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

ПНИПУ

Лабораторная работа  
“Приблизительное вычисление корней(вар 8)”

Выполнил:   
студент группы РИС-23-1б   
Мокрушин Никита Дмитриевич

Проверила:   
доцент кафедры ИТАС   
О.А. Полякова

Пермь, 2023 г.

**Поставновка задачи:** вычислить, имея функцию y=f(x) с помощью 3-х методов приблизительного вычисления корней, корни уравнения.

y=x- , y=0

**Используемые функции:**

y= x-

y`=

y``=

x= (x)=

Решение x- = 0

X 0.262441

**Исходные данные:**

x- = 0

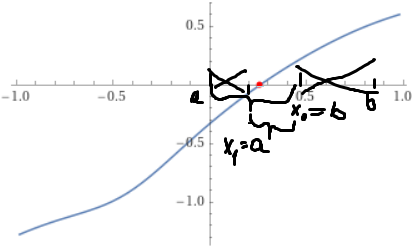
отрезок [0;0,85]

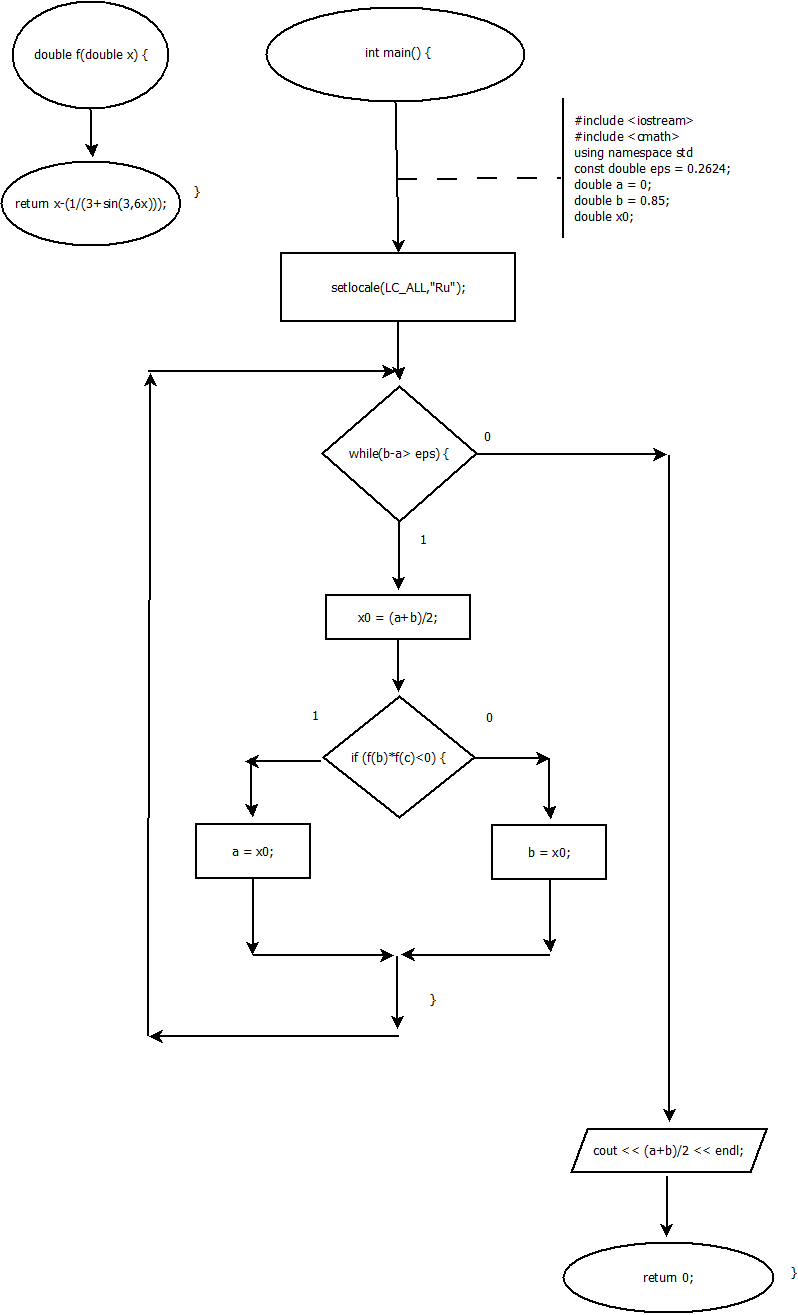
=0,2624

**Решение:**

**Метод половинного деления:**

Геометрическая интерпритация:

****

Блок-схема:

Код на языке С++:

#include <iostream>

#include <cmath>

using namespace std;

const double eps = 0.2624;

double f(double x) {

return x - (1 / (3 + sin(3.6 \* x)));

}

int main()

{

double a, b, x0;

a = 0;

b = 0.85;

while (b - a> eps )

{

x0 = (a + b) / 2;

if (f(b) \* f(x0) < 0) {

a = x0;

}

else

{

b = x0;

}

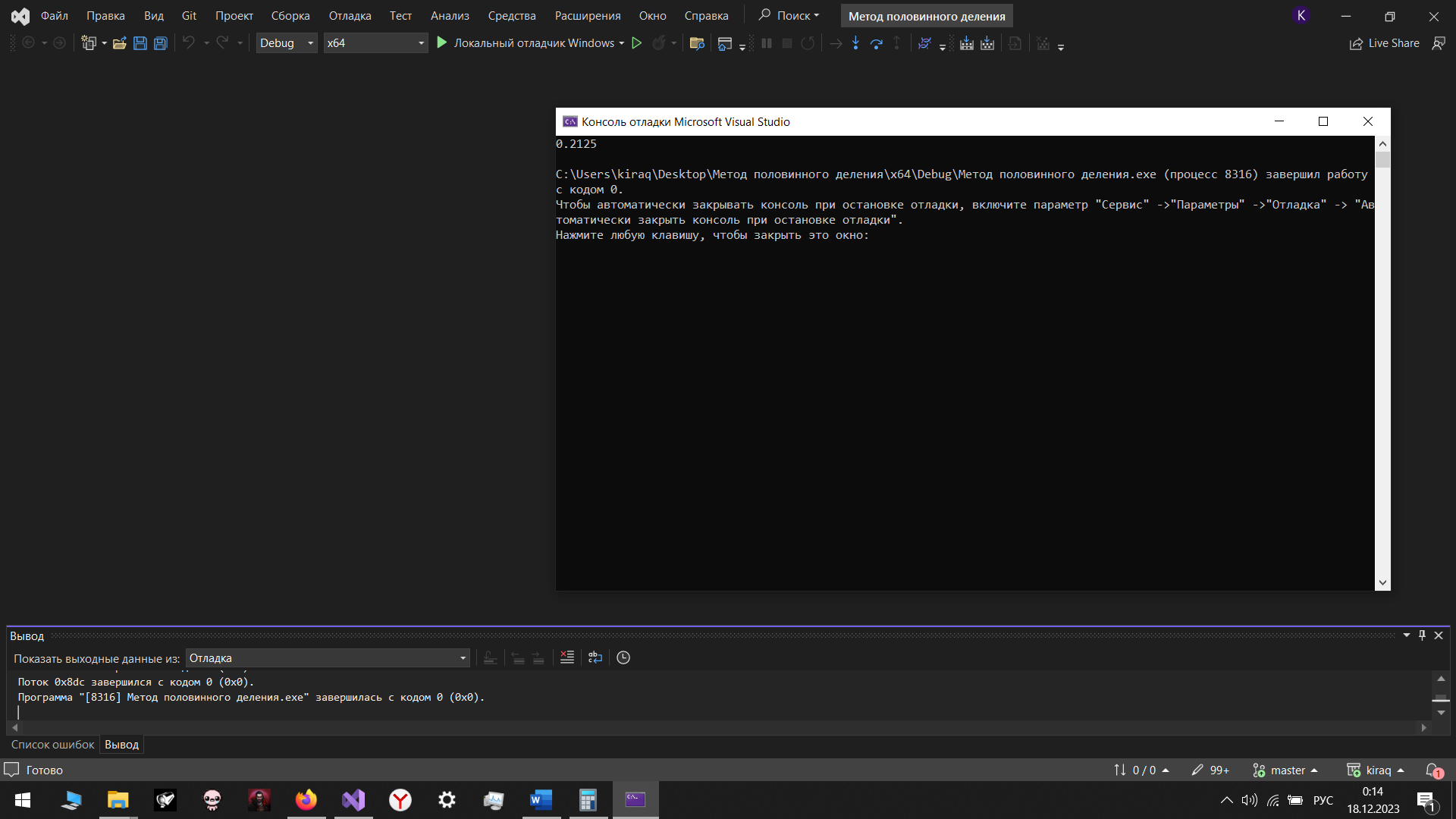
}

cout << (a + b) / 2 << endl;

return 0;

}

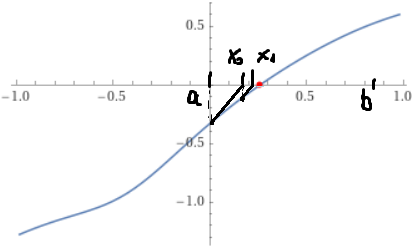
Работа программы:



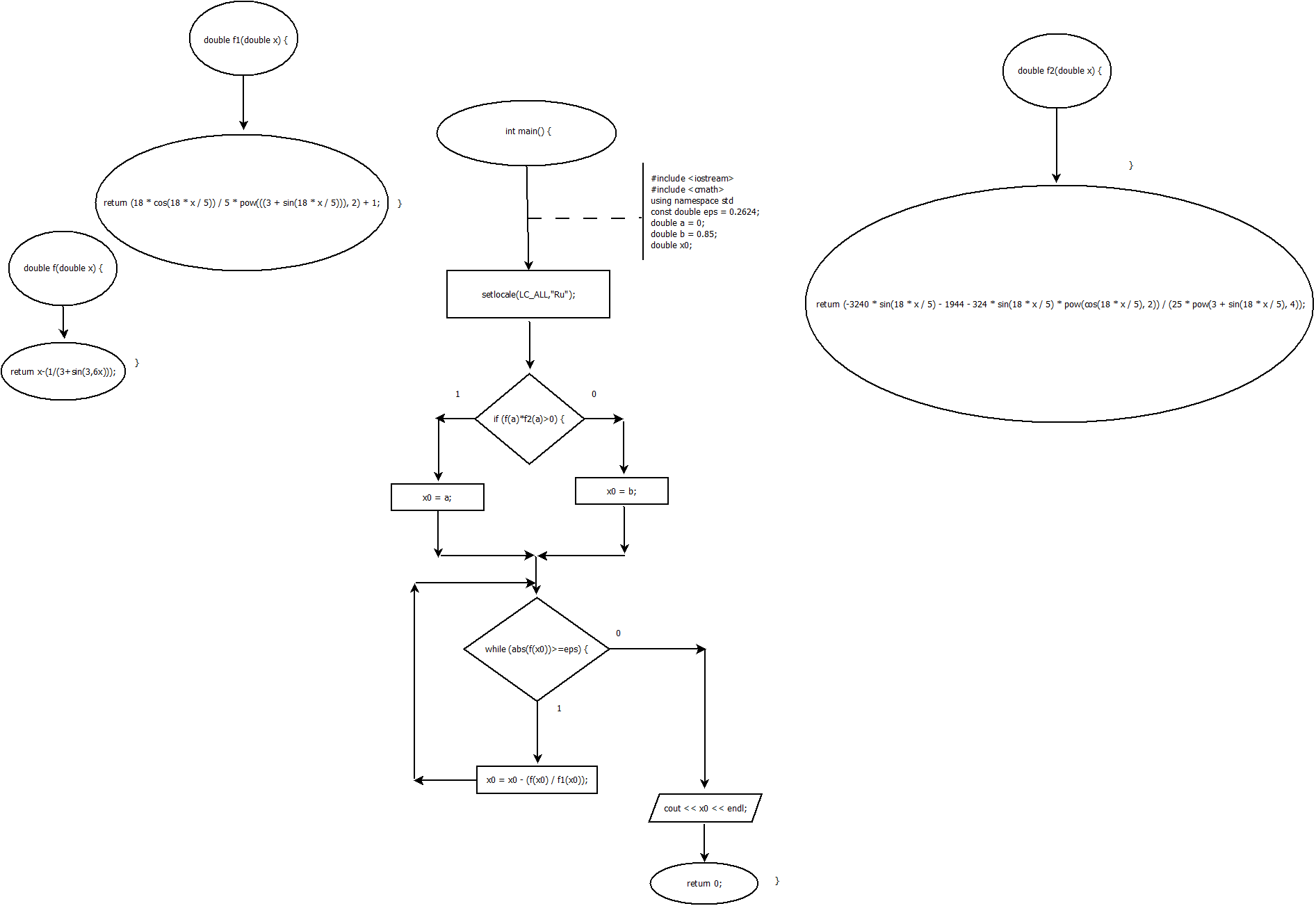
Погрешность = 0.049941

**Метод Ньютона:**

Геометрическая интерпритация:



Блок-схема:



Код на языке С++:

#include <iostream>

#include <cmath>

using namespace std;

const double eps = 0.2624;

double f(double x) {

return x - (1 / (3 + sin(3.6 \* x)));

}

double f1(double x) {

return (18 \* cos(18 \* x / 5)) / 5 \* pow(((3 + sin(18 \* x / 5))), 2) + 1;

}

double f2(double x) {

return (-3240 \* sin(18 \* x / 5) - 1944 - 324 \* sin(18 \* x / 5) \* pow(cos(18 \* x / 5), 2)) / (25 \* pow(3 + sin(18 \* x / 5), 4));

}

int main()

{

double a, b, x0;

a = 0;

b = 0.85;

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

{

x0 = a;

}

else

{

x0 = b;

}

while (abs(f(x0)) >= eps) {

x0 = x0 – (f(x0) / f1(x0));

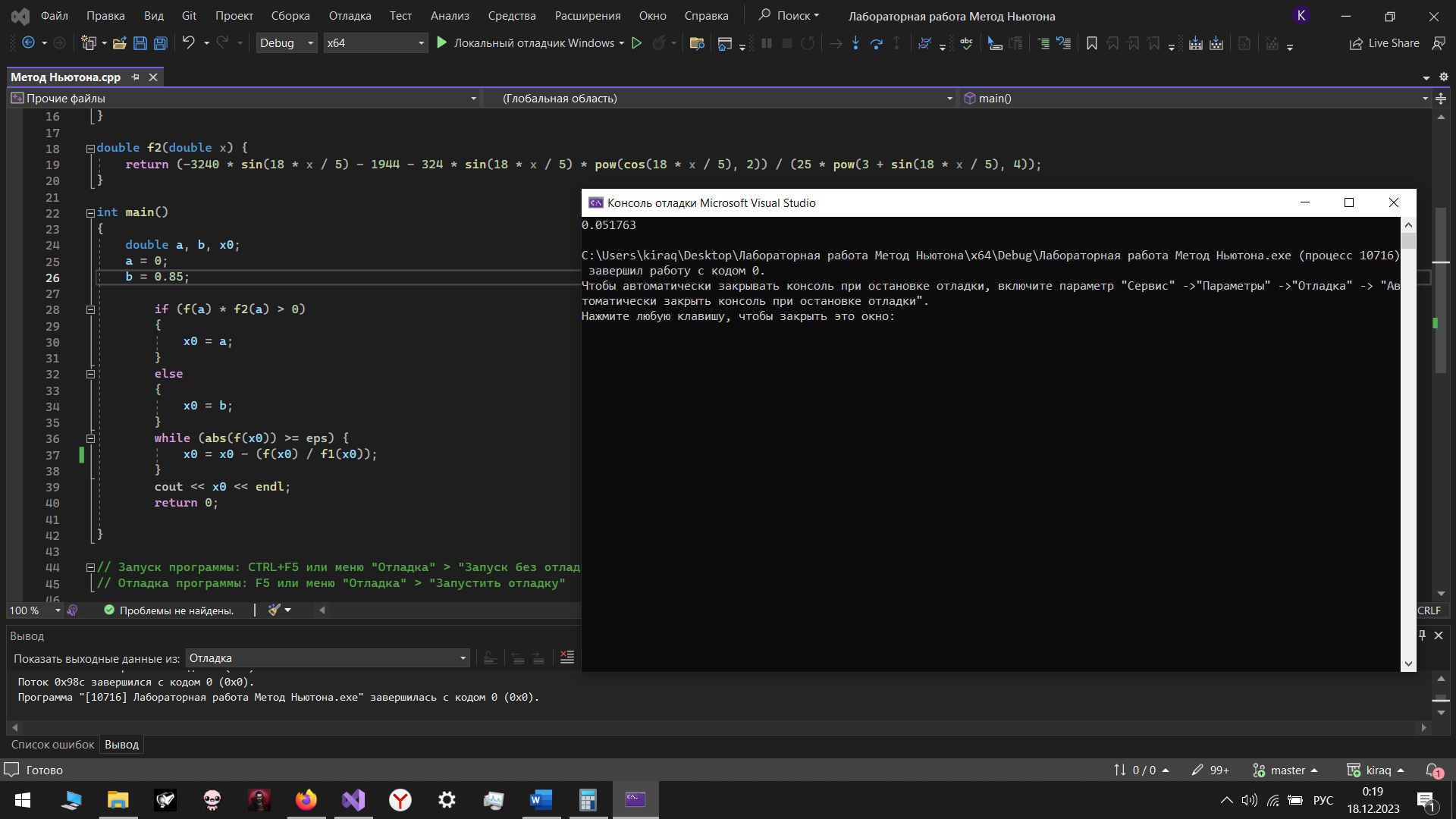
}

cout << x0 << endl;

return 0;

}

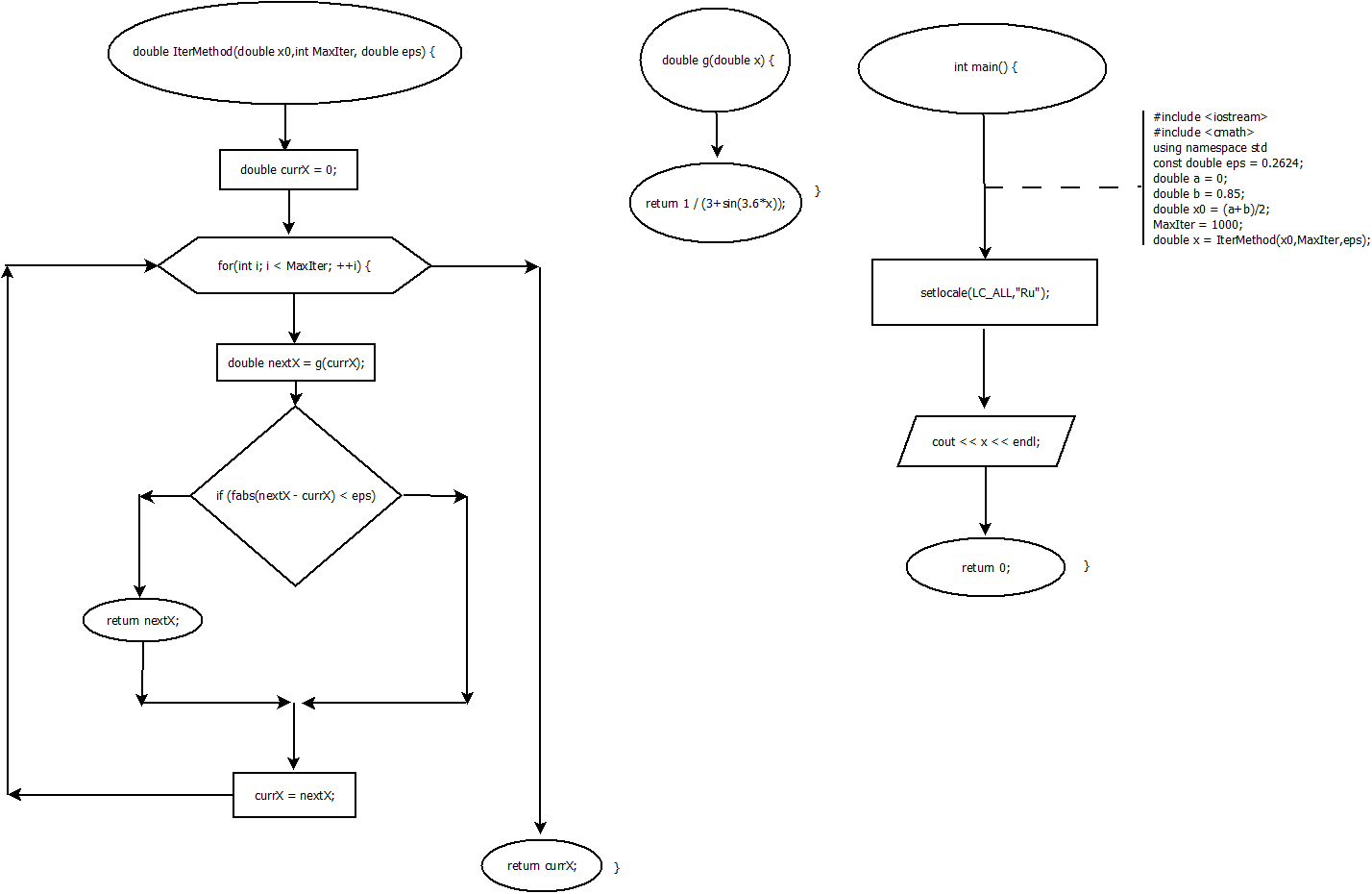
Работа программы:



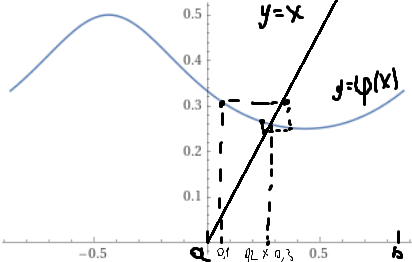
Погрешность = 0.210678

Метод Итераций:

Блок-схема:



Геометрическая интерпритация:



Код на языке С++:

#include <iostream>

#include <cmath>

using namespace std;

double g(double x) {

return 1 / (3+sin(3.6\*x));

}

double IterMethod(double x0, int MaxIter, double eps) {

double currX = x0;

for (int i = 0; i < MaxIter; ++i)

{

double nextX = g(currX);

if (fabs(nextX - currX) < eps)

{

return nextX;

}

currX = nextX;

}

return currX;

}

int main()

{

double a, b;

a = 0;

b = 0.85;

int MaxIter = 1000;

double eps = 0.2426;

double x0 = (a + b) / 2;

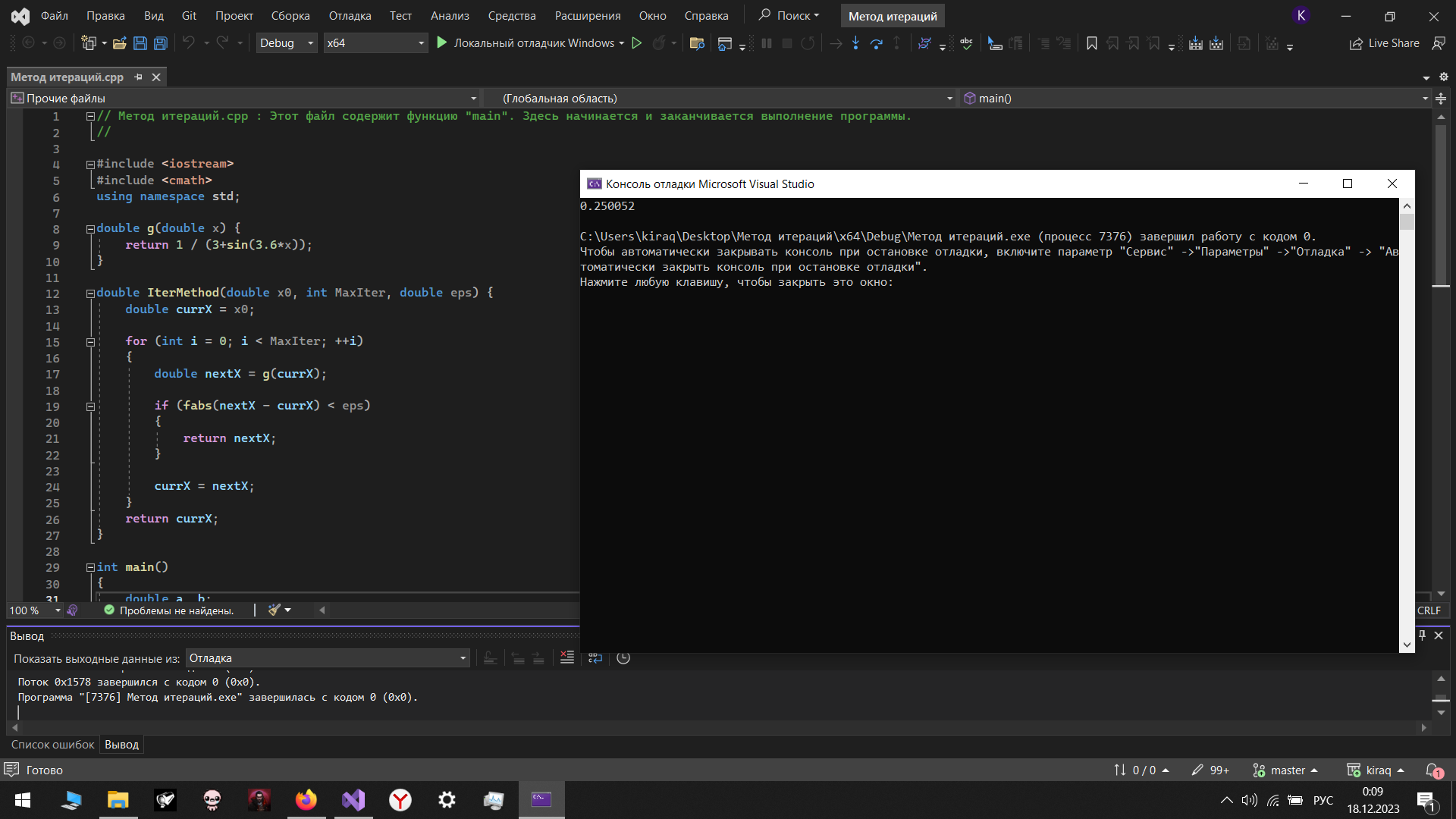
double x = IterMethod(x0, MaxIter, eps);

cout << x << endl;

return 0;

}

Работа программы:



Погрешность = 0.012389

**Вывод:** Все три метода работают как задуманно, самый точный метод – метод Итраций, самый неточный – метод Ньютона.