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).

13	$\frac{dx}{dt} = (a \cdot x^2 + bx + c) \cdot (\alpha \cdot x + \beta);$ $\alpha = 10; \beta = 1; a = 1;$ $b = 1; c = 3$

2. 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

$$x_{n+1} = x_n - f(x_n) \cdot \frac{x_n - x_0}{f(x_n) - f(x_0)};$$

- метод секущих

$$x_{n+1} = x_n - f(x_n) \cdot \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})};$$

- метод Ньютона

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

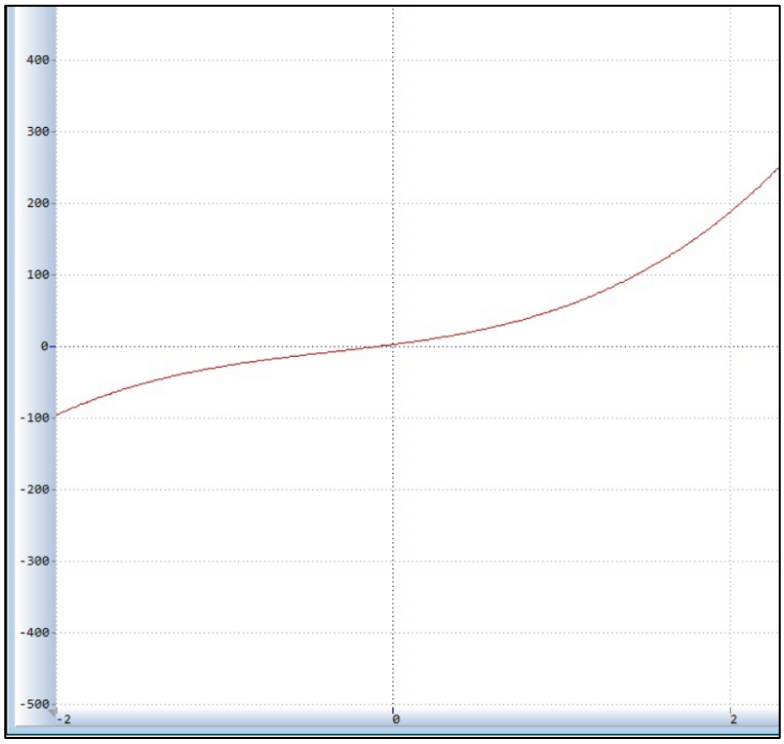
Сравнить вычислительные затраты методов. Процедуры должны иметь те же параметры, что и процедура ZEROIN.

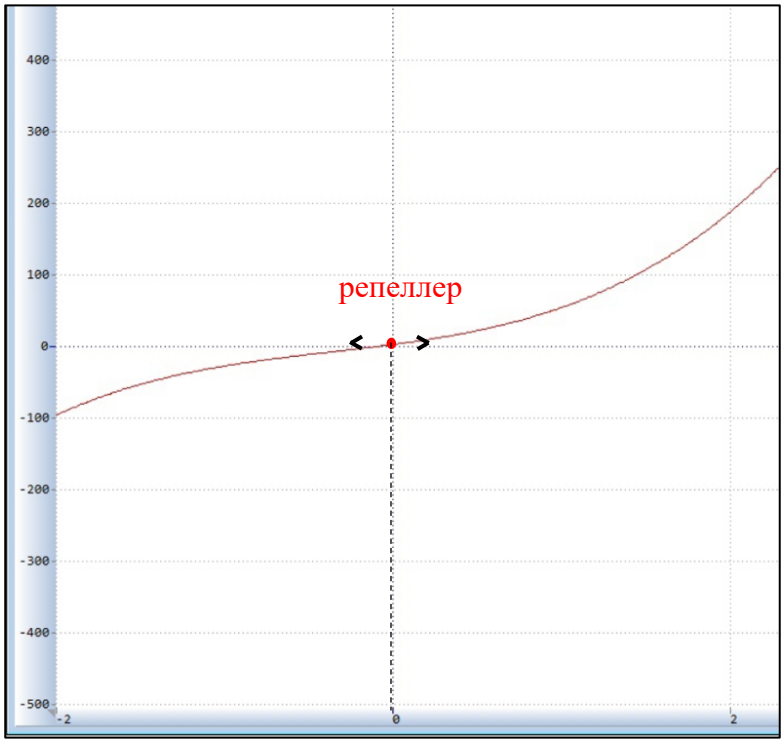
Учебные программы поиска корней функции одной переменной.

13.	Случайный поиск	Matlab	Метод Ньютона	MvStudium

Решение

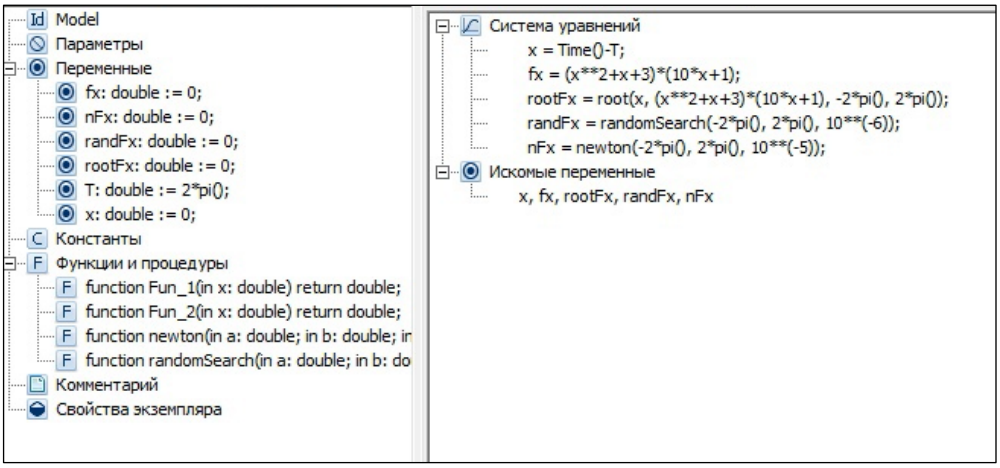
График правой части





AnyDinamics

Система уравнений и переменные



Вспомогательная функция Fun_1, которая используется для вычисления правой части.

```
function Fun_1(in x: double) return double is
begin
  return (x**2+x+3).(10.x+1);
end Fun_1;
```

Вспомогательная функция Fun_2, которая используется для нахождения производной правой части.

```
function Fun_2(in x: double) return double is
begin
  return 30.x**2+22.x+31;
end Fun_2;
```

Метод случайного поиска (MvStudium)

```
function randomSearch(in a: double; in b: double; in err: double) return double is
  b_x: double := a;
  b_f: double := abs(Fun_1(b_x));
  t_x: double := 0;
  t_f: double := 0;
  i: double := 0;
begin
  while i<10**5 loop
    t_x:=uniform(a, b);
    t_f:=abs(Fun_1(t_x));
    i:=i+1;
    if t_f<b_f then
      b_f:=t_f;
      b_x:=t_x;
    end if;
  end loop;
  return b_x;
end randomSearch;
```

Метод случайного поиска (Matlab)

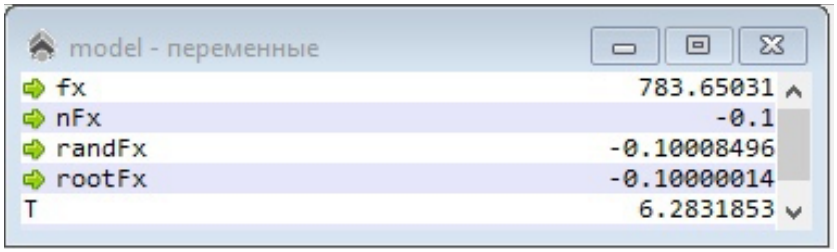
```
a=-2*pi;
b=2*pi;
b_x=a;
b_f=abs(b_x);
t_x=0;
t_f=0;
i=0;
while i<10^5
    t_x=unifrnd(a,b);
    t_f=abs(((t_x^2)+t_x+3)*(10*t_x+1));
    i=i+1;

    if(t_f<b_f)
        b_f=t_f;
        b_x=t_x;
    end
end
sprintf('root=%f', b_x)
```

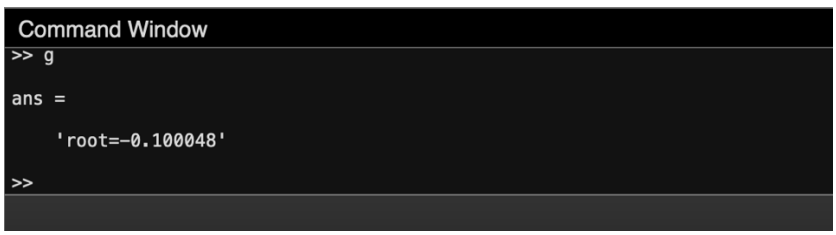
Метод Ньютона (MvStudium)

```
function newton(in a: double; in b: double; in err: double) return double is
  xn: double;
  xn1: double;
begin
  xn:=a;
  while xn1<=b loop
    xn1:=xn-Fun_1(xn)/Fun_2(xn);
    if abs(Fun_1(xn1)-Fun_1(xn))<err then
      return xn;
    end if;
    if abs(Fun_1(xn1)-Fun_1(xn))>err then
      xn:=xn1;
    end if;
  end loop;
  return 0;
end newton;
```

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.