Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

Лабораторная работа

**“Нахождение приближённых значений корней нелинейных уравнений”**

Выполнил:

студент группы РИС-23-1б

Жуланов Никита Андреевич

Проверила:

доцент кафедры ИТАС

О. А. Полякова

2023 г.

**Постановка задачи:**

Дано уравнение , необходимо найти приближённое значение корня этого уравнения тремя методами (половинного деления, Ньютона и итерационным), для каждого сделать анализ, блок-схему и геометрическую интерпретацию. Отрезок, содержащий корень: [1;2]. Точное значение для самопроверки: 1,3077.

**Решение методом половинного деления**

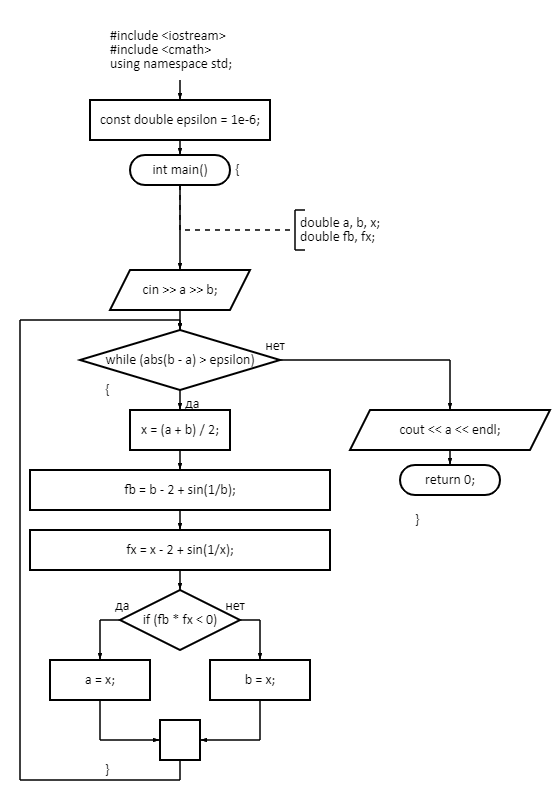
**Анализ:**

1. На отрезке функция монотонна и непрерывна => метод применим.
2. *f(1) \* f(2) < 0* => корень существует на отрезке.
3. Пусть *x –* середина отрезка; *f(a) \* f(x) < 0* => правая граница *b* получает значение *x*; иначе левая граница получает значение *x*.
4. Повторить 3 пункт, пока не достигнута нужная точность.

**Геометрическая интерпретация:**

|  |  |
| --- | --- |
|  |  |
|  |  |

**Блок-схема:**

****

**Решение методом Ньютона**

**Анализ:**

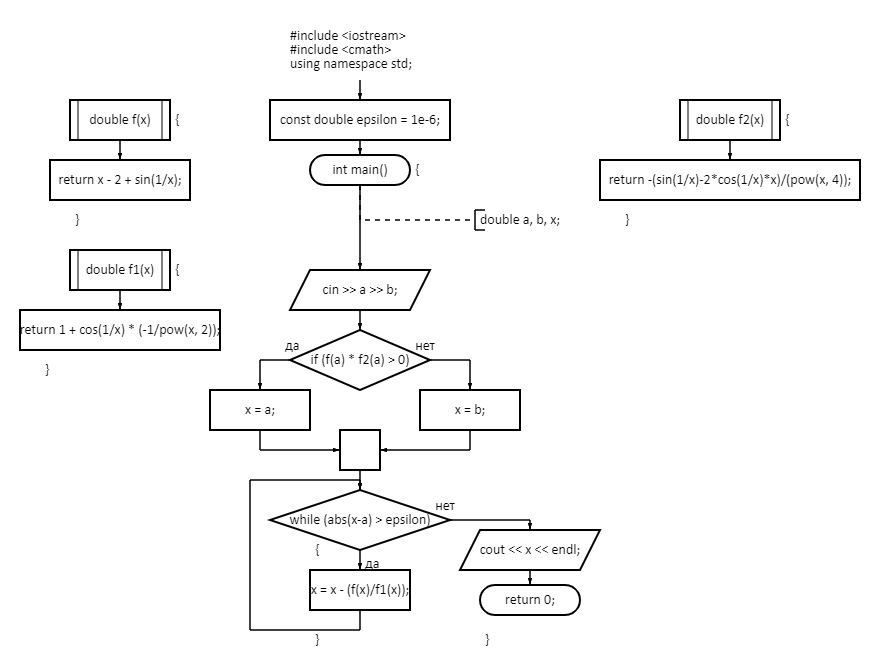
1. На отрезке функция монотонна и непрерывна, существует *f’’(x), f’(x) != 0, f’(x) > 0 и f’’(x) > 0* => метод применим.
2. *f(2) \* f’’(2) > 0 =>* начальное значение *x0* = 2.
3. Найти пересечение касательной в точке *x0* с осью *x* и в этой точке провести новую касательную.
4. Повторить 3 пункт, пока не достигнута нужная точность.

**Геометрическая интерпретация:**

|  |  |
| --- | --- |
|  |  |

Функция, касательная, 1 производная, 2 производная

**Блок-схема:**

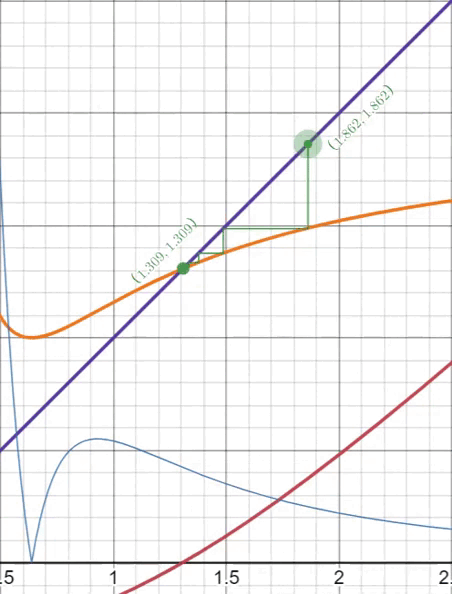
****

**Решение методом итераций**

**Анализ:**

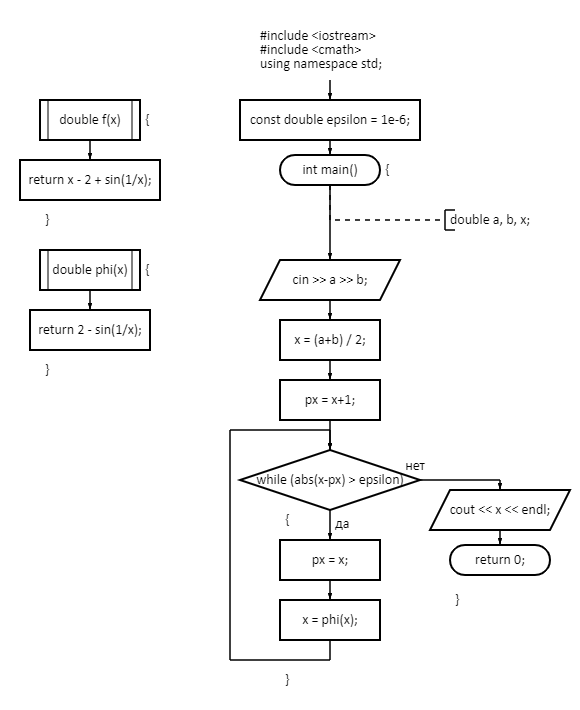
1. Выразить функцию .
2. Так как , метод применим.
3. Выбрать начальное значение *x* из отрезка.
4. Подставить *x* в функцию .
5. Повторить 4 пункт, пока не достигнута нужная точность.

**Геометрическая интерпретация:**



<https://www.desmos.com/calculator/w297tto9zy?lang=ru>

**Блок-схема:**



**Код на С++:**

#include <iostream>

#include <cmath>

using namespace std;

const double epsilon = 1e-6;

double f(double x) {

    return x - 2 + sin(1/x);

}

double f1(double x) {

    return 1 + cos(1/x)\*(-1/pow(x, 2));

}

double f2(double x) {

    return -(sin(1/x)-2\*cos(1/x)\*x)/pow(x, 4);

}

double phi(double x) {

    return 2 - sin(1/x);

}

int bisection(double a, double b) {

    double xn;

    double fb, fx;

    int i = 1;

    while (abs(b-a) > epsilon) {

        xn = (a + b) / 2;

        cout << "Шаг " << i << ":\t" << xn << endl;

        ++i;

        fb = f(b);

        fx = f(xn);

        if (fb \* fx < 0) {a = xn;} else {b = xn;}

    }

    cout << xn << endl;

    return 0;

}

int newtone(double a, double b) {

    double xn, px;

    int i = 1;

    if (f(a)\*f2(a) > 0) {xn = a;} else {xn = b;}

    px = xn + 1;

    while (abs(px-xn) > epsilon) {

        px = xn;

        xn = xn - f(xn)/f1(xn);

        cout << "Шаг " << i << ":\t" << xn << endl;

        ++i;

    }

    cout << xn << endl;

    return 0;

}

int iteration(double a, double b) {

    double px, xn = (a+b) / 2;

    int i = 1;

    px = xn+1;

    while (abs(px-xn) > epsilon) {

        px = xn;

        xn = phi(xn);

        cout << "Шаг " << i << ":\t" << xn << endl;

        ++i;

    }

    cout << xn << endl;

    return 0;

}

int main() {

    setlocale(LC\_ALL, "Russian");

    double l, r;

    cin >> l >> r;

    cout << "Метод половинного деления" << endl;

    bisection(l, r);

    cout << "Метод Ньютона" << endl;

    newtone(l, r);

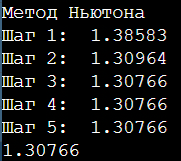
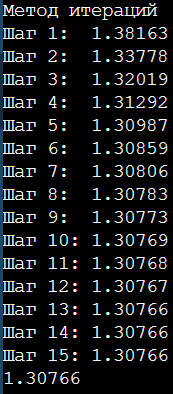
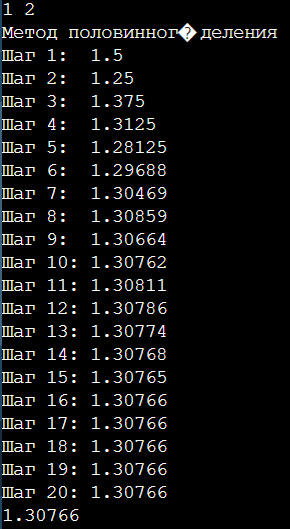
    cout << "Метод итераций" << endl;

    iteration(l, r);

    return 0;

}

**Результаты:**

****

**Вывод:**

У меня получилось реализовать все 3 алгоритма нахождения приближённого значения.