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: 09.03.04 – «Программная инженерия»

**Лабораторная работа №1.**

**«Решение нелинейных уравнений»**

Вариант 16

Выполнил студент гр. РИС-24-1б

Вицев Максим Олегович

Проверил:

Доц. каф. ИТАС

Ольга Андреевна Полякова

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(Оценка) (подпись)

\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

г. Пермь. 2024

Метод Ньютона

Отрезок: [0; 1]

Точность:

Геометрическая интерпретация

y

0

1

x

Решение

Обоснование стороны подхода к функции

Выбираем начальное значение корня на отрезке [0; 1]:

Вывод формулы поиска корня

Находим следующий приближенный корень

Проверка:

Находим следующий приближенный корень

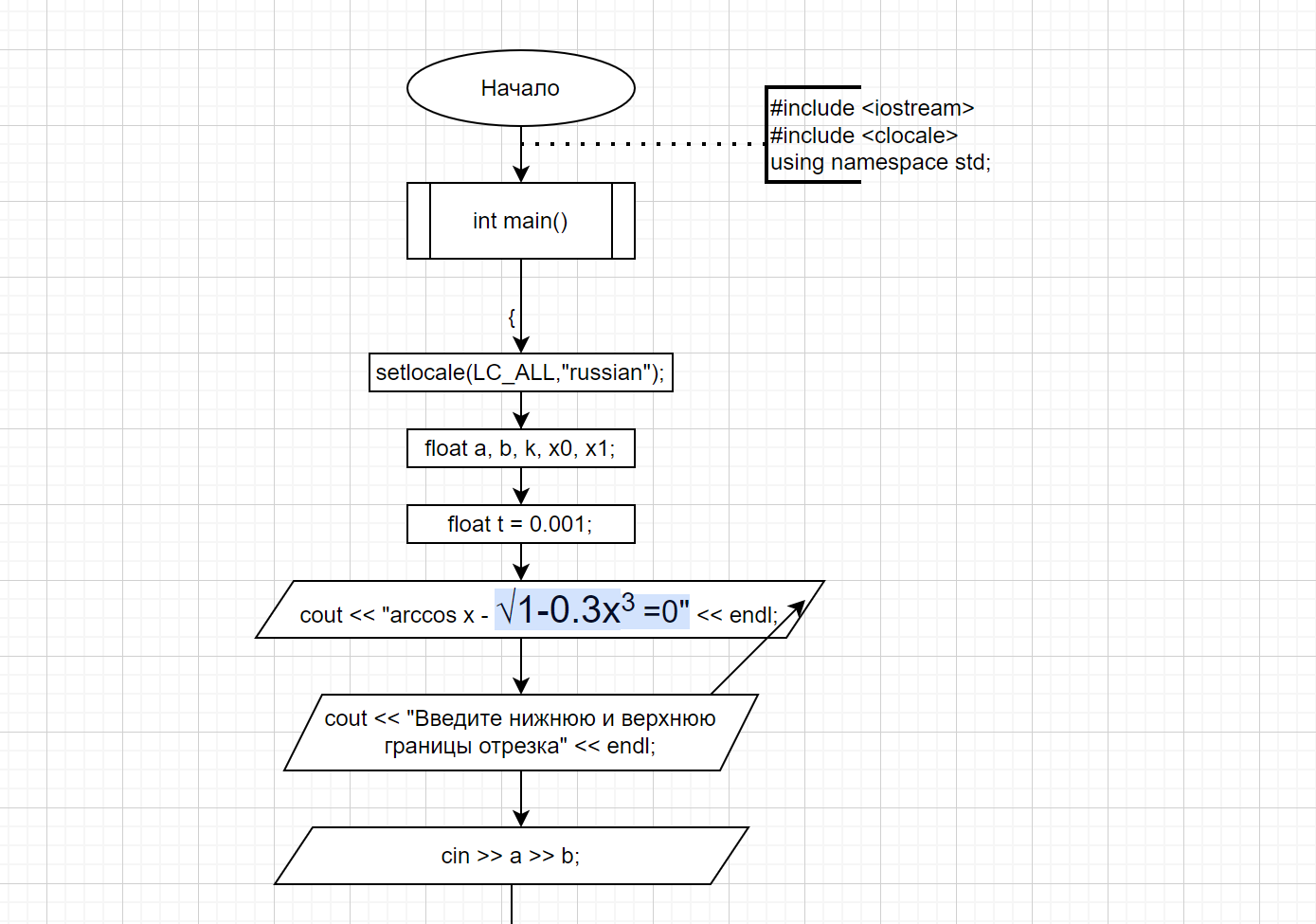
Проверка:

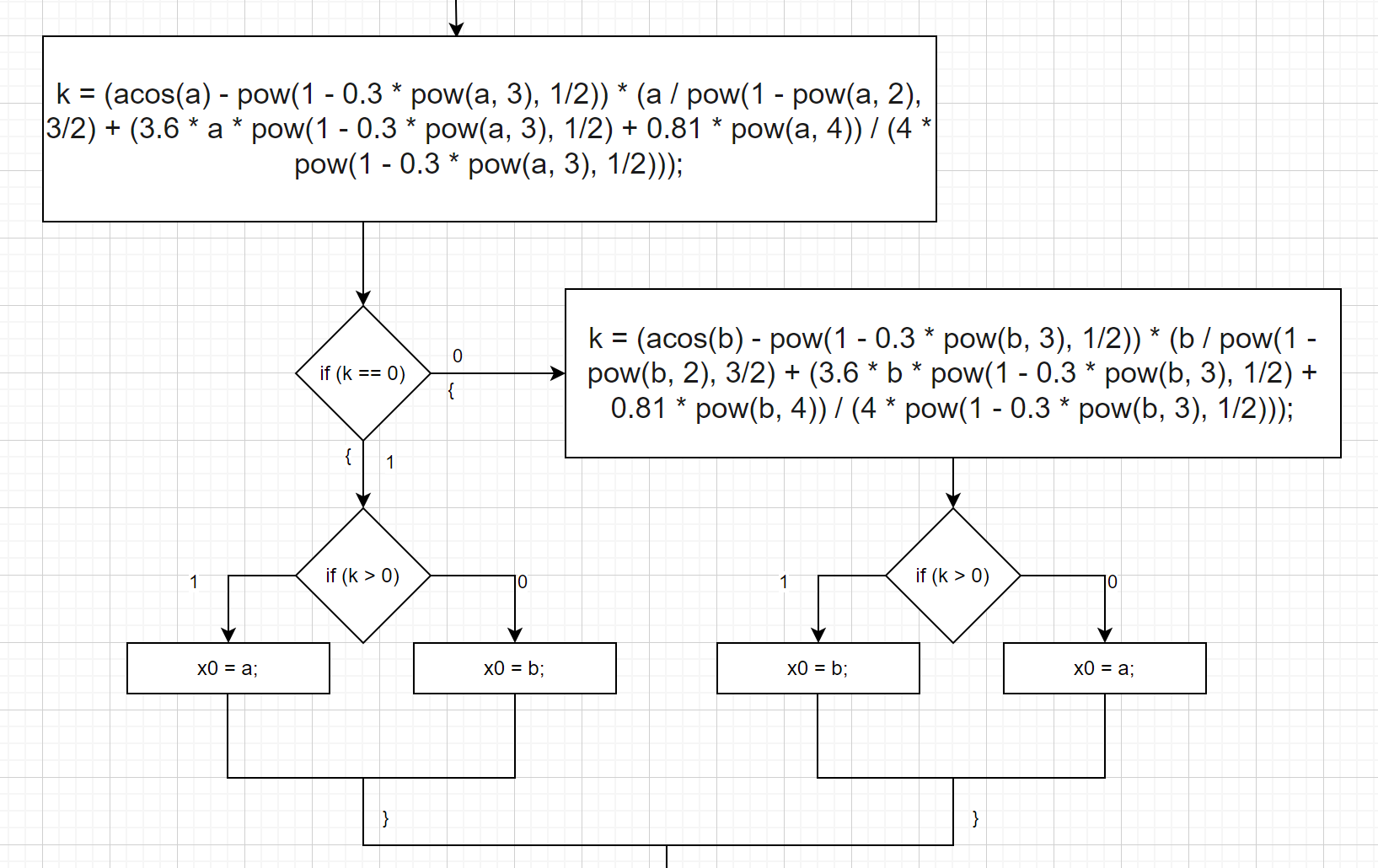
Находим следующий приближенный корень

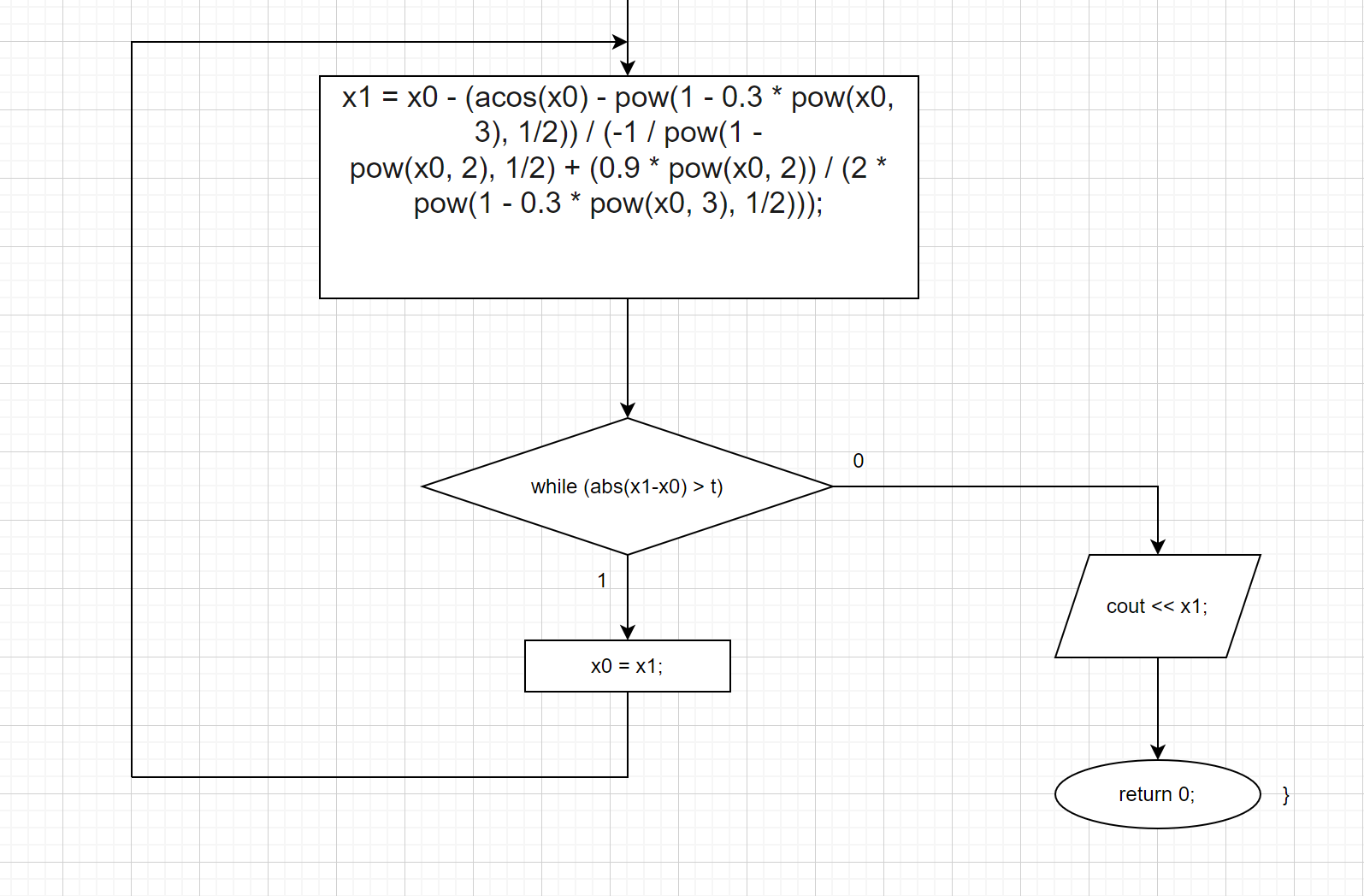
Проверка:

Ответ: Корень уравнения: 0.5629

Блок-схема







Код

#include <iostream>

#include <clocale>

#include <math.h>

using namespace std;

int main()

{

    setlocale(LC\_ALL, "Russian");

    float a, b, k, x0, x1;

    float t = 0.001;

    cout << "arccos x - √1-0.3x3 =0" << endl;

    cout << "Введите нижнюю и верхнюю границы отрезка" << endl;

    cin >> a >> b;

    k = (acos(a) - pow(1 - 0.3 \* pow(a, 3), 1/2)) \* (a / pow(1 - pow(a, 2), 3/2) + (3.6 \* a \* pow(1 - 0.3 \* pow(a, 3), 1/2) + 0.81 \* pow(a, 4)) / (4 \* pow(1 - 0.3 \* pow(a, 3), 1/2)));

    if (k == 0)

    {

        k = (acos(b) - pow(1 - 0.3 \* pow(b, 3), 1/2)) \* (b / pow(1 - pow(b, 2), 3/2) + (3.6 \* b \* pow(1 - 0.3 \* pow(b, 3), 1/2) + 0.81 \* pow(b, 4)) / (4 \* pow(1 - 0.3 \* pow(b, 3), 1/2)));

         if (k > 0)

        {

            x0 = b;

        }

        else

        {

            x0 = a;

        }

    }

    else

    {

        if (k > 0)

        {

            x0 = a;

        }

        else

        {

            x0 = b;

        }

    }

    x1 = x0;

    do

    {

        x0 = x1;

        x1 = x0 - (acos(x0) - pow(1 - 0.3 \* pow(x0, 3), 1/2)) / (-1 / pow(1 - pow(x0, 2), 1/2) + (0.9 \* pow(x0, 2)) / (2 \* pow(1 - 0.3 \* pow(x0, 3), 1/2)));

    } while (abs(x1 - x0) > t);

    cout << x1;

return 0;

}

Метод итераций

Отрезок: [0; 1]

Точность:

Геометрическая интерпретация

y

0

1

x

Решение

Преобразуем уравнение к итерационному виду

Обоснование стороны подхода к функции

Проводим проверку на сходимость (первое приближение к корню)

Находим следующий приближенный корень

Проверка

Находим следующий приближенный корень

Проверка

Находим следующий приближенный корень

Проверка

Находим следующий приближенный корень

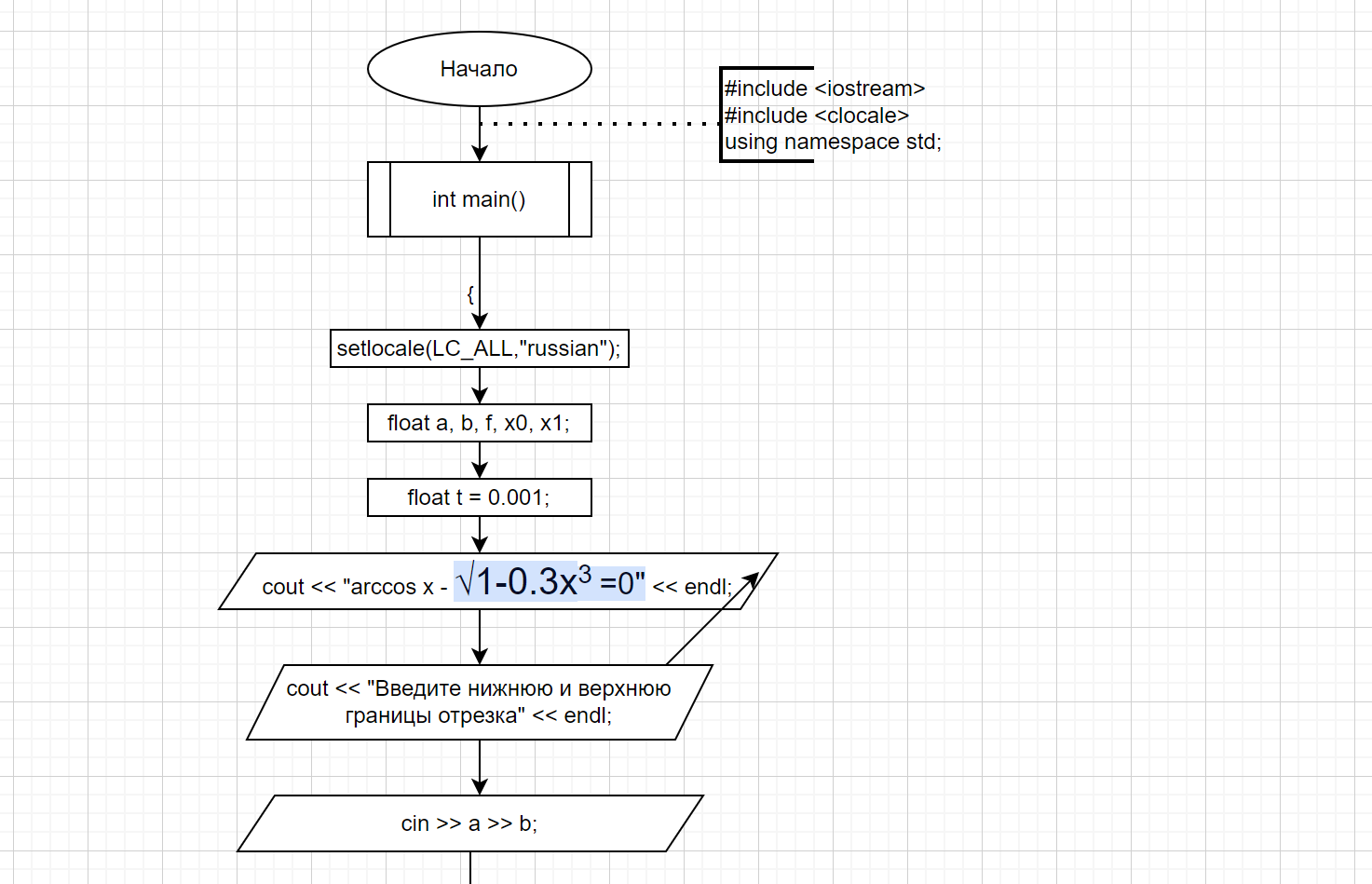
Проверка

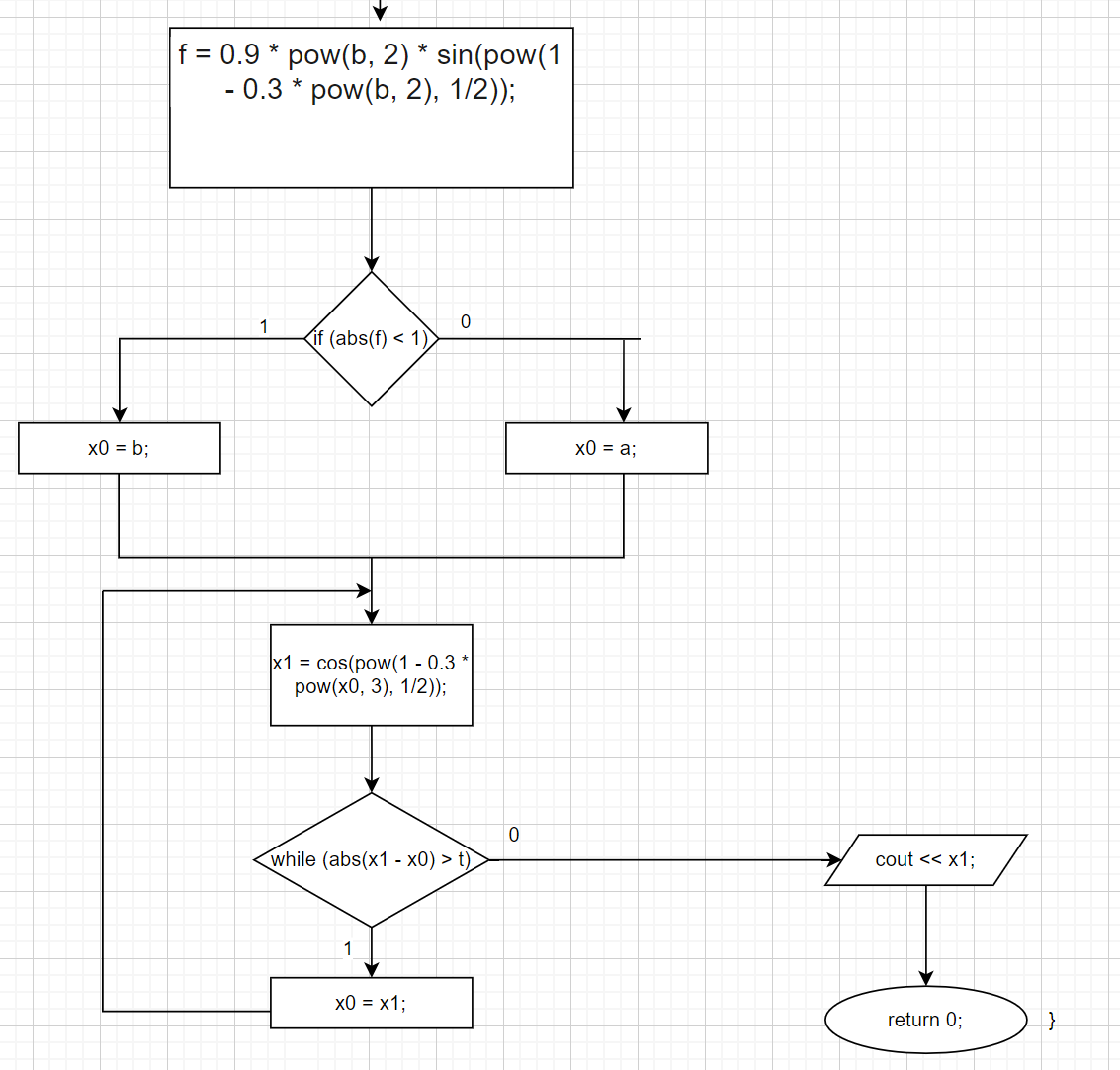
Находим следующий приближенный корень

Проверка

Ответ: Корень уравнения: 0.5629

Блок-схема





Код

#include <iostream>

#include <clocale>

#include <math.h>

using namespace std;

int main()

{

    setlocale(LC\_ALL, "Russian");

    float a, b, f, x0, x1;

    float t = 0.001;

    cout << "arccos x - √1-0.3x3 =0" << endl;

    cout << "Введите нижнюю и верхнюю границы отрезка" << endl;

    cin >> a >> b;

    f = 0.9 \* pow(b, 2) \* sin(pow(1 - 0.3 \* pow(b, 2), 1/2));

    if (abs(f) < 1)

    {

        x0 = b;

    }

    else

    {

        x0 = a;

    }

    x1 = x0;

    do

    {

        x0 = x1;

        x1 = cos(pow(1 - 0.3 \* pow(x0, 3), 1/2));

    } while (abs(x1 - x0) > t);

    cout << x1;

    return 0;

}

Метод половинного деления

Отрезок, на котором существует корень уравнения: [0; 1]

Точность:

Геометрическая интерпретация

y

0

1

x

0.5

0.75

0.625

Решение

Найдем середину отрезка

Точка считается первым приближенным значением корня

Обоснование стороны подхода к функции

Корень находится на отрезке [a; ], если

Корень находится на отрезке [; b], если

Отрезок, не содержащий корня, отбрасывается.

Отрезок [0; 0.5] отбрасывается

Граница a сдвигается в точку .

Остается отрезок [0.5; 1]

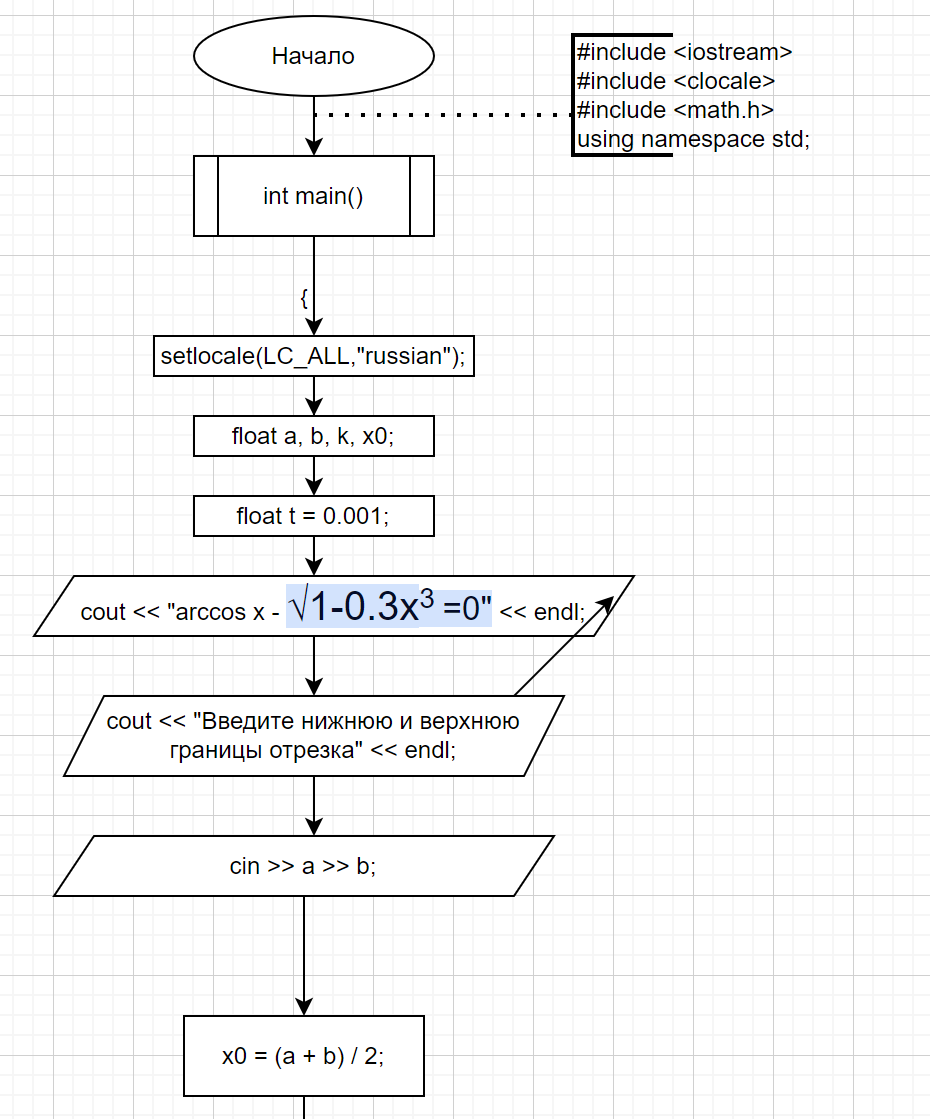
Действие повторяется: находим середину отрезка

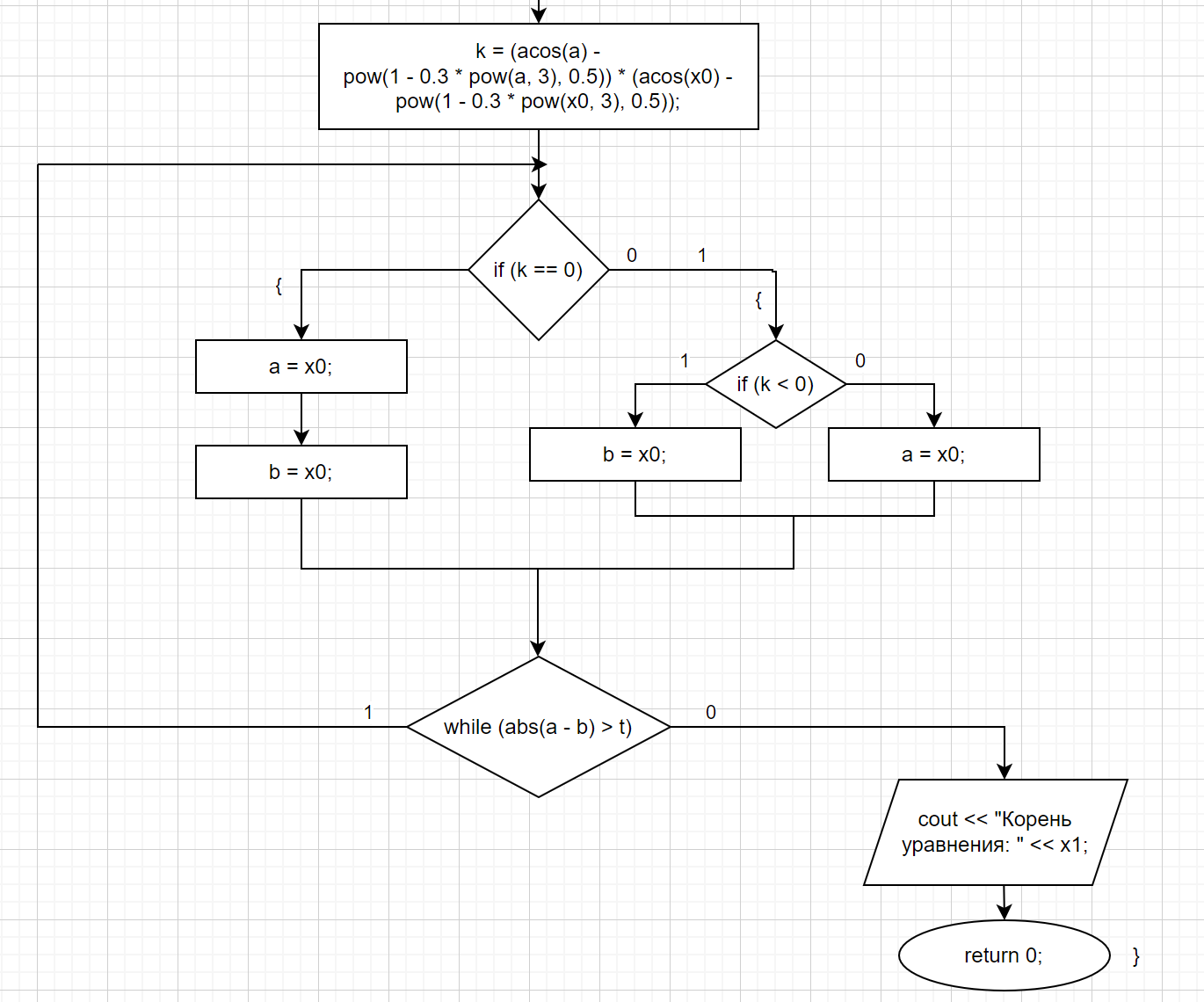
Граница b сдвигается в точку . Отрезок [0.75; 1] отбрасывается. Остается отрезок [0.5; 0.75].

Действия повторяется пока не выполнится условие:

Решение с помощью цикла

Блок-схема





Код

#include <iostream>

#include <clocale>

#include <math.h>

using namespace std;

int main()

{

    setlocale(LC\_ALL, "Russian");

    float a, b, k, x0;

    float t = 0.001;

    cout << "arccos x - √1-0.3x3 =0" << endl;

    cout << "Введите нижнюю и верхнюю границы отрезка" << endl;

    cin >> a >> b;

    do

    {

        x0 = (a + b) / 2;

        k = (acos(a) - pow(1 - 0.3 \* pow(a, 3), 0.5)) \* (acos(x0) - pow(1 - 0.3 \* pow(x0, 3), 0.5));

        if (k == 0)

        {

            a = x0;

            b = x0;

        }

        else

        {

            if (k < 0)

                b = x0;

            else a = x0;

        }

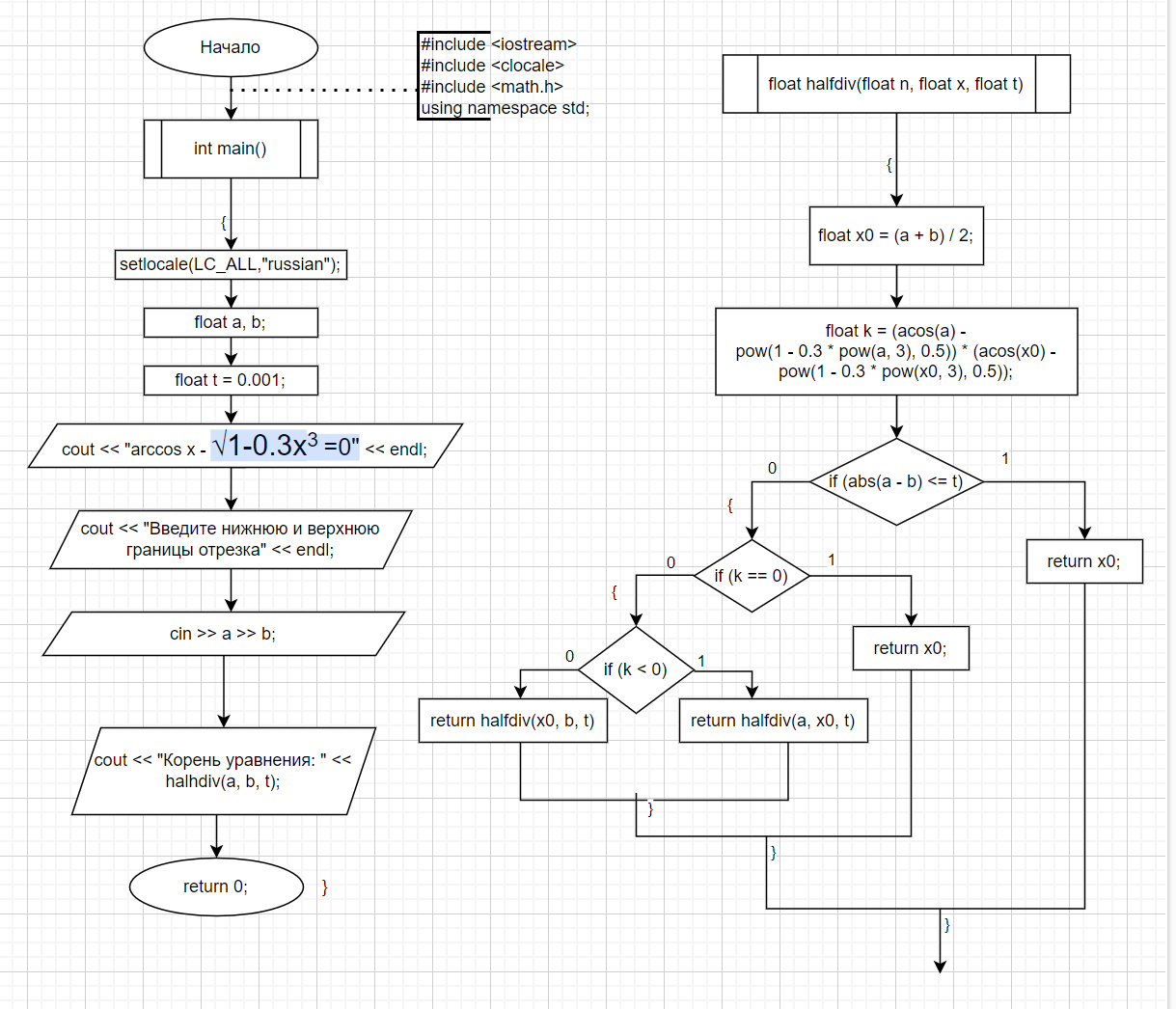
    } while (abs(a - b) > t);

    cout << "Корень уравнения: " << x0;

}

Решение с помощью рекурсии

Блок-схема



Код

#include <iostream>

#include <clocale>

#include <math.h>

using namespace std;

float halfdiv(float a, float b, float t)

{

    float x0 = (a + b) / 2;

    float k = (acos(a) - pow(1 - 0.3 \* pow(a, 3), 0.5)) \* (acos(x0) - pow(1 - 0.3 \* pow(x0, 3), 0.5));

    if (abs(a - b) <= t)

        return x0;

    else

    {

        if (k == 0)

            return x0;

        else

        {

        if (k < 0)

            return halfdiv(a, x0, t);

        else return halfdiv(x0, b, t);

        }

    }

}

int main()

{

    setlocale(LC\_ALL, "Russian");

    float a, b;

    float t = 0.001;

    cout << "arccos x - √1-0.3x3 =0" << endl;

    cout << "Введите нижнюю и верхнюю границы отрезка" << endl;

    cin >> a >> b;

    cout << "Корень уравнения: " << halfdiv(a, b, t);

}

Ссылка и скриншот на GitHub