$$\begin{cases} x' = -x + x^3, \\ y' = -2y \end{cases}$$

```
%Vektorinis laukas
syms x(t) y(t) t
[x,y]=meshgrid(-5:0.5:5,-5:0.5:5);
dx=-x+x.^3;
dy=-2*y;
dyu = dy./sqrt(dx.^2+dy.^2);
dxu = dx./sqrt(dx.^2+dy.^2);
quiver(x,y,dxu,dyu)
xmin=x(1)-(x(1)-x(2))/2;
xmax=x(end)+(x(1)-x(2))/2;
ymin=y(1)-(y(1)-y(2))/2;
ymax=y(end)+(y(1)-y(2))/2;
axis([xmin xmax ymin ymax]);
hold on;
axis square; xlabel('x'), ylabel('y')
title('Vektorinis laukas ir nulinės izoklinės')
clear all;
%Randame sistemos ramybės taškus
syms x y
f1=-x+x.^3;
f2=-2*y;
spr=vpasolve([f1==0,f2==0],[x,y])
```

```
spr = struct with fields:
    x: [3×1 sym]
    y: [3×1 sym]
```

```
spr.x
```

ans =

$$\begin{pmatrix} 0 \\ -1.0 \\ 1.0 \end{pmatrix}$$

```
spr.y
```

ans =

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

```
%Nulinės izoklinės
fimplicit(f1==0,'-r','LineWidth',2)
hold on;
fimplicit(f2==0,'-r','LineWidth',2)
hold on;
```

```
linestyle = '-';
line([-5 5], [0 0], 'Color', 'black', 'LineStyle', linestyle);
line([0 0],[-5 5], 'Color', 'black', 'LineStyle', linestyle);
hold on;
%Greičių analizė izoklinėse ir trajektorijų elgsena nulinių
% izoklinių suformuotose srityse
%x'=0
sprx=solve(f1==0,x)
```

sprx =

$$\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

```
greitisy=subs(f2,x,sprx)
```

greitisy =

$$\begin{pmatrix} -2\,y \\ -2\,y \\ -2\,y \end{pmatrix}$$

```
spr1x=solve(greitisy>0,y,'IgnoreAnalyticConstraints', true,'Real', true, 'ReturnConditions', tr
```

spr1x = *struct with fields:*

```
             y: [1×1 sym]
    parameters: [1×1 sym]
    conditions: [1×1 sym]
```

```
spr1x.conditions
```

ans = $x < 0$

```
spr2x=solve(greitisy<0,y,'Real', true, 'ReturnConditions', true)
```

spr2x = *struct with fields:*

```
             y: [1×1 sym]
    parameters: [1×1 sym]
    conditions: [1×1 sym]
```

```
spr2x.conditions
```

ans = $0 < x$

```
%y'=0
spry=vpasolve(f2==0,y)
```

spry = 0

```
greitisx=subs(f1,y,spry)
```

greitisx = $x^3 - x$

```
spr1y=solve(greitisx>0,x,'IgnoreAnalyticConstraints', true,'Real', true, 'ReturnConditions', tr
```

```
spr1y = struct with fields:
             x: [2×1 sym]
    parameters: [1×1 sym]
    conditions: [2×1 sym]
```

```
spr1y.conditions
```

ans =

$$\begin{pmatrix} 1 < x \\ x < 0 \land -1 < x \end{pmatrix}$$

```
spr2y=solve(greitisx<0,x,'IgnoreAnalyticConstraints', true,'Real', true, 'ReturnConditions', tr
```

```
spr2y = struct with fields:
             x: [2×1 sym]
    parameters: [1×1 sym]
    conditions: [2×1 sym]
```

```
spr2y.conditions
```

ans =

$$\begin{pmatrix} x < -1 \\ 0 < x \land x < 1 \end{pmatrix}$$

```
%Atliekame dinaminės sistemos lienarizaciją
%ir sudarome Žordano matricą
A=[diff(f1,x) diff(f1,y);diff(f2,x) diff(f2,y)]
```

A =

$$\begin{pmatrix} 3\,x^2 - 1 & 0 \\ 0 & -2 \end{pmatrix}$$

```
%Apskaičiuojame Žordano matricą ramybės taške ir nustatome
% kiekvieno ramybės taško tipą
A1=subs(A,[x,y],[spr.x(1),spr.y(1)])
```

A1 =

$$\begin{pmatrix} -1 & 0 \\ 0 & -2 \end{pmatrix}$$

```
eig(A1)
```

ans =

$$\begin{pmatrix} -2 \\ -1 \end{pmatrix}$$

```
[V,D]=eig(A1)
```

V =

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

D =

$$\begin{pmatrix} -2 & 0 \\ 0 & -1 \end{pmatrix}$$

```
A2=subs(A,[x,y],[spr.x(2),spr.y(2)])
```

A2 =

$$\begin{pmatrix} 2.0 & 0 \\ 0 & -2 \end{pmatrix}$$

```
eig(A2)
```

ans =

$$\begin{pmatrix} -2 \\ 2.0 \end{pmatrix}$$

```
[V,D]=eig(A2)
```

V =

$$\begin{pmatrix} 1.0 & 0 \\ 0 & 1.0 \end{pmatrix}$$

D =

$$\begin{pmatrix} 2.0 & 0 \\ 0 & -2.0 \end{pmatrix}$$

```
A3=subs(A,[x,y],[spr.x(3),spr.y(3)])
```

A3 =

$$\begin{pmatrix} 2.0 & 0 \\ 0 & -2 \end{pmatrix}$$

```
eig(A3)
```

ans =

$$\begin{pmatrix} -2 \\ 2.0 \end{pmatrix}$$

```
[V,D]=eig(A3)
```

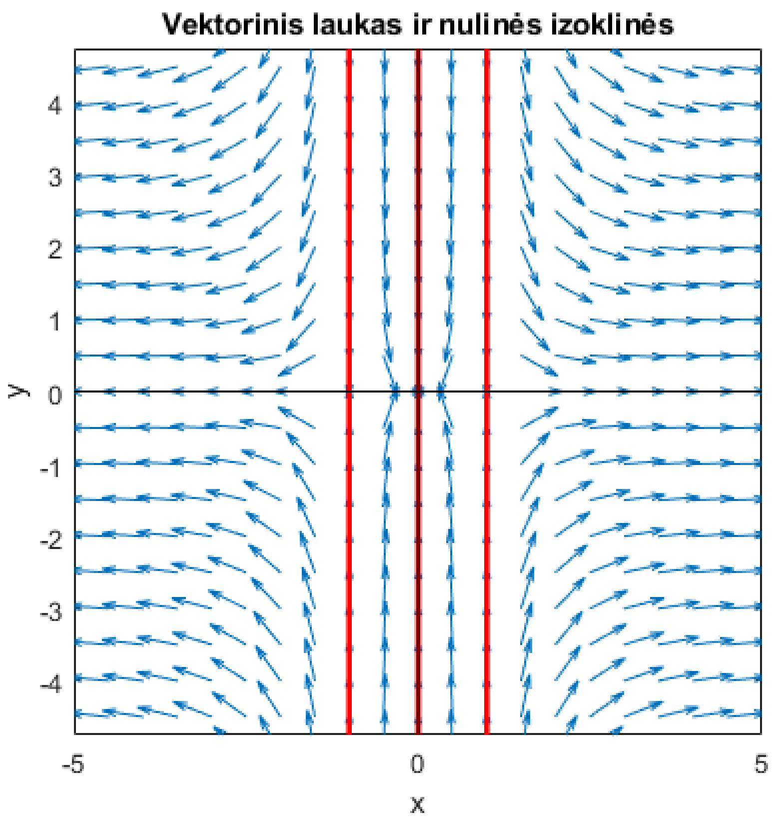
V =

$$\begin{pmatrix} 1.0 & 0 \\ 0 & 1.0 \end{pmatrix}$$

D =

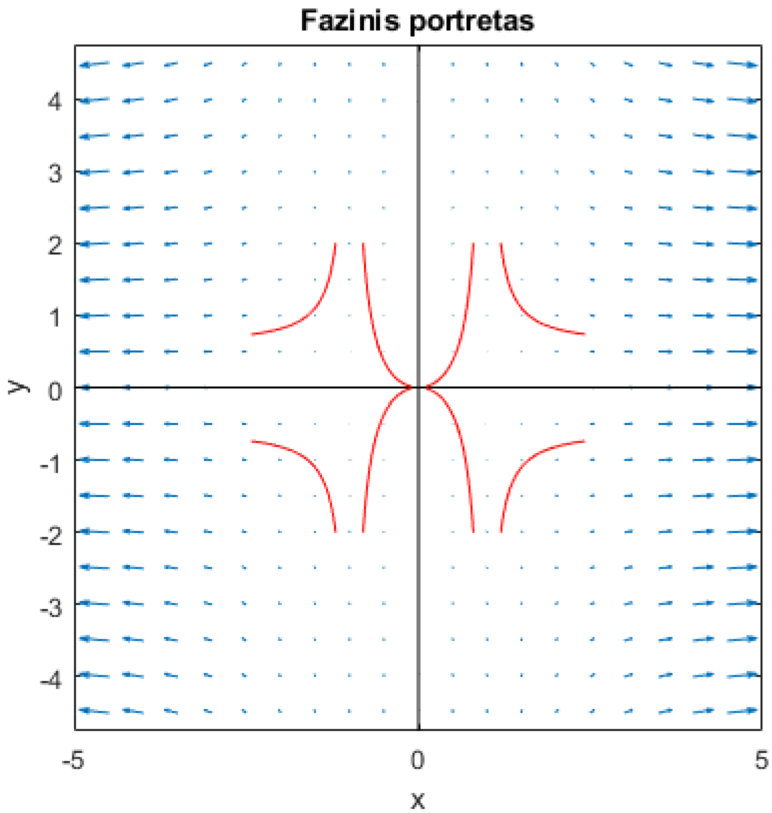
$$\begin{pmatrix} 2.0 & 0 \\ 0 & -2.0 \end{pmatrix}$$

```
hold off;
```




```
syms x(t) y(t) t
[x,y]=meshgrid(-5:0.5:5,-5:0.5:5);
dx=-x+x.^3;
dy=-2*y;
quiver(x,y,dx,dy)
xmin=x(1)-(x(1)-x(2))/2;
xmax=x(end)+(x(1)-x(2))/2;
ymin=y(1)-(y(1)-y(2))/2;
ymax=y(end)+(y(1)-y(2))/2;
axis([xmin xmax ymin ymax]);
hold on;
axis square; xlabel('x'), ylabel('y')
title('Fazinis portretas')
```

```
hold on;
linestyle = '-';
line([-5 5], [0 0], 'Color', 'black', 'LineStyle', linestyle);
line([0 0],[-5 5], 'Color', 'black', 'LineStyle', linestyle);
hold on;
clear all;
tiks(1:2)=1e-3;
tiksl=tiks.';
options=odeset('AbsTol',tiksl);
[t,Y]=ode45(@sist,[0 2.5],[-0.8;2],options);
plot(Y(:,1),Y(:,2),'r')
hold on;
[t,Y]=ode45(@sist,[0 2.5],[0.8;-2],options);
plot(Y(:,1),Y(:,2),'r')
hold on;
[t,Y]=ode45(@sist,[0 2.5],[0.8;2],options);
plot(Y(:,1),Y(:,2),'r')
hold on;
[t,Y]=ode45(@sist,[0 0.5],[-1.2;2],options);
plot(Y(:,1),Y(:,2),'r')
hold on;
[t,Y]=ode45(@sist,[0 0.5],[1.2;2],options);
plot(Y(:,1),Y(:,2),'r')
hold on;
[t,Y]=ode45(@sist,[0 0.5],[1.2;-2],options);
plot(Y(:,1),Y(:,2),'r')
hold on;
[t,Y]=ode45(@sist,[0 0.5],[-1.2;-2],options);
plot(Y(:,1),Y(:,2),'r')
hold on;
[t,Y]=ode45(@sist,[0 2.5],[-0.8;-2],options);
plot(Y(:,1),Y(:,2),'r')
hold off;
```




```
function sistema=sist(t,Y)
s(1)=-Y(1)+(Y(1)).^3;
s(2)=-2*Y(2);
sistema1=[s(1:2)];
sistema=sistema1.';
end
```