.

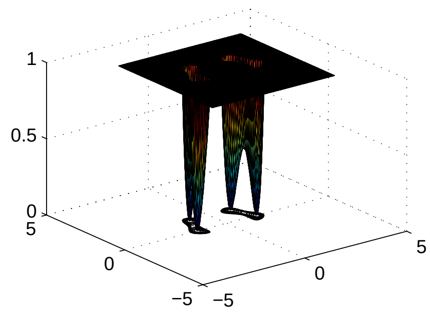
Vi experimenterar med programmet genom att testa olika x- och y-intervall, olika nivåer C i $f(x, y) = C$, testar 3D-plot för att på detta sätt bestämma fyra startgissningar för ekvationssystemet ovan.

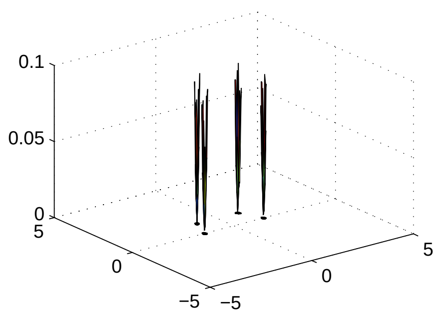
Med koden

```
f=inline('(x.^2 + y.^2-2.12).^2+(y.^2  -x.^2.*y-0.04).^2','x','y');
%x=-1.5:0.02:-0.5; y=-1:0.05:1.2;
x=-3:0.05:3; y=-3:0.05:3;
[X,Y]=meshgrid(x,y);
Z=f(X,Y);
subplot(2,2,1)
contour(X,Y,Z,[5;1;0.5;0.1])
grid
subplot(2,2,2)
ind=find(Z>1); Z(ind)=1;
surfc(X,Y,Z)
subplot(2,2,3)
ind=find(Z>0.1); Z(ind)=inf;
surfc(X,Y,Z)
```

får vi följande tre grafer







De fyra lösningarna synes ligga vid $(1, 1), (1.4, 0), (-1, 1), (-1.4, 0)$. Programmet i 3. ger lösningarna med hög noggrannhet.