Федеральное агентство по образованию

Государственное общеобразовательное учреждение высшего профессионального образования

**Пермский государственный технический университет**

**Лабораторная работа:**

Решение нелинейных уравнений

Выполнил:

студент группы РИС-23-2б

Колосов Данил Евгеньевич

Проверила:

доцент кафедры ИТАС

Ольга Андреевна Полякова

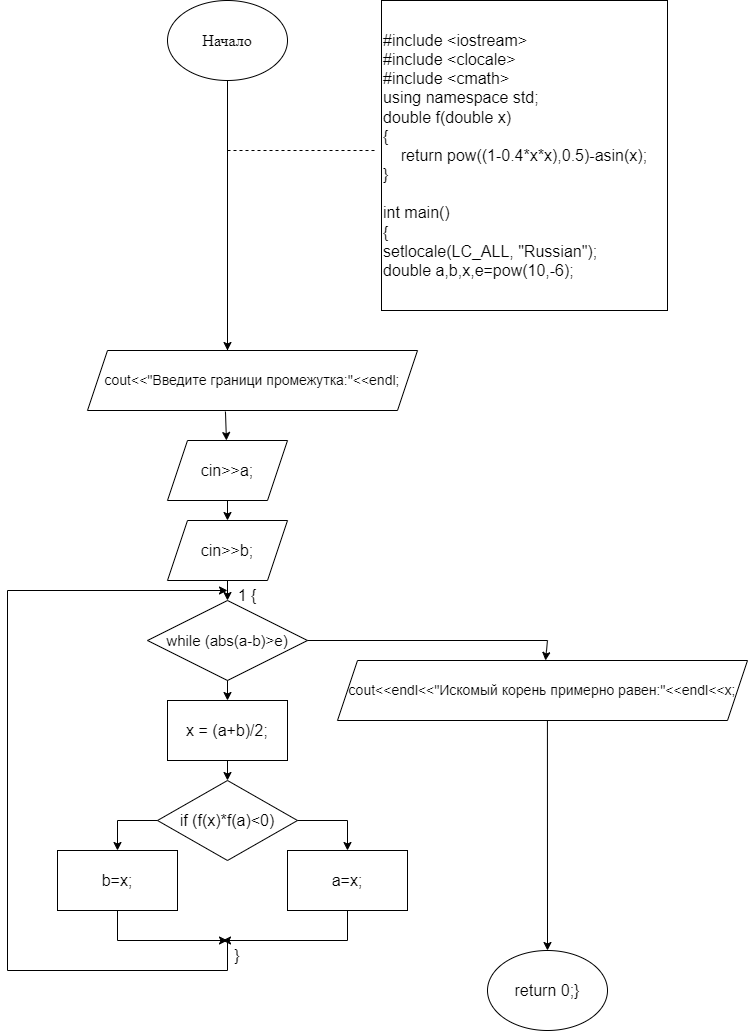
2023 г.

Метод половинного деления

Дано уравнение функции t6 и границы интервала ab, на котором есть корень, задана точность .

Построение алгоритма для решения нелинейных уравнений методом половинного деления

1. На интервале выбирается равный
2. Находим значения функции в токах a, b, x.
3. Выбираем тот из дух равных интервалов (ax или xb), где произведение значений функции на концах интервала будет больше нуля.
4. Присваиваем a или b, в зависимости от выбранного интервала, значение x.
5. Выполняем предыдущие шаги до того момента пока не будет выполнено условие проверки

Блок схема

0

Код программы

#include <iostream>

#include <clocale>

#include <cmath>

using namespace std;

double f(double x)

{

return pow((1-0.4\*x\*x),0.5)-asin(x);

}

int main()

{

setlocale(LC\_ALL, "Russian");

double a,b,x,e=pow(10,-6);

cout<<"Введите границы промежутка:"<<endl;

cin>>a;

cin>>b;

while (abs(a-b)>e)

{

x = (a+b)/2;

cout<<x<<endl;

if (f(x)\*f(a)<0)

{b=x;}

else

a=x;

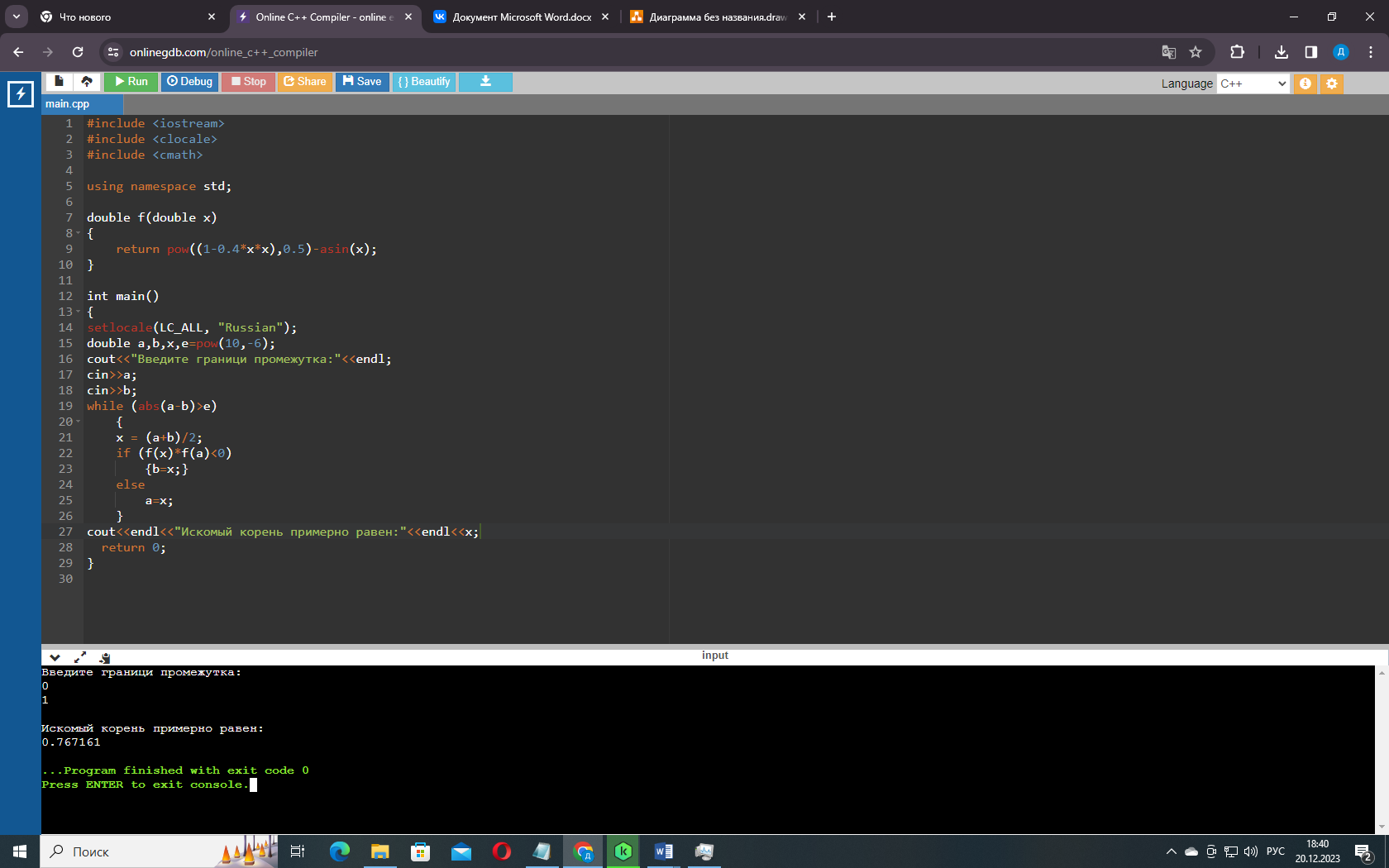
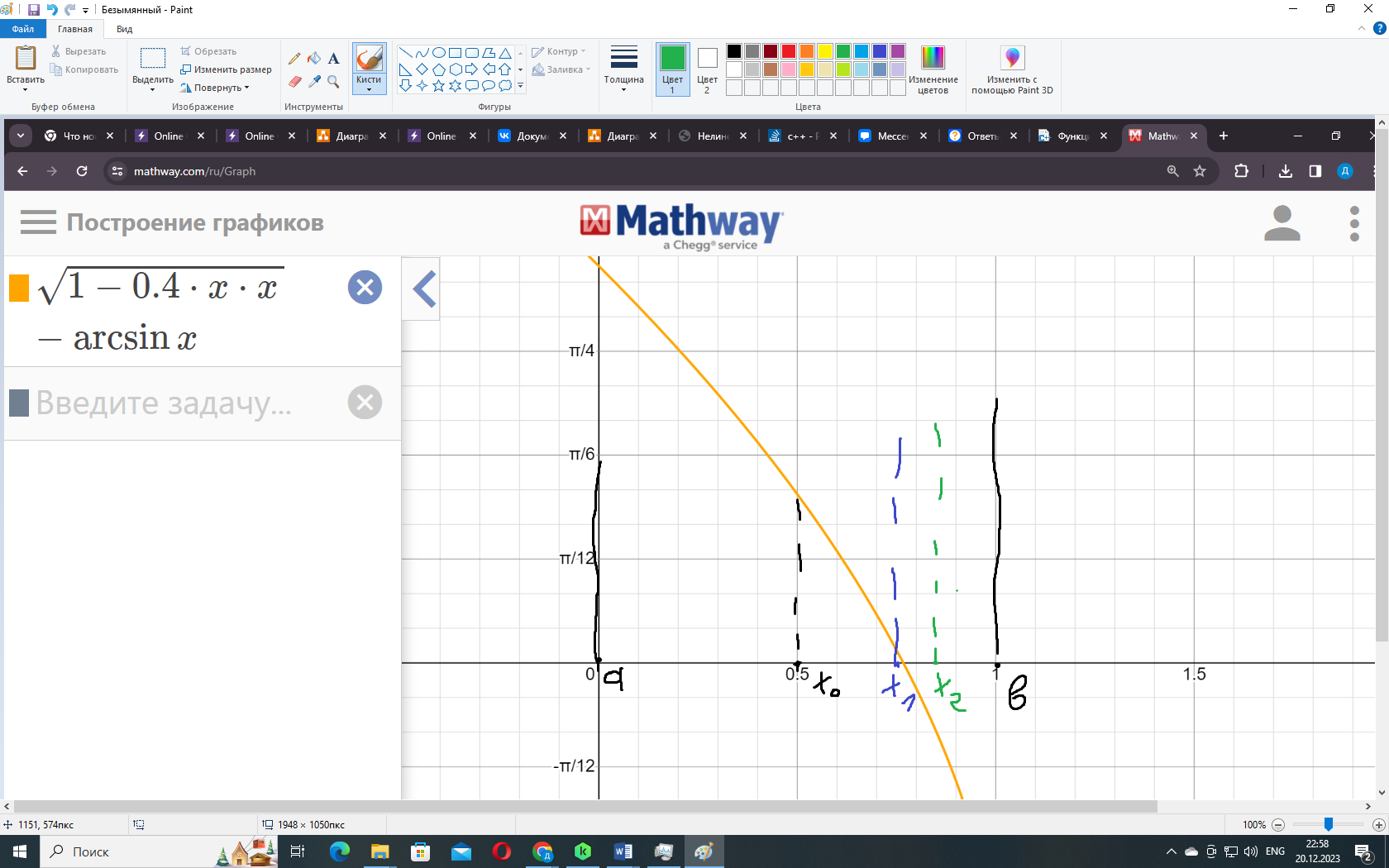
}

cout<<endl<<"Искомый корень примерно равен:"<<endl<<x;

return 0;

}

Результаты работы

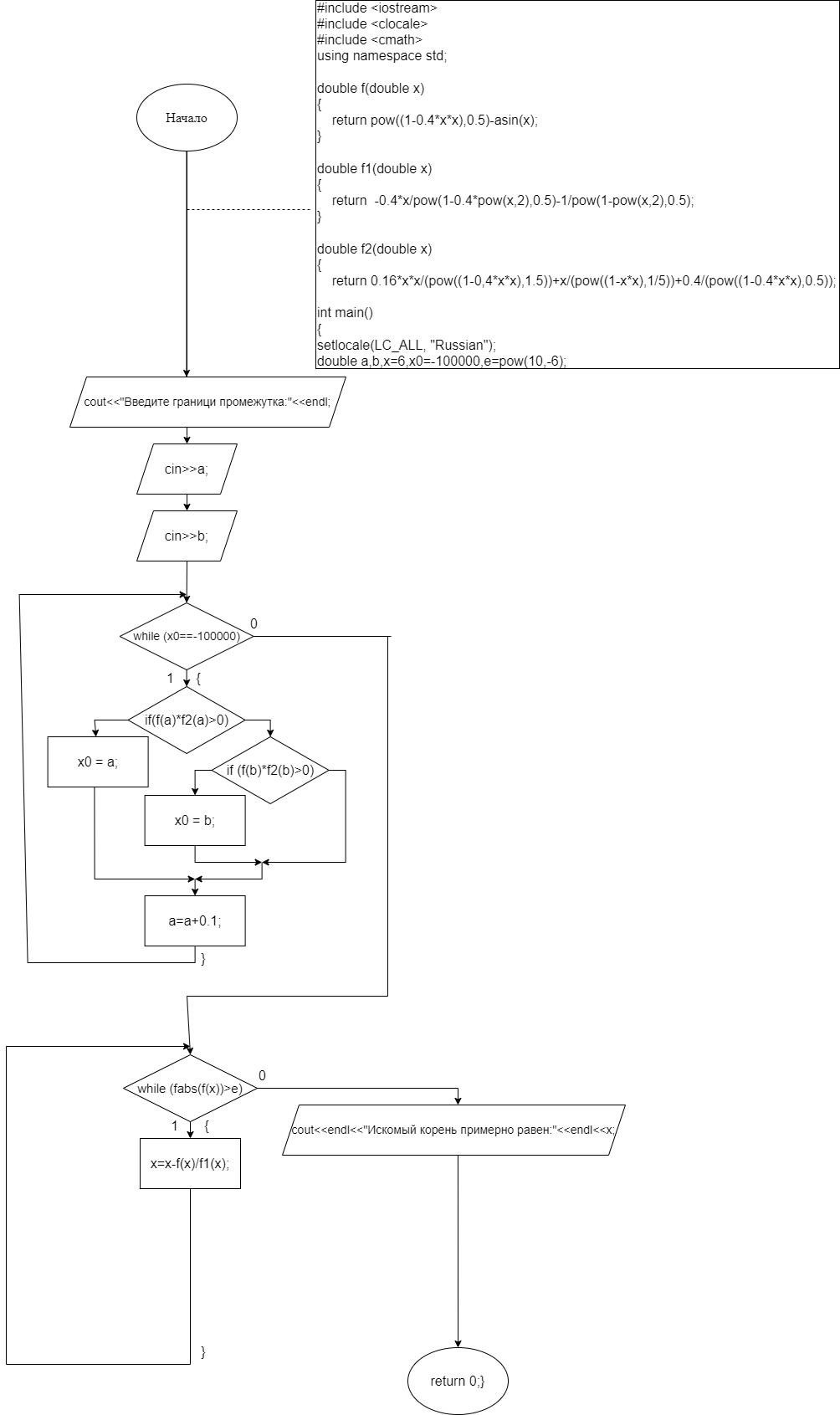


Метод Ньютона

Дано уравнение функции t6 и границы интервала ab, на котором есть корень, задана точность .

Построение алгоритма для решения нелинейных уравнений методом Ньютона

1. Находится первая и вторая производные функции
2. За начальное приближение выбирается левый конец интервала, если удовлетворяет неравенству и правый, если он удовлетворяет неравенству . Если оба конца не подходят, то берётся любая точка на интервале, удовлетворяющая неравенству.
3. Последующие приближения находятся по формуле до техх пор пока не вополнится условие

Блок схема

Код программы на С++

#include <iostream>

#include <clocale>

#include <cmath>

using namespace std;

double f(double x)

{

return pow((1-0.4\*x\*x),0.5)-asin(x);

}

double f1(double x)

{

return -0.4\*x/pow(1-0.4\*pow(x,2),0.5)-1/pow(1-pow(x,2),0.5);

}

double f2(double x)

{

return 0.16\*x\*x/(pow((1-0,4\*x\*x),1.5))+x/(pow((1-x\*x),1/5))+0.4/(pow((1-0.4\*x\*x),0.5));//(0.4\*pow((1-0.4\*pow(x,2)),0.5)+0.16\*x\*x/pow(1-0.4\*pow(x,2),0.5)+x/(pow(1-pow(x,2),1.5)))/(1-0.4\*pow(x,2));

}

int main()

{

setlocale(LC\_ALL, "Russian");

double a,b,x=-100000,e=pow(10,-6);

cout<<"Введите границы промежутка:"<<endl;

cin>>a;

cin>>b;

while (x==-100000){

if (f(a)\*f2(a)>0)

{x = a;}

else

{

if (f(b)\*f2(b)>0)

x = b;}

a=a+0.1;

}

while(fabs(f(x))>e)

{

x=x-f(x)/f1(x);

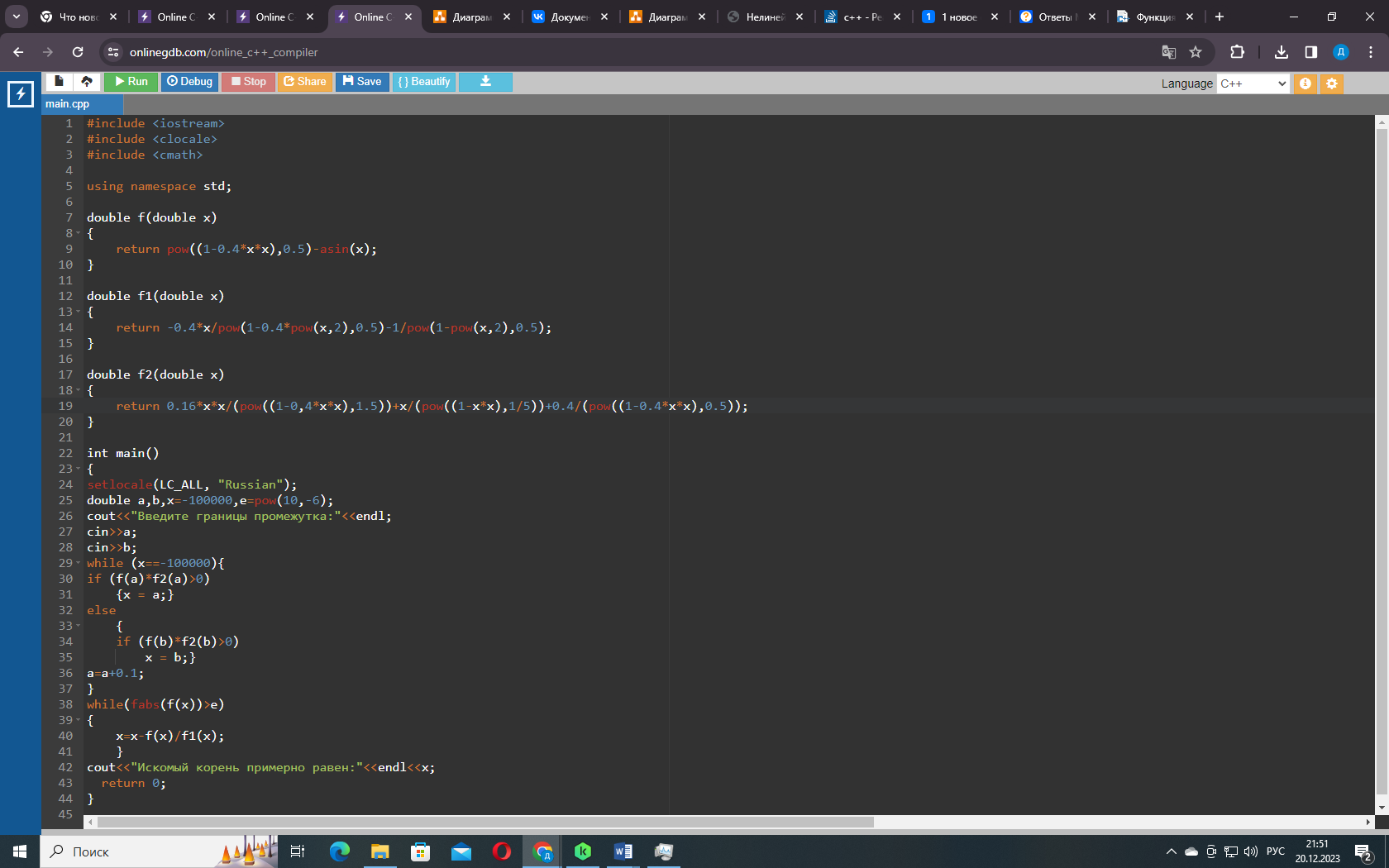
cout<<x<<endl;

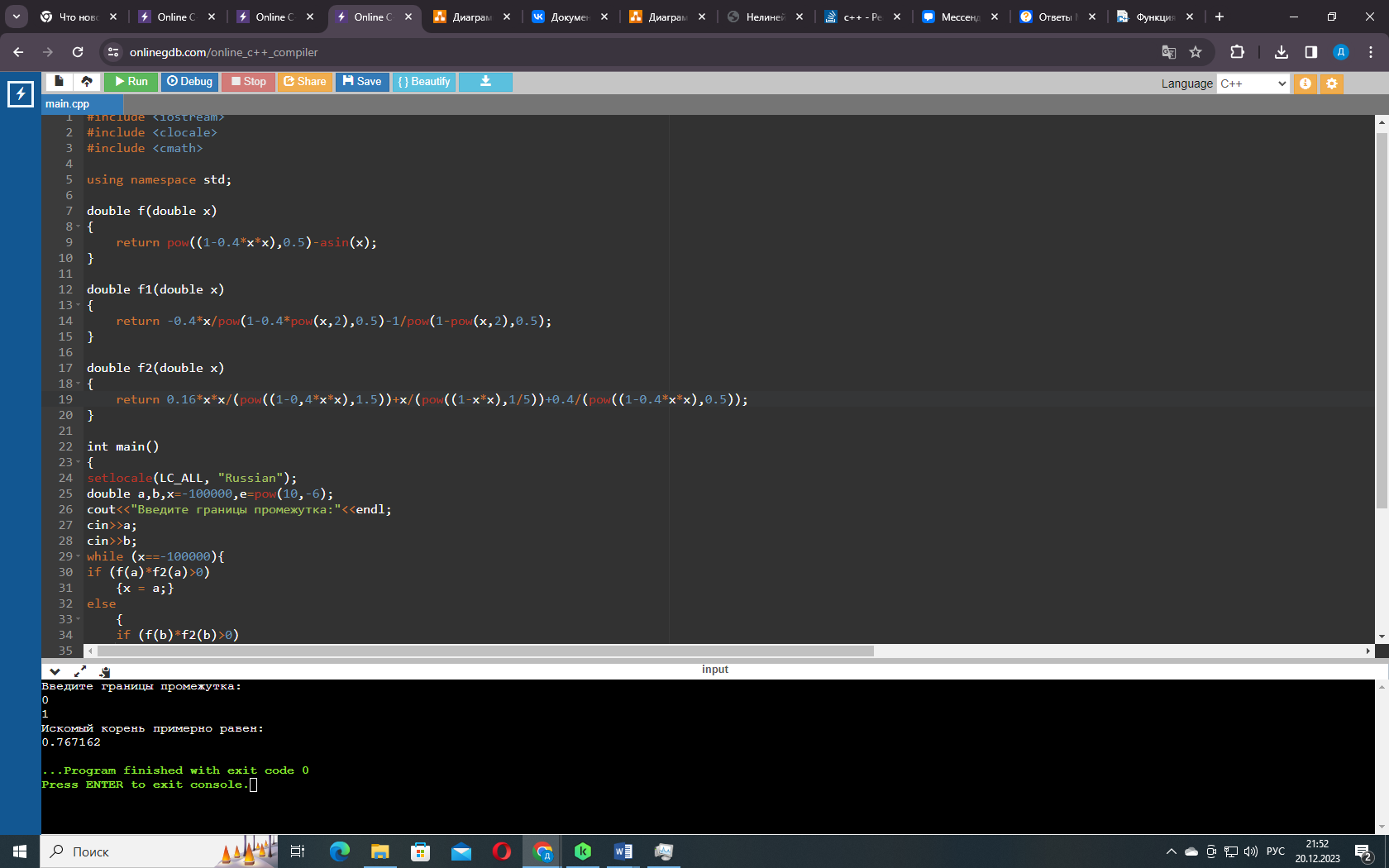
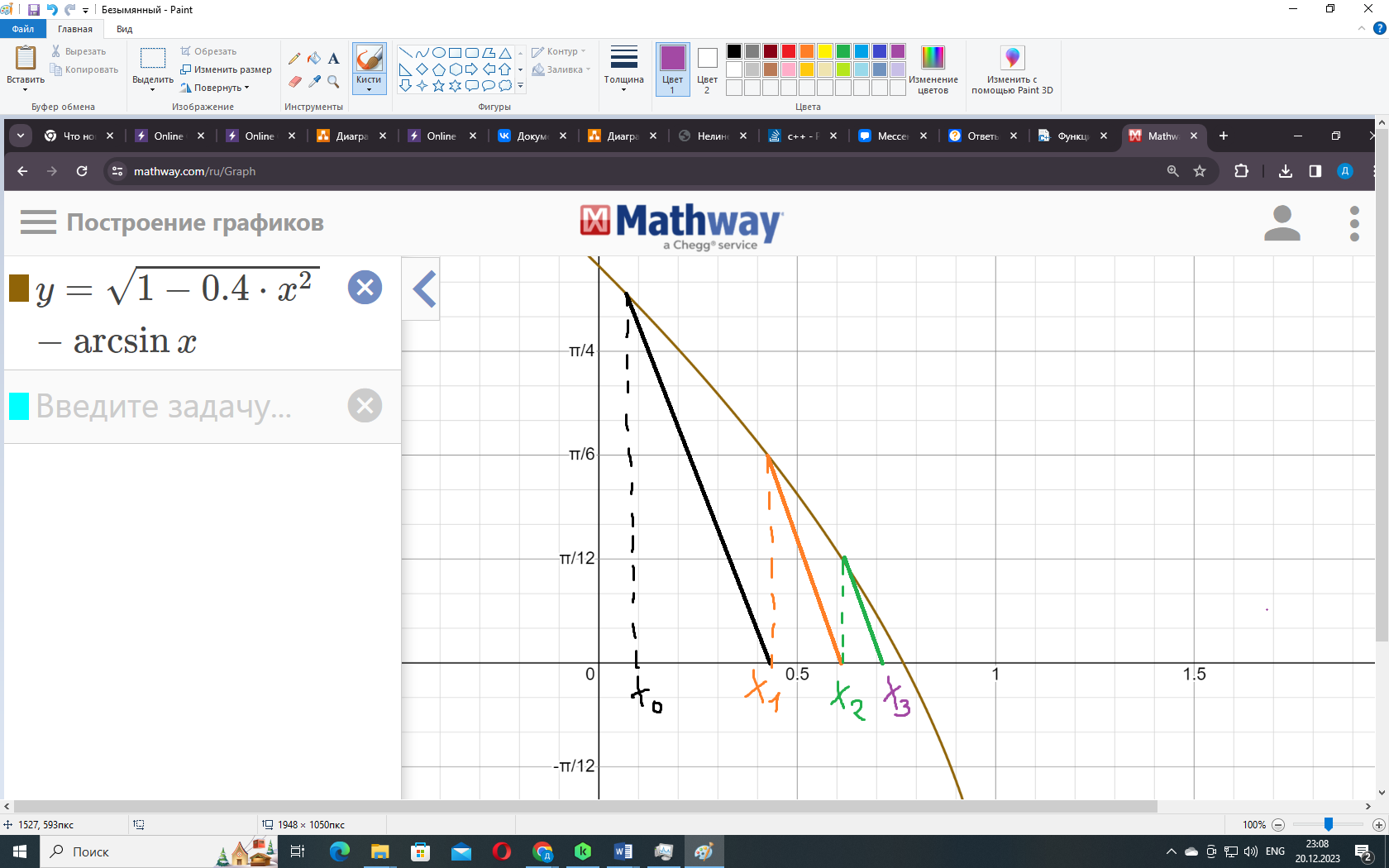
}

cout<<"Искомый корень примерно равен:"<<endl<<x;

return 0;

}

Пример работы

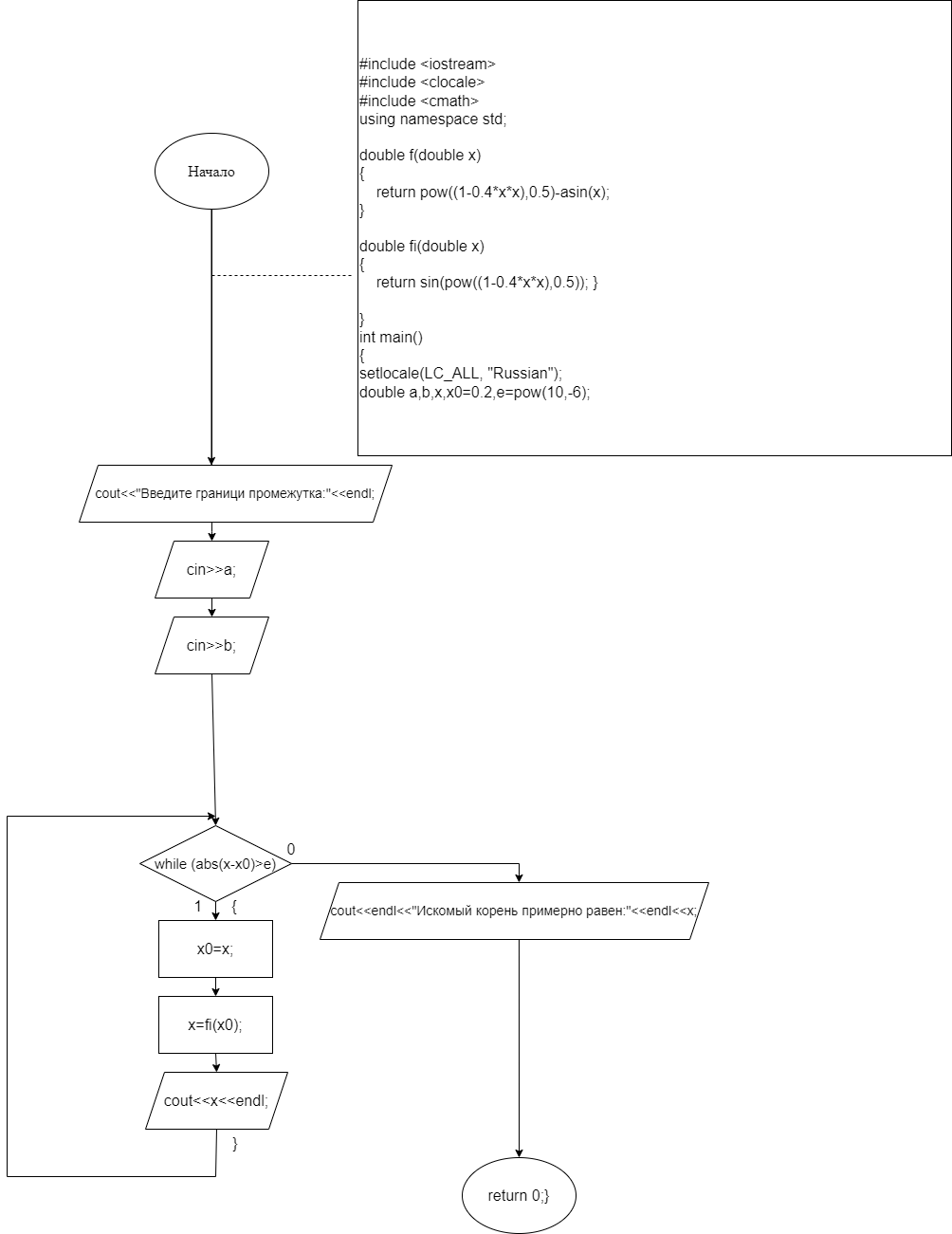


Метод итерации

Дано уравнение функции t6 и границы интервала ab, на котором есть корень, задана точность .

Построение алгоритма для решения нелинейных уравнений методом итерации

1. Представить данную функцию в виде
2. Выбрать произвольное начальное приближение (например x=0.1)
3. Высчитывать последующие приближения по формуле , пока не выполнится условие

Блок схема

Код программы на с++

#include <iostream>

#include <clocale>

#include <cmath>

using namespace std;

double f(double x)

{

return pow((1-0.4\*x\*x),0.5)-asin(x);

}

double fi(double x)

{

return sin(pow((1-0.4\*x\*x),0.5)); }

int main()

{

setlocale(LC\_ALL, "Russian");

double a,b,x,x0=0.2,e=pow(10,-6);

cout<<"Введите границы промежутка:"<<endl;

cin>>a;

cin>>b;

while (abs(x-x0)>e){

x0=x;

x=fi(x0);

cout<<x<<endl;

}

cout<<"Искомый корень примерно равен:"<<endl<<x;

return 0;

}

Результат работы

