Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа**

**“Решение нелинейных уравнений”**

Выполнил:

студент группы РИС-23-2б

Швацкий Даниил Александрович.

Проверила:

доцент кафедры ИТАС

О.А. Полякова

2023 г.

Постановка задачи:

Дано уравнение , заданный отрезок содержащий корень [0 ; 1], зданная тоность = e = 10^-6. Найти корень уравнения используя методы: половинного деления, Ньютона, итераций.

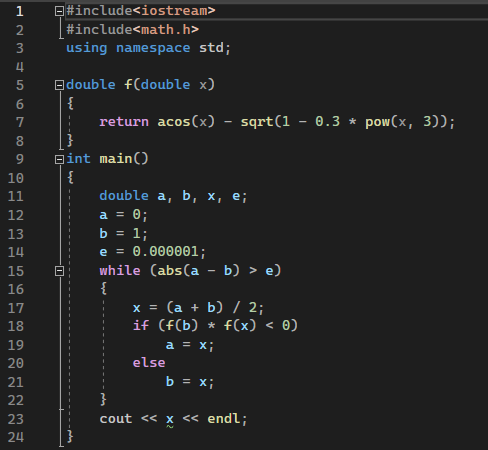
Метод половинного деления

Разработка алгоритма:

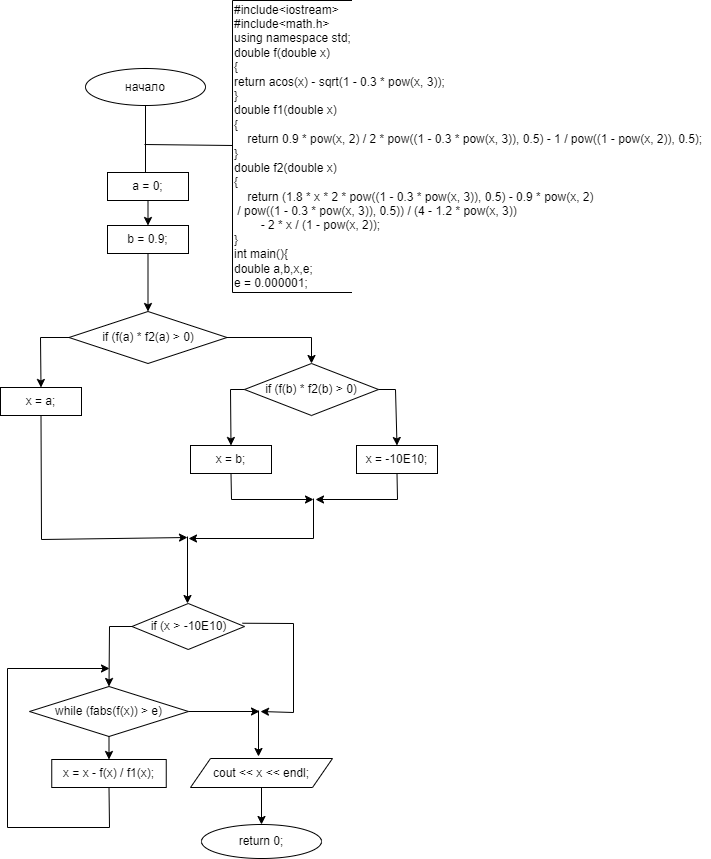
Создадим функцию double f(double x), которая будет возвращать значение функции, заданной пользователем.

В функции main() объявим переменные a,b,x,e типа double. Переменные a, b – это начала и конца отрезка, в котором есть хотя бы один корень, соответственно. Переменная x = (a + b) / 2 (половина заданного отрезка). Если f(b)\*f(x) < 0, то рассматриваем новый отрезок [x; b], иначе новый отрезок, в котором находится корень [a; x] (проверка откуда брать следующую половину отрезка). На каждом шаге проверяем приближение к корню, то есть повторяем эти действия до тех пор, пока abs(a – b) > e, где e – заданная точноть, для этого используем цикл while (abs(a-b) > e). В конце работы программы пользователю выводится x – корень уравнения.

Код на C++ :



Блок-схема:



Метод Ньютона

Разработка алгоритма:

Так как для данного метода нужно посчитать первую и вторую производную от заданной функции, то посчитаем их заранее и запишем в отдельные функции. Тогда double f(double x) – возвращает значение функции,

double f1(double x) – возвращает значение производной от этой функции, double f2(double x) – возвращает значение второй производной от этой функции.

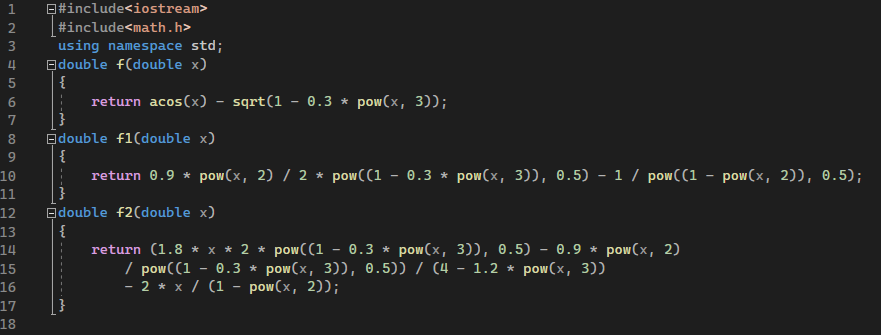
Объявим переменные a,b,x,e типа double. Где a , b – заданный отрезок, на котором находится корень, e – заданная точность (= 0.000001)

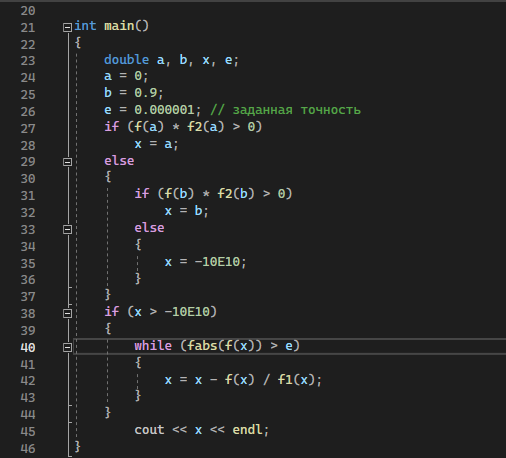
Далее проверяем условие сходимости if( f(a)\*f’’(a) ), если условие выолняется то x = a, иначе проверяем условие if( f(b)\*f’’(b) ), если это условие выполянется, то x = b, иначе x = -10^11.

Если x > -10^11, значит есть корень, находим его по формуле:

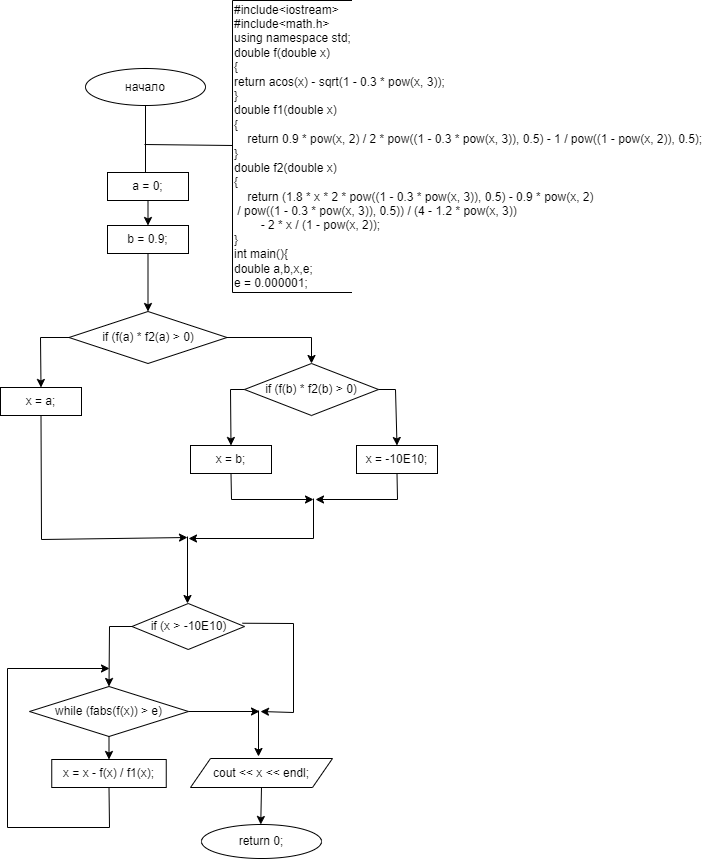
x = x – f(x) / f1(x). И делаем это до тех пор, пока корень не станет соответствовать заданной точности e. После чего выводим x на экран.

Код на C++ :





Блок-схема:



Метод итераций

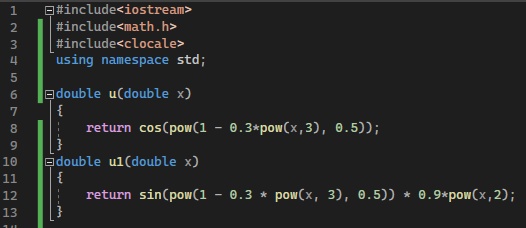
Разработка алгоритма:

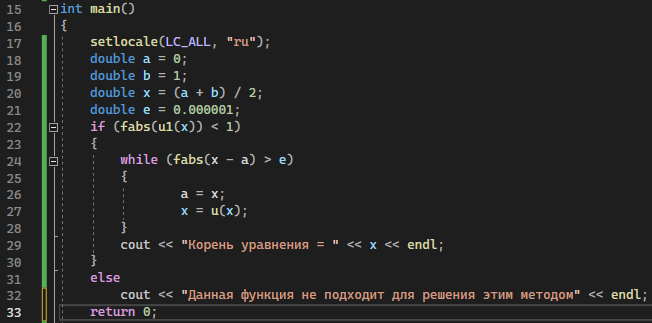
Выразим изначальную функцию через переменную x, x = φ(x). Найдем производную φ’(x), чтобы проверить удловлетворяет ли фукнция условию |φ’(x)| < 1, если условие выполняется, то можно переходить к поиску корня. Инициализируем переменные a,b,x,e типа double. Переменные a, b – это начало и конец отрезка, который содержит корень уравнения, соответственно. Переменная x – это точка с которой начинается поиск корня, она должна входить в заданный отрезок, например, x = (a+b) / 2.

Напишем функции double u(double x), double u1(double x). Функция u(x) – это функция φ(x), а u1(x) – производная φ(x). Функция u1(x) нужна для проверки подходит ли уравнение для решения этим методом, а u(x) для поиска корня.

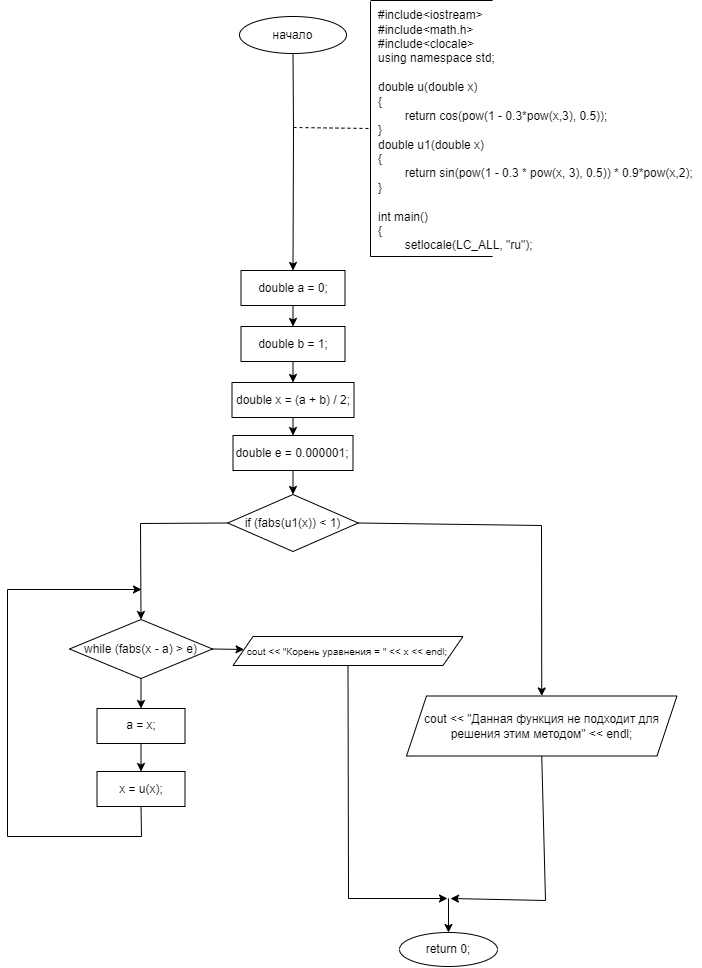
Если fabs(u1(x))<1, то начинаем поиск корня, x = u(x). Повторяем это действие пока не достигнем заданной точности. While(fabs(x – a) > e). После чего выводим корень на экран.

Код на C++ :





Блок-схема:



Пример выполнения алгоритма:



