Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление подготовки: 09.03.01 - «Информатика и вычислительная техника»

**Лабараторная работа №1**

**“Методы решения нелинейных уравнений”**

**Вариант №12**

Выполнил студент гр. ИВТ-24-2б

Исмагилов Артур Альбертович

Проверил:

Доц. каф. ИТАС

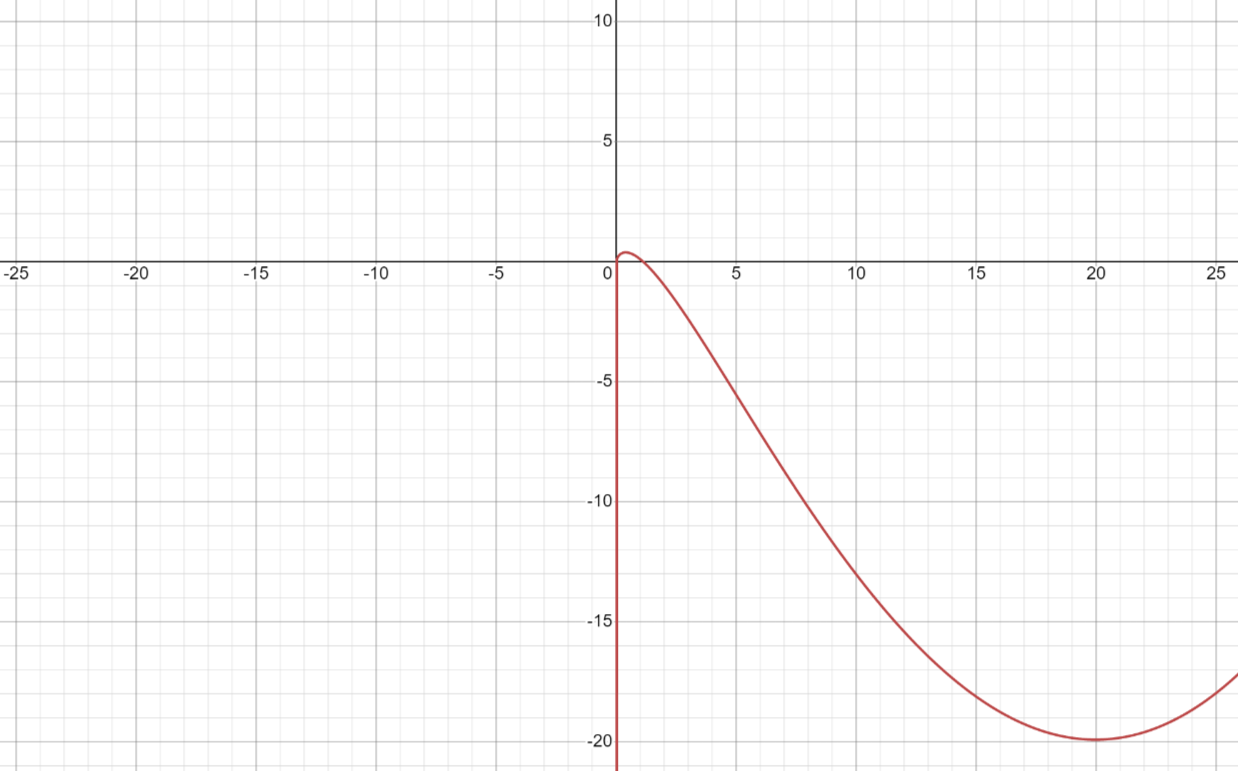
О.А.Полякова.

г.Пермь, 2024

**Постановка задачи**

Решить уравнение: 3 методами: методом Ньютона, итерации и методом половинного деления (дихотомии). C заданной точностью на отрезке

**Геометрическая интерпретация**

****

**Анализ задачи**

**Дано уравнение**  которое нужно решить 3 методами (метод Ньютона, метод половинного деления, метод итераций). Рассмотрим их поподробнее:

**Метод Ньютона**

1. Определение функции:
2. Вычисление производной:

* Первая производная:

1. Подставляем значения промежутка в формулу:

4.Проверяем, если |x(n)-x(n+1)|<(заданной точности), то корнем является x(n) или x(n+1), в противном случае продолжаем итерации, пока длина интервала не станет меньше заданной точности

**Метод половинного деления**

1. Определение функции:
2. Устанавливаем начальные значения интервала и *b*, где функция меняет знак.
3. Итерации:

* Находим среднюю точку интервала *x*:
* Если , то корень найден
* Если , обновляем интервал:
* Иначе:
* Продолжаем итерации, пока длина интервала не станет меньше заданной точности

**Метод итераций**

1. Определение функции:
2. Выражаем x:

f(x)=e^(0.1\*x)

3.Находим производную

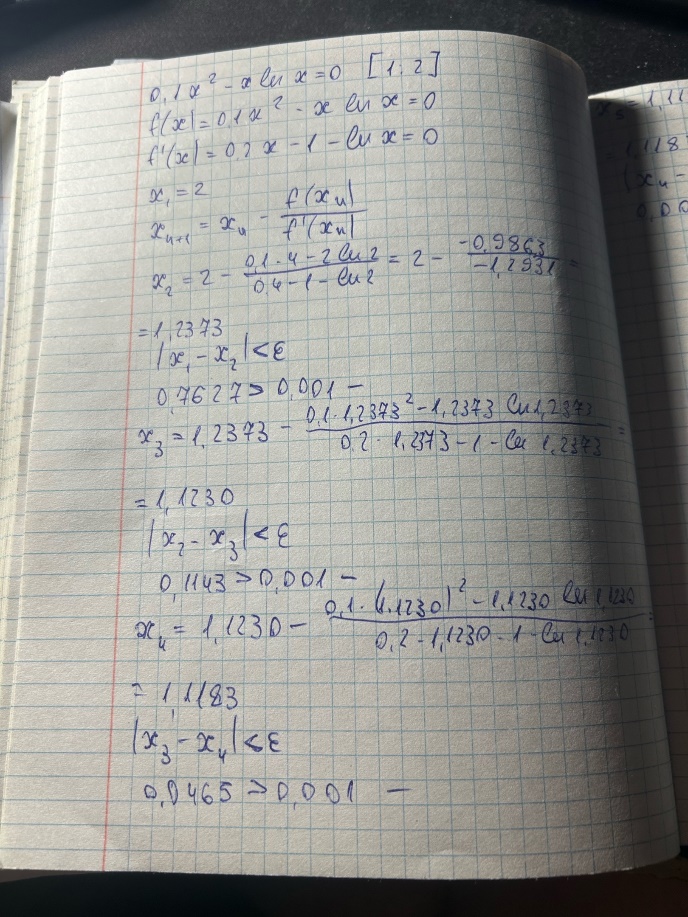
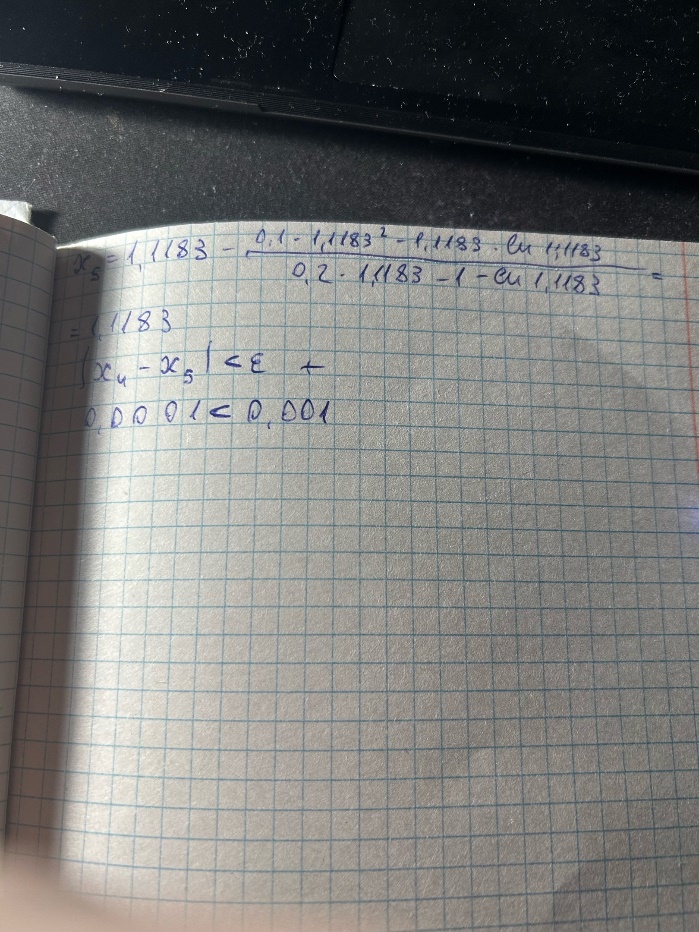
f(x)=(e^(x/10)/10)

4.Подставим значение в производную, результат должен быть меньше 1:

f(1)=0.11<1, следовательно с этим уравнением можно работать

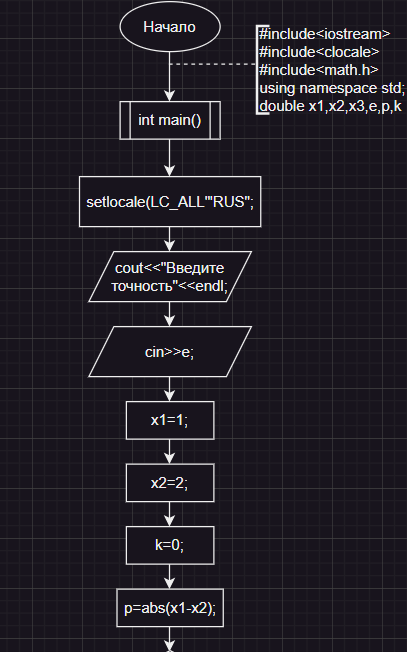
5.Находим x подставляя полученный корень в f(x)=e^(0.1\*x) продолжаем итерации пока разница между значениями по модулю не станет меньше

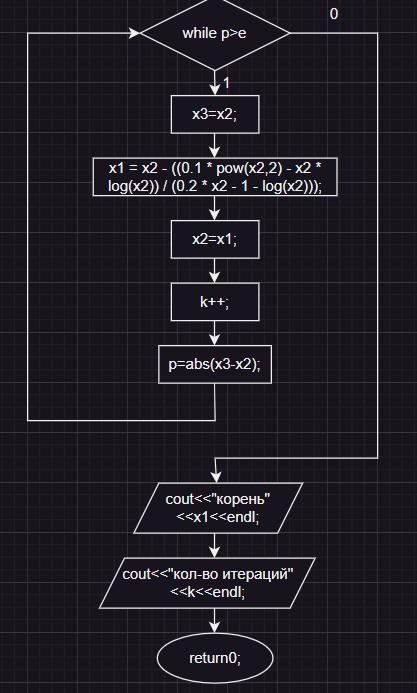
**Поитерационное решение метода Ньютона**

****

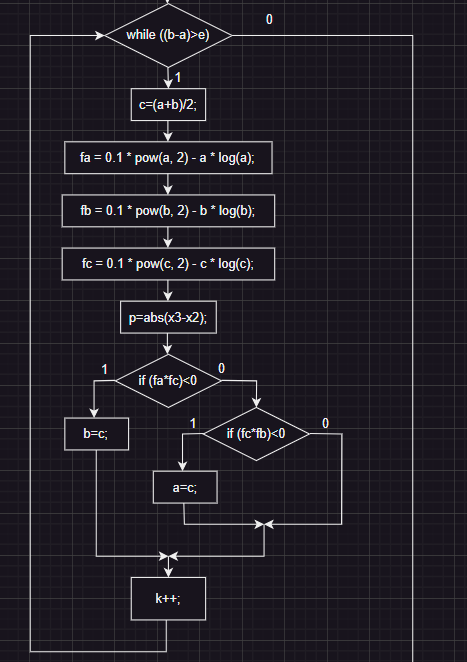
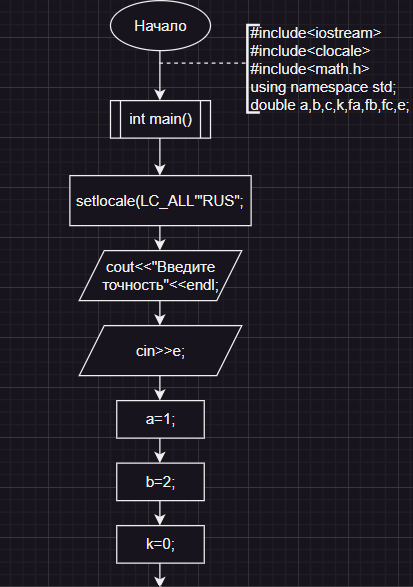
**Блок-схемы**

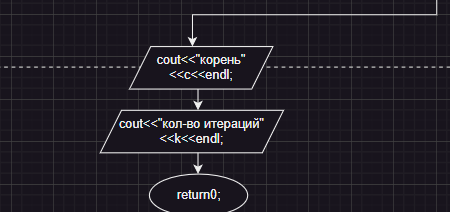
1. **Метод Ньютона**

****

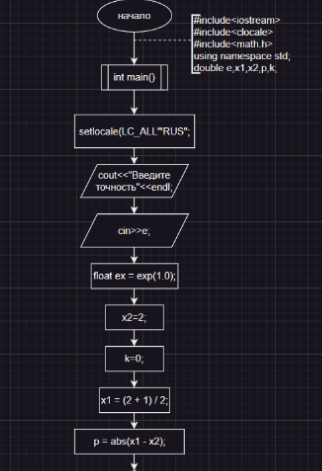
****

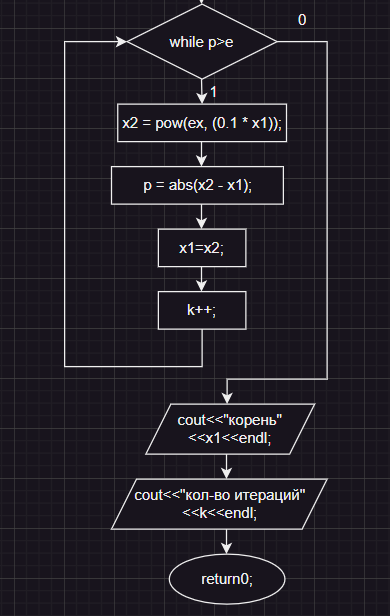
**2.Метод половинного деления**

****

****

**3. Метод итераций**

****

****

**Код программы**

1. **Метод Ньютона**

#include<iostream>

#include<clocale>

#include<math.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUS");

double x1, x2, x3, e, p, k;

cout << "Введите точность" << endl;

cin >> e;

x1 = 1; x2 = 2; k = 0;

p = abs(x1 - x2);

while (p>e)

{

x3 = x2;

x1 = x2 - ((0.1 \* pow(x2,2) - x2 \* log(x2)) / (0.2 \* x2 - 1 -log(x2)));

x2 = x1;

k++;

p = abs(x3 - x2);

}

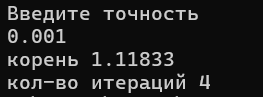
cout << x1 << endl;

cout << k;

return 0;

}

Вывод программы



**2.Метод половинного деления**

#include<iostream>

#include<clocale>

#include<math.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUS");

double a, b, c, k, fa, fb, fc, e;

cout << "введите точность" << endl;

cin >> e;

a = 1; b = 2; k = 0;

while ((b - a) > e)

{

c = (a + b) / 2;

fa = 0.1 \* pow(a, 2) - a \* log(a);

fb = 0.1 \* pow(b, 2) - b \* log(b);

fc = 0.1 \* pow(c, 2) - c \* log(c);

if (fa \* fc < 0)

{

b = c;

}

else

{

if (fc \* fb < 0)

{

a = c;

}

}

k++;

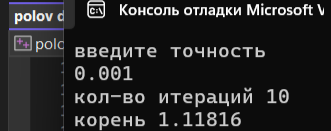
}

cout << "кол-во итераций " <<k << endl;

cout << "корень "<< c << endl;

}

Вывод программы



**3.Метод итераций**

#include <iostream>

#include <math.h>

#include <clocale>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUS");

double e, x1, x2, p, k;

float ex = exp(1.0);

k = 0;

x2 = 2;

cout << "Введите точность" << endl;

cin >> e;

x1 = (2 + 1) / 2;

p = abs(x1 - x2);

while (p > e)

{

x2 = pow(ex, (0.1 \* x1));

p = abs(x2 - x1);

x1 = x2;

k++;

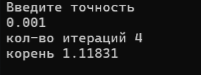
}

cout <<"кол-во итераций "<< k << endl;

cout <<"корень "<< x1 << endl;

}

Вывод программы



Ссылка на гит хаб -https://github.com/ArturIsm82