**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ПЕТРА ВЕЛИКОГО**

**Институт компьютерных наук и технологий**

**Кафедра «Распределенные вычисления и компьютерные сети»**

Лабораторная работа №2

по дисциплине: Моделирование динамических систем

Вариант 2

Выполнил

студент гр. <*подпись*> В.Б.Борисов

Руководитель

от кафедры <*подпись*> Ю.Б.Сениченков

Санкт-Петербург

2016

**Постановка задачи**

Дано уравнение

Построить график правой части дифференциального уравнения как функции от x и пометить на нем особые точки на промежутке [ − 2 π ; 2 π ] . Определить, какие из них устойчивы, а какие — нет (построить фазовый портрет).

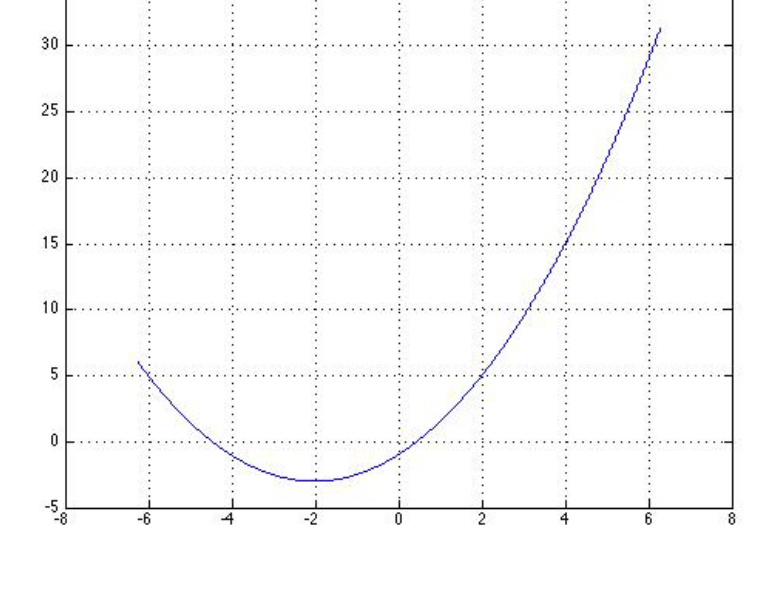
Написать программу поиска корней функции одной переменной на языке MvStudium и MVL. Использовать локальные и глобальные методы поиска: на первом этапе метод золотого сечения (MvStudium), на втором этапе— метод секущих (MvStudium).

Сравнить вычислительные затраты методов. Процедуры должны иметь те же параметры, что и процедура ZEROIN.

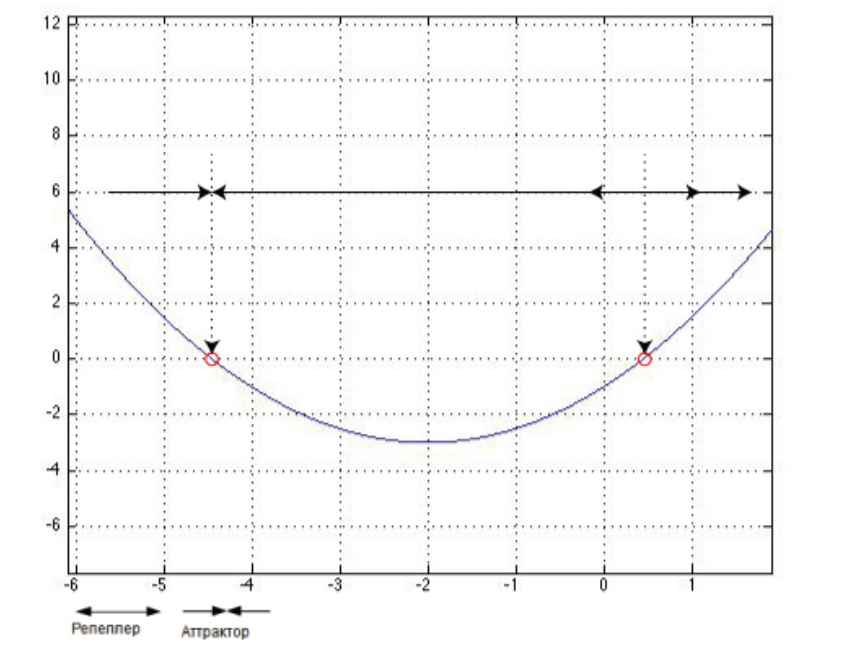
**Ход Решения**

График правой части дифференциального уравнения на промежутке

[−2 π;2 π ] :



Фазовый портрет:

Первая особая точка на промежутке [ − 2 π ; 2 π ] устойчива, т.к. кривая имеет отрицательный наклон; вторая – неустойчива, кривая положительно наклонена.

Код Matlab:

>> x = [-2\*pi:0.0001:2\*pi];

>> y = fun(x);

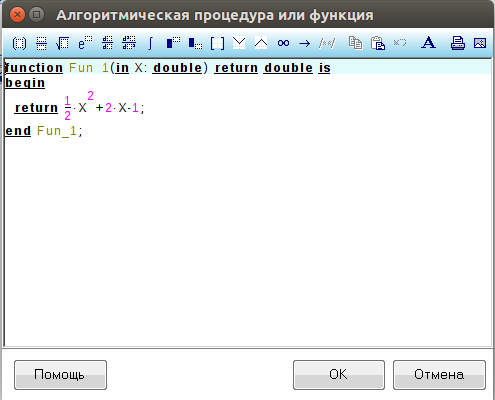
>> plot(x,y), grid

function y = fun( x )

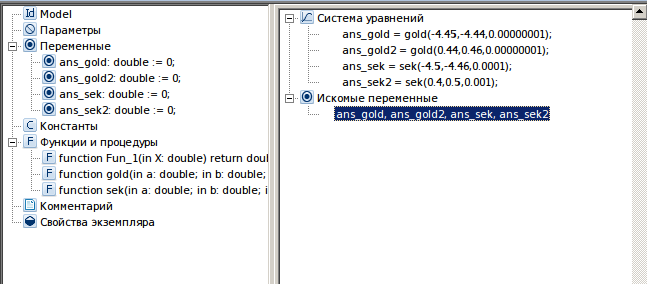
y = (x.^2)/2+2\*x-1;

end

Задаем функцию из условия:



Общее окно программы:



**Реализация метода золотого сечения в MvStudium**

function gold(in a: double; in b: double; in err: double) return double is

c: double := 0;

x1: double := a;

x2: double := b;

t: double := a;

k: double := b;

xf1: double := 0;

xf2: double := 0;

k1: double := 0;

k2: double := 0;

A: double := 0;

B: double := 0;

begin

k1:=(sqrt(5)-1)/2;

k2:=1-k1;

x1:=t+(k-t)\*k1;

x2:=k-(k-t)\*k1;

xf1:=Fun\_1(x1);

xf2:=Fun\_1(x2);

A:=xf1;

B:=xf2;

while abs(k-t)>err loop

if A>=B then

k:=x2;

x2:=x1;

B:=A;

x1:=t+(k-t)\*k1;

A:=Fun\_1(x1);

else

t:=x1;

x1:=x2;

A:=B;

x2:=k-(k-t)\*k1;

B:=Fun\_1(x2);

end if;

end loop;

c:=(k+t)/2;

return c;

end gold;

**Реализация метода секущих в MvStudium**

function sek(in a: double; in b: double; in err: double) return double is

x0: double := a;

x: double := b;

r: double := 0;

d\_: double := 0;

y: double := 0; -- c: double := 0;

begin

r:=x-x0;

d\_:=Fun\_1(x0);

y:=Fun\_1(x);

r:=r/(d\_-y)\*y;

d\_:=y;

x:=x+r;

while abs(r)<err loop

y:=Fun\_1(x);r:=r/(d\_-y)\*y;

d\_:=y;

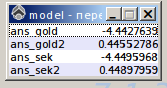
x:=x+r;

end loop;

return x;

end sek;

Полученные решения:



**Проверка решений через встроенную функцию Matlab fzero**

>> x = [-2\*pi:0.000001:2\*pi];

>> y = fun(x);

>> first=fzero(@fun,[-4.9, -4.2],001);

>> second=fzero(@fun,[0.2, 0.7],001);

>> first, second

first = -4.4495

second = 0.4495

**Вывод**

Корни, найденные с помощью методов золотого сечения и метода секущих,примерно совпали по значению с корнями, полученными через встроенную функцию Matlab. Чем точнее выбран промежуток тем точнее будут корни.