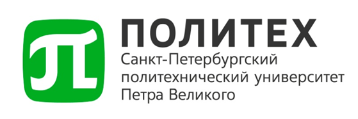
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

***«*САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»**

Институт компьютерных наук и технологий

**Высшая школа программной инженерии**



**ЛАБОРАТОРНАЯ РАБОТА №2**

**Определение особых точек для функции, построение методов поиска**

**корней уравнения в Matlab и MvStudium**

по дисциплине «Математическое моделирование»

Студент А. М. Потапова

гр. 3530202/90202

Руководитель Ю.Б. Сениченков

Ст. преподаватель

Санкт-Петербург

2022 г

**Задание 2\_2**

**Постановка задачи**

1. Построить графики правых частей дифференциальных уравнений из табл. 1.2 как функций от *x* и пометить на графиках особые точки на промежутке [−2π; 2π] . Определить, какие из них устойчивы, а какие — нет (построить фазовый портрет).

Изображение выглядит как текст

Автоматически созданное описание

1. Написать программу поиска корней функции одной переменной на языках Matlab и MVL. Использовать глобальные и локальные методы поиска: на первом этапе методы деления отрезка пополам, метод золотого сечения, случайный поиск, на втором этапе— локальные методы:

* метод ложного положения
* метод секущих
* метод Ньютона

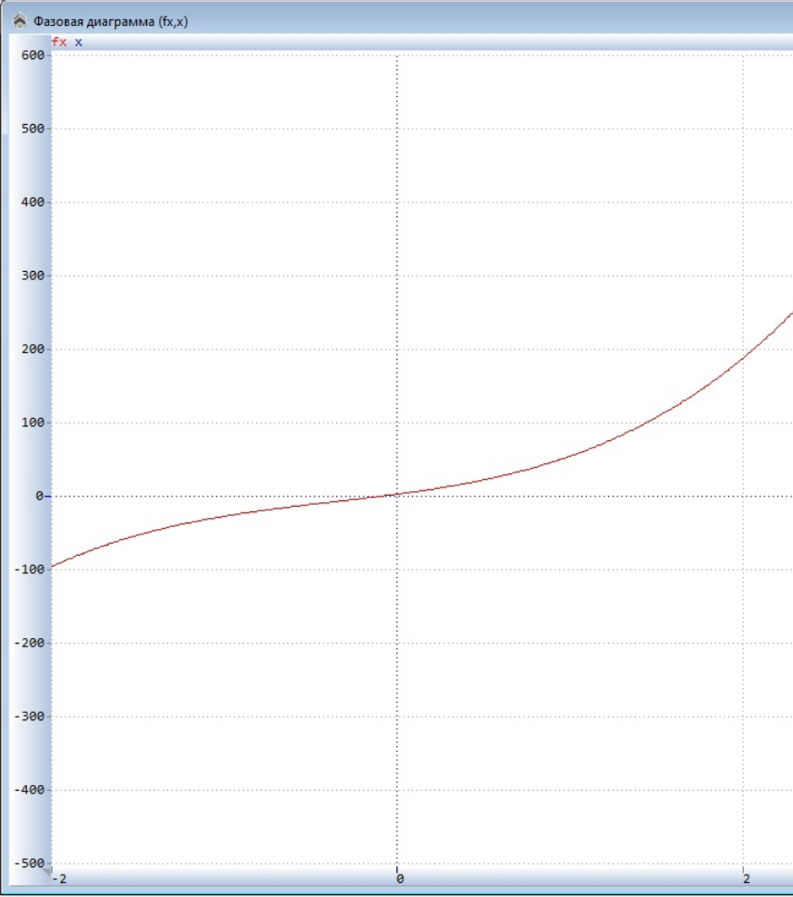
Сравнить вычислительные затраты методов. Процедуры должны иметь те же параметры, что и процедура ZEROIN.

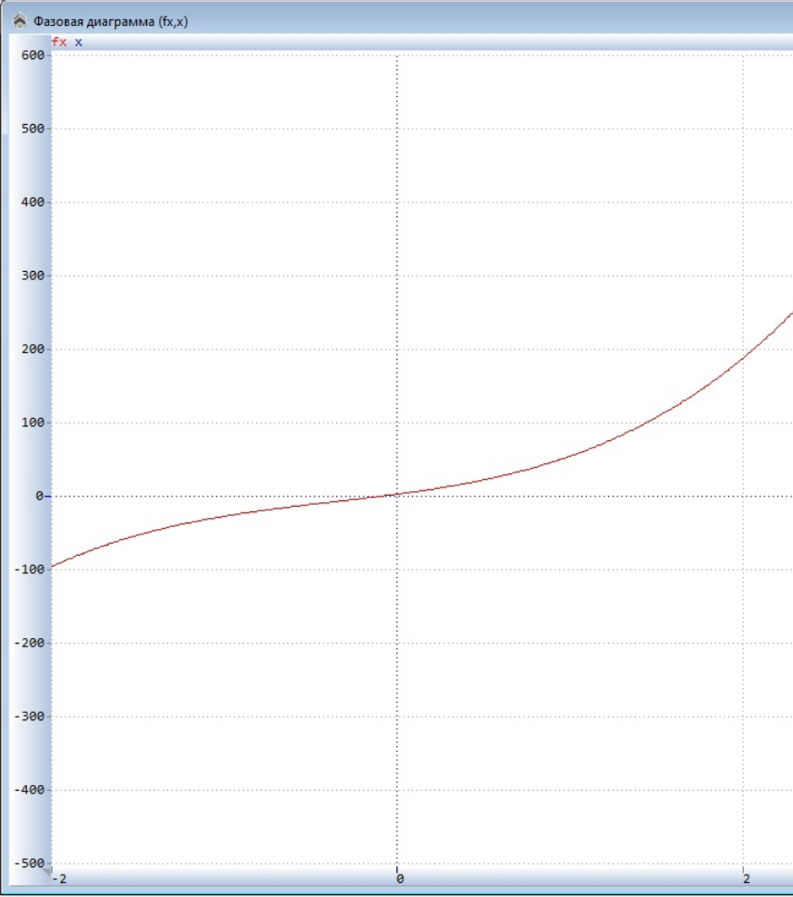
Учебные программы поиска корней функции одной переменной.



**Решение**

*График правой части*





репеллер

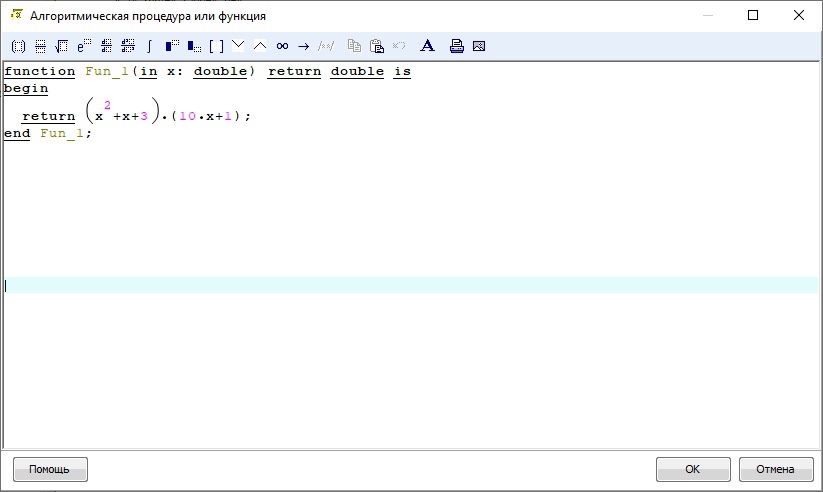
**AnyDinamics**

*Система уравнений и переменные*

**Изображение выглядит как текст

Автоматически созданное описание**

Вспомогательная функция Fun\_1, которая используется для вычисления правой части.

****

Вспомогательная функция Fun\_2, которая используется для нахождения производной правой части.

*Изображение выглядит как текст

Автоматически созданное описание*

***Метод случайного поиска (MvStudium)***

*Изображение выглядит как текст

Автоматически созданное описание*

***Метод случайного поиска (Matlab)***

a=-2\*pi;

b=2\*pi;

b\_x=a;

b\_f=abs(b\_x);

t\_x=0;

t\_f=0;

i=0;

while i<10^5

t\_x=unifrnd(a,b);

t\_f=abs(((t\_x^2)+t\_x+3)\*(10\*t\_x+1));

i=i+1;

if(t\_f<b\_f)

b\_f=t\_f;

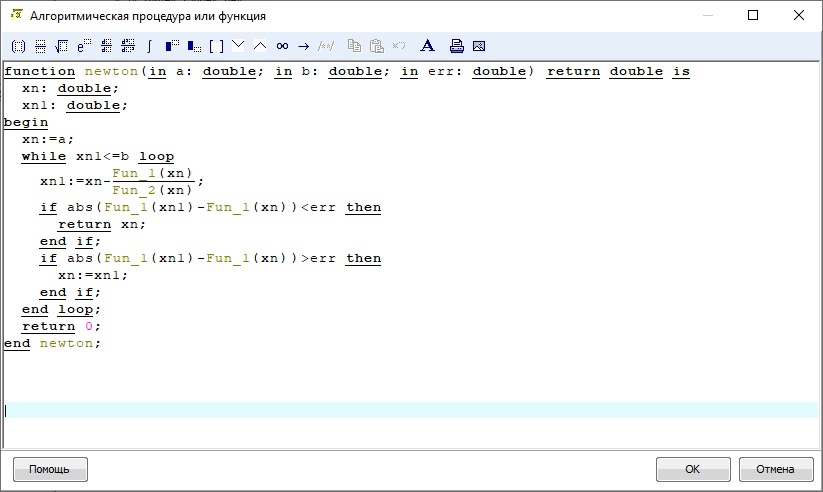
b\_x=t\_x;

end

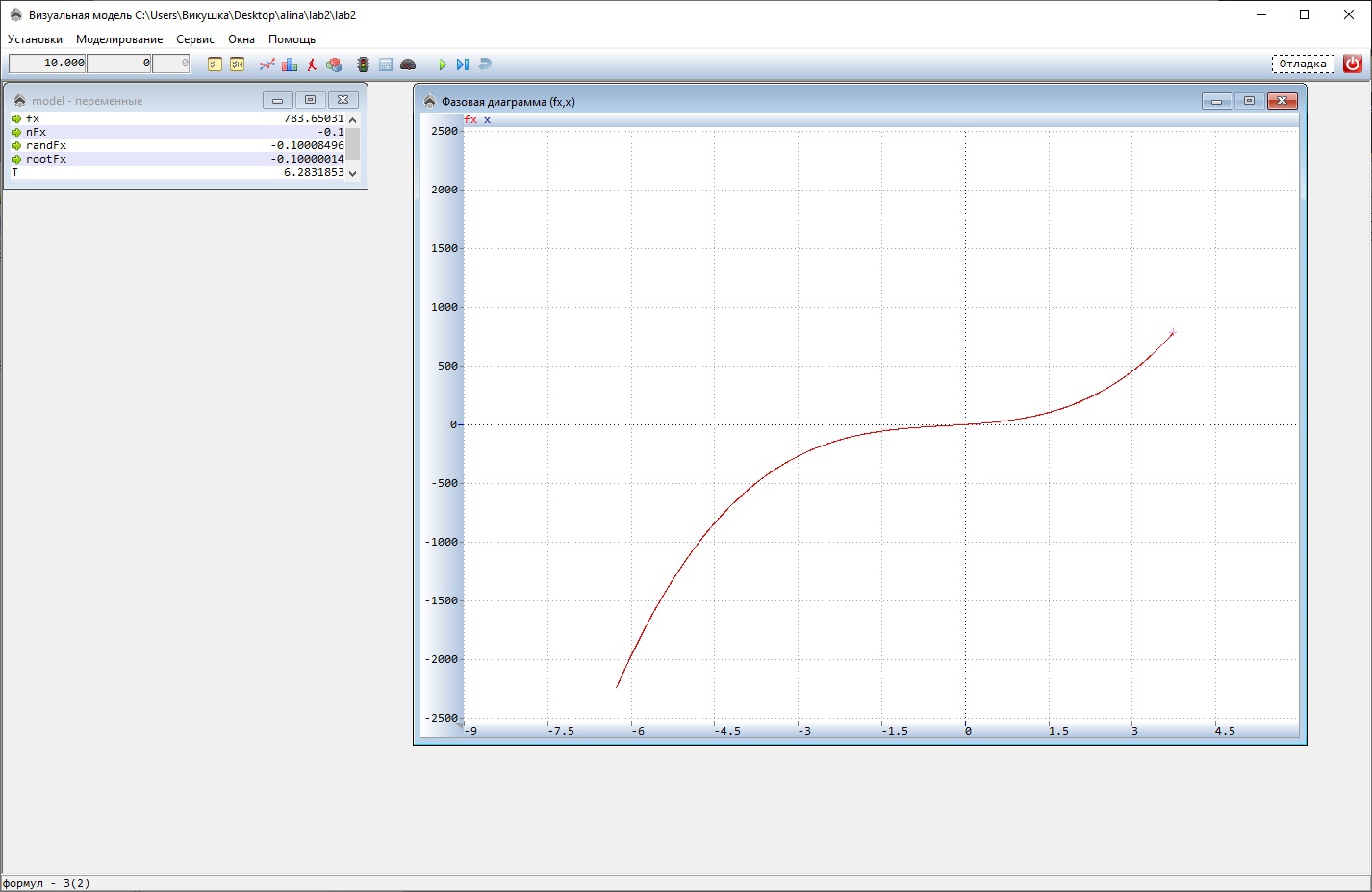
end

sprintf('root=%f', b\_x)

***Метод Ньютона (MvStudium)***



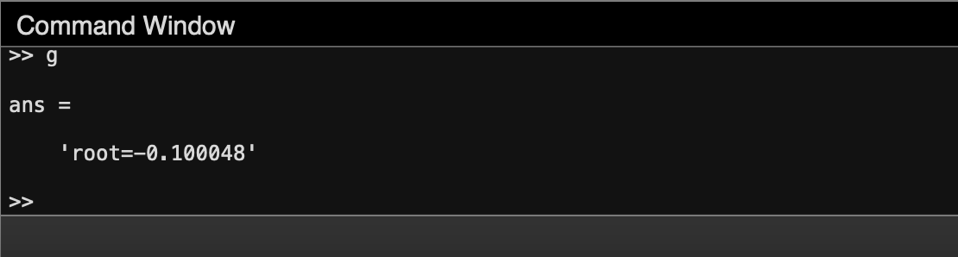
**Результат**

******

←метод Ньютона

←метод случайного поиска

←функция root()

******

←метод случайного поиска

**Вывод**

Значения корней, полученные с помощью метода случайного поиска и метода Ньютона, совпадают со значениями, полученными с помощью математического пакета, с заданной погрешностью.