Проведем анализ системы (1.1):



Анализ будем проводить с помощью Matlab. Код программы и код функции, реализующие анализ приведены в приложении 1 и 2 соответственно.

В результате исполнения программы получен следующий результат (рис. 1):

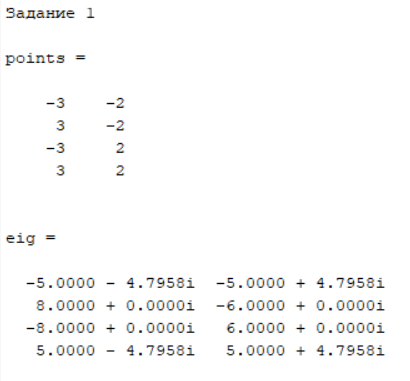


Рисунок - Результат работы программы

И следующий эскиз фазового портрета (рис. 2)

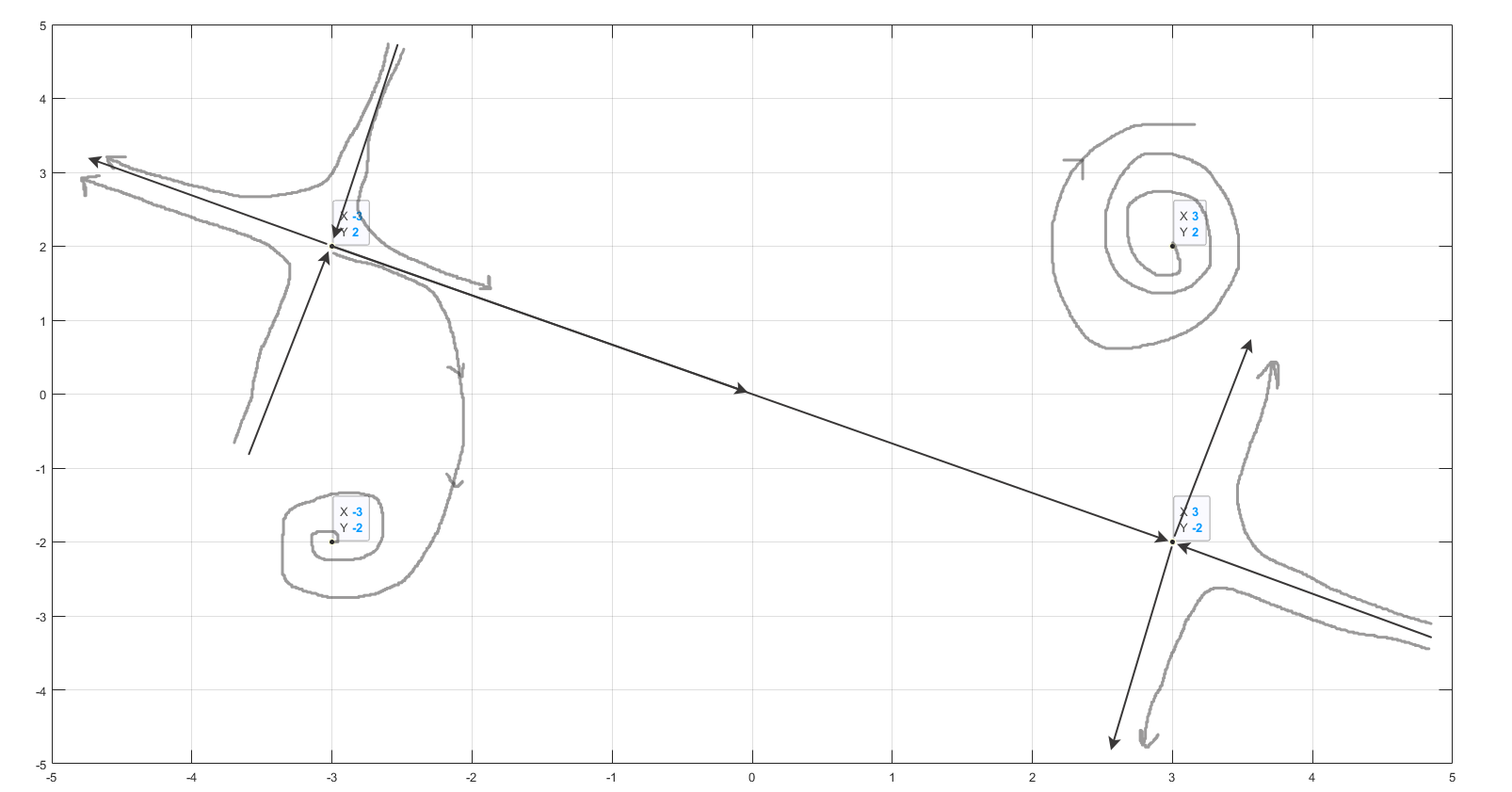


Рисунок - эскиз фазового портрета

Таким образом имеем четыре состояния равновесия с координатами:

[-3 -2], [3 -2], [-3 2], [3 2], первое из которых является устойчивым фокусом, второе и третье седлом, а четвертое – неустойчивым фокусом.

**Приложение 1**

clc, clear, close all

disp('Задание 1')

syms x y

args = [x y];

f = [x^2 - y^2 - 5, x^2 + y^2 - 13];

[points, eig] = special\_points(f, args)

figure(1)

plot(points(:,1),points(:,2),'.')

ylim([-5 5])

xlim([-5 5])

grid on

**Приложение 2**

function [points, stable] = special\_points(func, arg)

switch size(arg,2)

case 1

point\_struct = solve(func, arg, 'Real',true, 'ReturnConditions',true); % roots structure

fields = fieldnames(point\_struct);

x = char(fields(1));

points = point\_struct.x;

if size(point\_struct.parameters, 2) > 0 % check if func is periodic

period\_counter = 0; % periodic roots counter

j = 0; % integer iterator for periodic roots

points = [];

while period\_counter < size(point\_struct.x,1)

for i = 1:size(point\_struct.x,1)

if subs(point\_struct.x(i),point\_struct.parameters,j) < 2\*pi && subs(point\_struct.x(i),point\_struct.parameters,j) >= 0 % check if the root within the peroid

points = [points; subs(point\_struct.x(i),point\_struct.parameters,j)]; % append new roots

else

if subs(point\_struct.x(i),point\_struct.parameters,j) >= 0

period\_counter = period\_counter + 1;

end

end

end

j = j + 1;

end

end

points = eval(points);

df = diff(func);

stable = subs(df, arg, points);

for i = 1:size(stable,1)

if eval(stable(i)) < 0

stable(i) = 'stable';

else

stable(i) = 'unstable';

end

end

case 2

point\_struct = solve(func, arg, 'Real',true, 'ReturnConditions',true); %

points = [point\_struct.x point\_struct.y];

if size(point\_struct.parameters, 2) > 0 % check if func is periodic

period\_counter = 0; % periodic roots counter

j = 0; % integer iterator for periodic roots

points = [];

while period\_counter < size(point\_struct.x,1)

for i = 1:size(point\_struct.x,1)

if subs(point\_struct.x(i),point\_struct.parameters,j) < 2\*pi && subs(point\_struct.x(i),point\_struct.parameters,j) >= 0 % check if the root within the peroid

points = [points; subs(point\_struct.x(i),point\_struct.parameters,j)]; % append new roots

else

if subs(point\_struct.x(i),point\_struct.parameters,j) >= 0

period\_counter = period\_counter + 1;

end

end

end

j = j + 1;

end

end

points = eval(points);

A(1,1) = diff(func(1),arg(1));

A(1,2) = diff(func(1),arg(2));

A(2,1) = diff(func(2),arg(1));

A(2,2) = diff(func(2),arg(2));

a = [0; 0];

for j = 1:size(points,1)

A1 = subs(A,arg(1),points(j,1));

A1 = eval(subs(A1,arg(2),points(j,2)));

a(:,j) = eig(A1);

end

stable = a';

end

end