

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ЭЛЕКТРОТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА «ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
АВТОМАТИЗИРОВАННЫХ СИСТЕМ»

ОТЧЁТ ПО ПРАКТИЧЕСКИМ ЗАДАЧАМ

Дисциплина: «Основы алгоритмизации и программирования»

Выполнил:
Студент группы ИВТ-21-26
Безух Владимир Сергеевич

Проверил:
Доцент кафедры ИТАС
Полякова Ольга Андреевна

Пермь,
Декабрь 2021

Постановка задачи

Вычислить приближенное значение корня функции $3 \sin(\sqrt{x}) + 0,35x - 3,8 = 0$ при помощи дихотомического метода и метода касательных.

Анализ задачи

1. Дихотомический поиск:

- а) Найти середину x_0 отрезка $[a; b]$.
- б) Отбросить половину отрезка, где нет корня. Останется отрезок $[a; x_0]$ или $[x_0; b]$.
- в) Найти середину полученного в шаге «б» отрезка x_1 .
- г) Сравнить расстояние между точками x_0 и x_1 с заданной точностью ϵ .
- д) Повторять до тех пор, пока не будет найден приближенный корень.

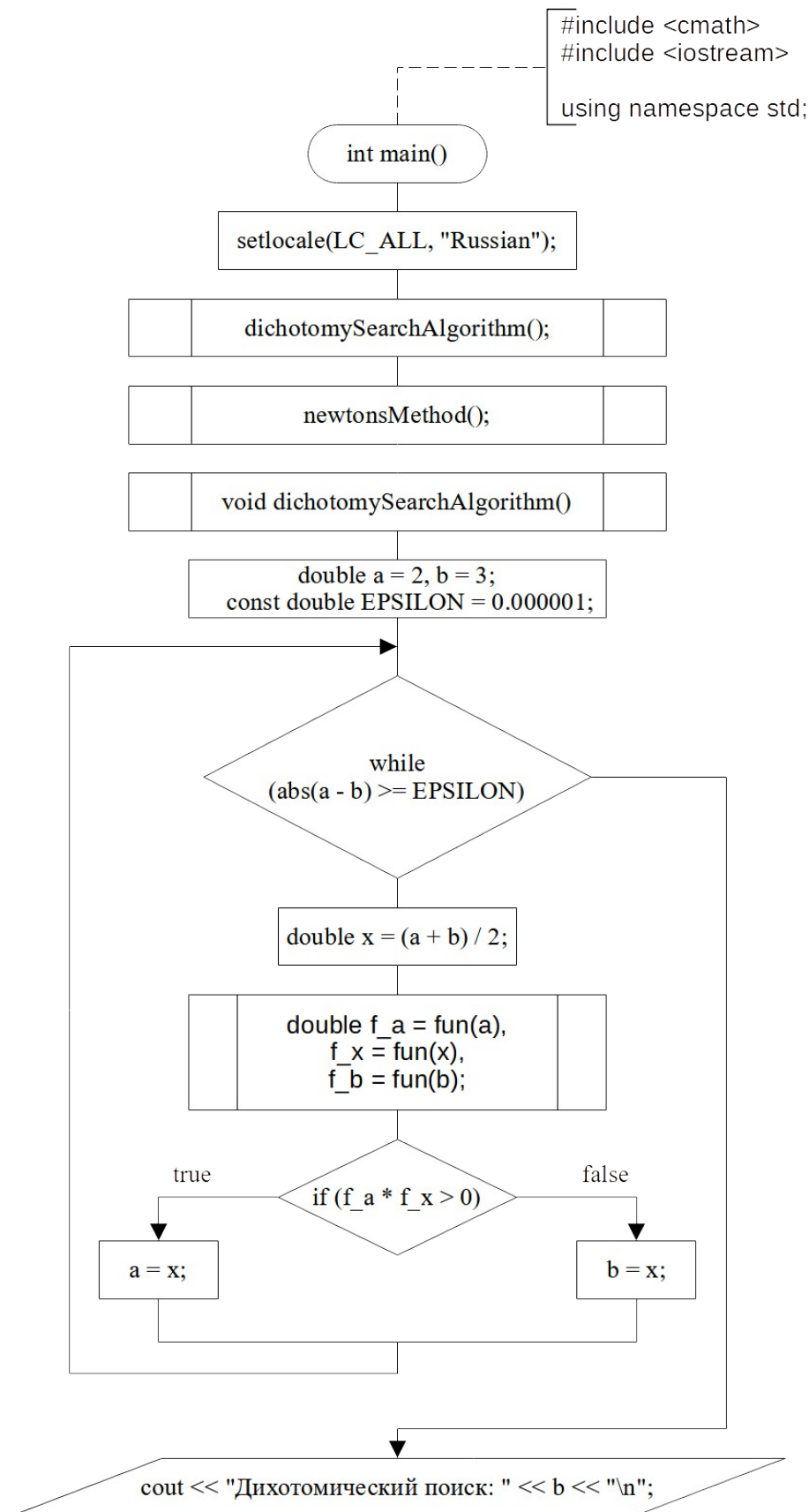
2. Метод Ньютона:

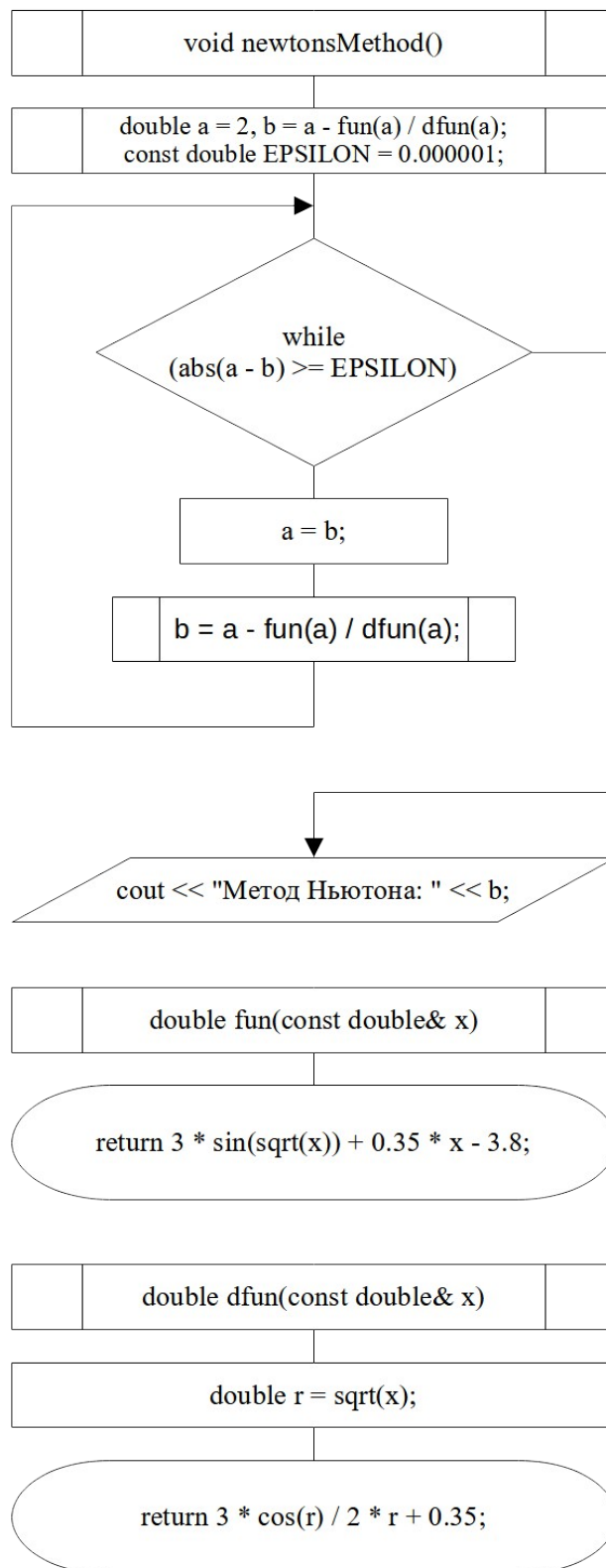
- а) Ищем приближенное по формуле $a = b - f(b) / f'(b)$.
- б) Сравнить расстояние между полученными приближенными с заданной точностью.
- в) Повторять до тех пор, пока не будет найден приближенный корень.

Описание переменных

double a = 2, b = 3; — границы гарантирующего отрезка
const double EPSILON = 0.000001; — точность

Блок-схема





Исходный код

```
#include <cmath>
#include <iostream>

using namespace std;

double fun(const double& x) { return 3 * sin(sqrt(x)) + 0.35 * x - 3.8; } // 2.29854
double dfun(const double& x) { double r = sqrt(x); return 3 * cos(r) / 2 * r + 0.35; }

void dichotomySearchAlgorithm()
{
    double a = 2, b = 3;
    const double EPSILON = 0.000001;

    while (abs(a - b) >= EPSILON)
    {
        double x = (a + b) / 2;
        double f_a = fun(a), f_x = fun(x), f_b = fun(b);

        if (f_a * f_x > 0)
            a = x;
        else
            b = x;
    }

    cout << "Дихотомический поиск: " << b << "\n";
}

void newtonsMethod()
{
    double a = 2, b = a - fun(a) / dfun(a);
    const double EPSILON = 0.000001;

    while (abs(a - b) >= EPSILON)
    {
        a = b;
        b = a - fun(a) / dfun(a);
    }

    cout << "Метод Ньютона: " << b;
}

int main()
{
    setlocale(LC_ALL, "Russian");

    dichotomySearchAlgorithm();
    newtonsMethod();
}
```

Скриншот консольного интерфейса программы

```
Дихотомический поиск: 2.29854  
Метод Ньютона: 2.29854
```


Анализ результатов

Двумя способами найден искомый с заданной точностью корень нелинейного уравнения.