КОМИТЕТ ПО ОБРАЗОВАНИЮ ПРАВИТЕЛЬСТВА САНКТ-ПЕТЕРБУРГА

Санкт-Петербургское государственное

бюджетное профессиональное образовательное учреждение

«Колледж информационных технологий»

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2**

**по численным методам**

**Специальность 09.02.07**

**«Информационные системы и программирование»**

**Специализация**

**«Разработчик web и мультимедийных приложений»**

**(по программе базовой подготовки)**

Выполнил студент 91 гр.: А.В. Виноградов\_\_\_\_\_\_\_\_\_

Тел.

e-mail

Принял преподаватель: И.П. Смирнова \_\_\_\_\_\_\_\_\_\_

Санкт-Петербург 2020

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc20467494)

**ПРАКТИЧЕСКАЯ РАБОТА 1** 4

**ПРАКТИЧЕСКАЯ РАБОТА 2** 8

**ВВЕДЕНИЕ**

Задача – рассмотреть метод Ньютона (касательных), используя уравнение из предыдущего задания и сравнить результаты. Для этого:

1. Найти начальное приближение корня х0 графическим методом (MS Excel).

2. Написать программу на языке программирования (.exe файл **или** текст + скриншот результата).

Для проверки графического метода использовалась программа MS Excel. Для программной проверки использовался язык C, написание кода происходило в Visual Studio Code, проверка работы программы происходила в командной строке Visual Studio Code.

**ПРАКТИЧЕСКАЯ РАБОТА 1**

Изначальная функция: f(x) = x3 – 2x + 2

Первая производная функции: f’(x) = 3x2 – 2

Точность: ε = 0,001

Сначала пробуем метод Ньютона в MS Excel (рисунок 1). Столбец n – число итераций, столбец x – значение (при этом значение «х» при n = 0 мы выставляем сами, и данное значение называется **нулевым приближением)**, функция f(x) – значение функции при заданном x, производная f’(x) – значение производной функции при заданном x, приращение dx – частное f(x) и f’(x). Последующие значения «х» мы получаем при вычитании приращения из прошлого значения «х». Результат:



Рисунок 1 – Результат в Excel

Произошло зацикливание при расчетах данной функции. Этот метод не всегда подходит для уточнения корней.

Проверим, как себя поведет программа на C. Скриншот программы (рисунок 2):

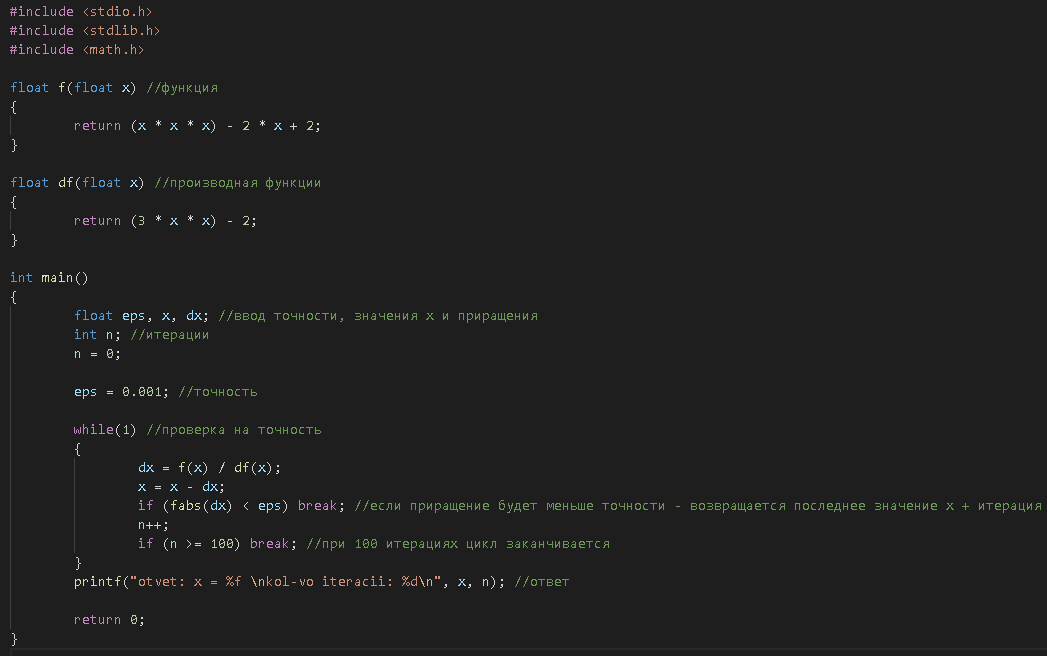


Рисунок 2 – Текст программы

Текст программы:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

float f(float x) //функция

{

        return (x \* x \* x) - 2 \* x + 2;

}

float df(float x) //производная функции

{

        return (3 \* x \* x) - 2;

}

int main()

{

        float eps, x, dx; //ввод точности, значения х и приращения

        int n; //итерации

        n = 0;

        eps = 0.001; //точность

        while(1) //проверка на точность

        {

                dx = f(x) / df(x);

                x = x - dx;

                if (fabs(dx) < eps) break; //если приращение будет меньше точности - возвращается последнее значение х + итерация

                n++;

                if (n >= 100) break; //при 100 итерациях цикл заканчивается

        }

        printf("otvet: x = %f \nkol-vo iteracii: %d\n", x, n); //ответ

        return 0;

}

Результат работы программы в командной строке Visual Studio Code (рисунок 3):

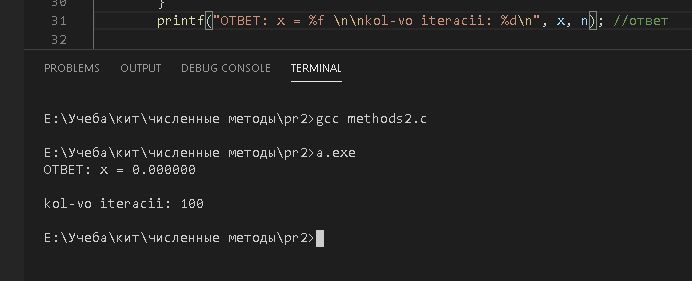


Рисунок 3 – Результат работы

Вывод: программное и графическое решения показали, что данный метод не подходит для нашей функции. Воспользуемся другой функцией, точность оставим прежней.

**ПРАКТИЧЕСКАЯ РАБОТА 2**

Функция: f(x) = x3 + 2x2 – 8x + 3

Первая производная функции: f’(x) = 3x2 + 4x – 8

Точность: ε = 0,001

Проверяем данную функцию в MS Excel. Итерация с необходимой точностью выделена зеленой (рисунок 4).

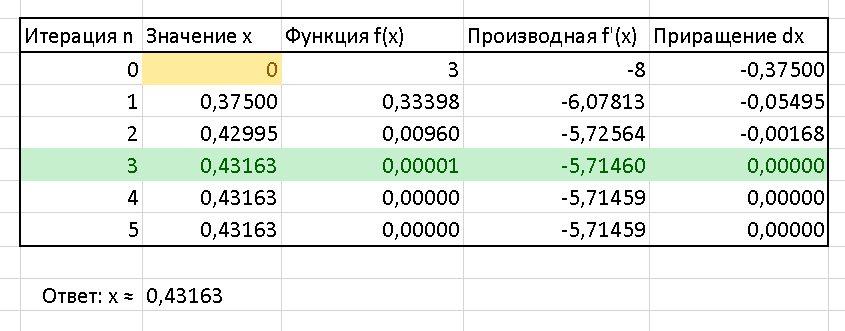


Рисунок 4 – Результат в Excel

Сравним приблизительный ответ, полученный в MS Excel с ответом, полученный в программе. Изменяем функцию в коде на новую (рисунок 5):

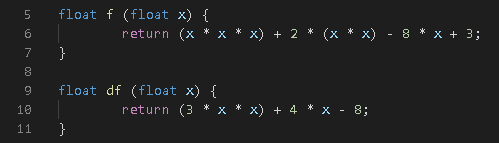


Рисунок 5 – Измененная функция в коде программы

Результат работы программы (рисунок 6):

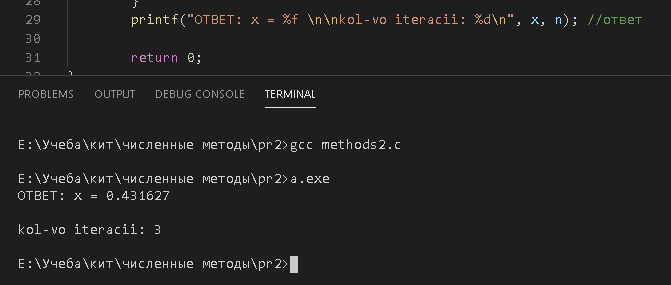


Рисунок 5 – Результат работы

Вывод: В MS Excel x ≈ 0.**4316**3; в программе x = 0.**4316**27. Точность соблюдена.