**­­­­Московский Авиационный Институт**

(Национальный Исследовательский Университет)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

Курсовой проект

по курсу «Информатика»

1 семестр

Задание 4.

Тема: «Процедуры и функции в качестве параметров»

|  |  |
| --- | --- |
| Студент: | Доянова В. А. |
| Группа: | М8О-111Б-22 |
| Преподаватель: | Аносова Н. П. |
| Подпись: |  |
| Оценка: |  |

Москва, 2022

**Содержание**

**Задание3**

Формулировка задания3

Вариант3

**Теоретическая часть4**

Метод итерации4

Метод Ньютона6

Метод дихотомии8

**Практическая часть9**

Описание переменных9

Описание алгоритма10

Код программы11

Выходные данные12

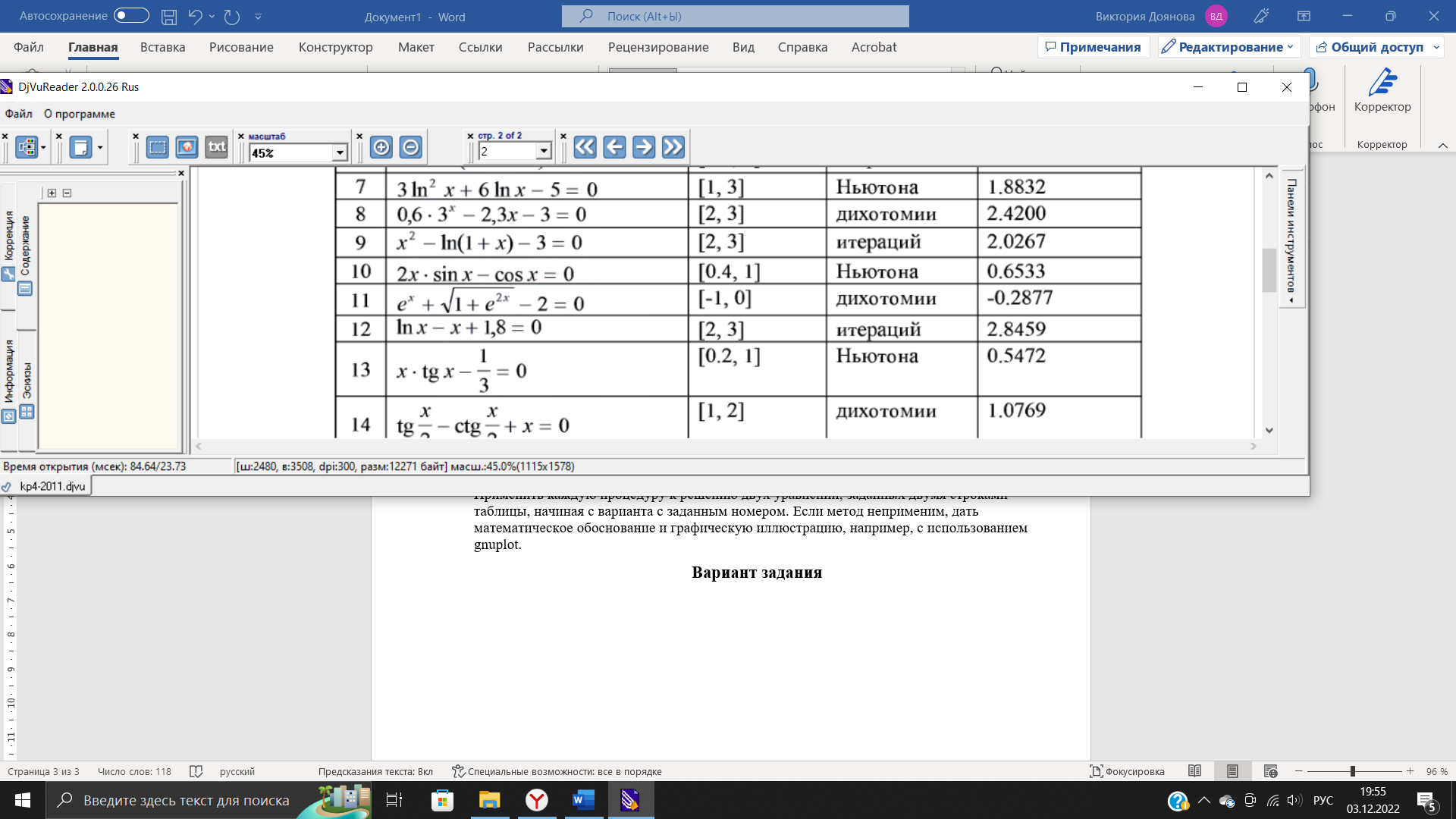
**Заключение13**

**Задание**

**Формулировка задания**

Составить программу на языке Си с процедурами решения трансцендентных алгебраических уравнений различными численными методами (итераций, Ньютона и половинного деления - дихотомии). Нелинейные уравнения оформить как параметры-функции, разрешив относительно неизвестной величины в случае необходимости. Применить каждую процедуру к решению двух уравнений, заданных двумя строками таблицы, начиная с варианта с заданным номером. Если метод неприменим, дать математическое обоснование и графическую иллюстрацию, например, с использованием gnuplot.

**Вариант задания**



**Теоретическая часть**

**Метод итераций**

Метод итераций – численный метод решения уравнений. Метод основан на принципе сжимающего отображения, который применительно к численным методам в общем виде также может называться методом простой итерации или методом последовательных приближений.

Идея метода состоит в приведение уравнения f(x) = 0 к эквивалентному уравнению x = φ(x) так, чтобы отображение φ(x) было сжимающим. Итерации начинаются со значения xM –середины отрезка. Преобразование можно делать разными способами. В частности, сохраняет корни уравнение вида φ(x)= x – d(x)f(x), если d(x) ≠ 0 на исследуемом отрезке. В данном случае возьмём d(x) = 1/f ’(xM). Это приводит к методу Ньютона и имеет условие сходимости d(x)\*f ’(x) >0. Тогда итерационный процесс принимает следующий вид: xk+1 = xk - d0 f(xk). Условием окончания итераций является достижение нужной точности между предыдущим и следующим значением.

1. Рассмотрим, можно ли использовать метод итерации на уравнении №10 () на отрезке [0.4, 1]. Найдем производную и середину отрезка:

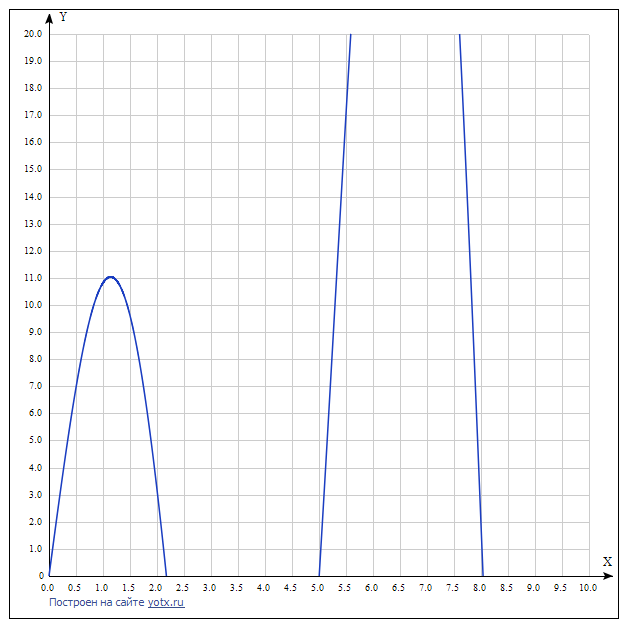
f ’(x) = 3sin(x) + 2xcos(x)

xM = (0,4 + 1)/2 = 0,7

d = f ’(xM) = 1,93 + 1,07 = 3

Проверим условия сходимости для метода итераций, построив графики производных, сжимающих отображения:

Y = d\* f ’(x) – построим график этой функции, проверяя на условие d\* f ’(x) > 0:



Условие выполнено, значит, метод итераций может быть применен к этой функции.

Приведем функцию к виду x = f(x):

1. Рассмотрим, можно ли использовать метод итерации на уравнении №11 () на отрезке [-1, 0]. Найдем производную и середину отрезка:

f ’(x) =

xM = (-1+0)/2 = -0,5

d = f ’(xM) = 0,11 : 0,8 = 0,14

Проверим условия сходимости для метода итераций, построив графики производных, сжимающих отображения:

Y = d\* f ’(x) =– построим график этой функции, проверяя на условие d\* f ’(x) > 0: Изображение выглядит как текст, компьютер, снимок экрана, дисплей

Автоматически созданное описание

Условие не выполнено, значит, метод итераций не может быть применен к этой функции.

**Метод Ньютона**

Метод Ньютона – это итерационный численный метод нахождения корня заданной функции. Поиск решения осуществляется путем построения последовательных приближений и основан на принципах простой итерации.

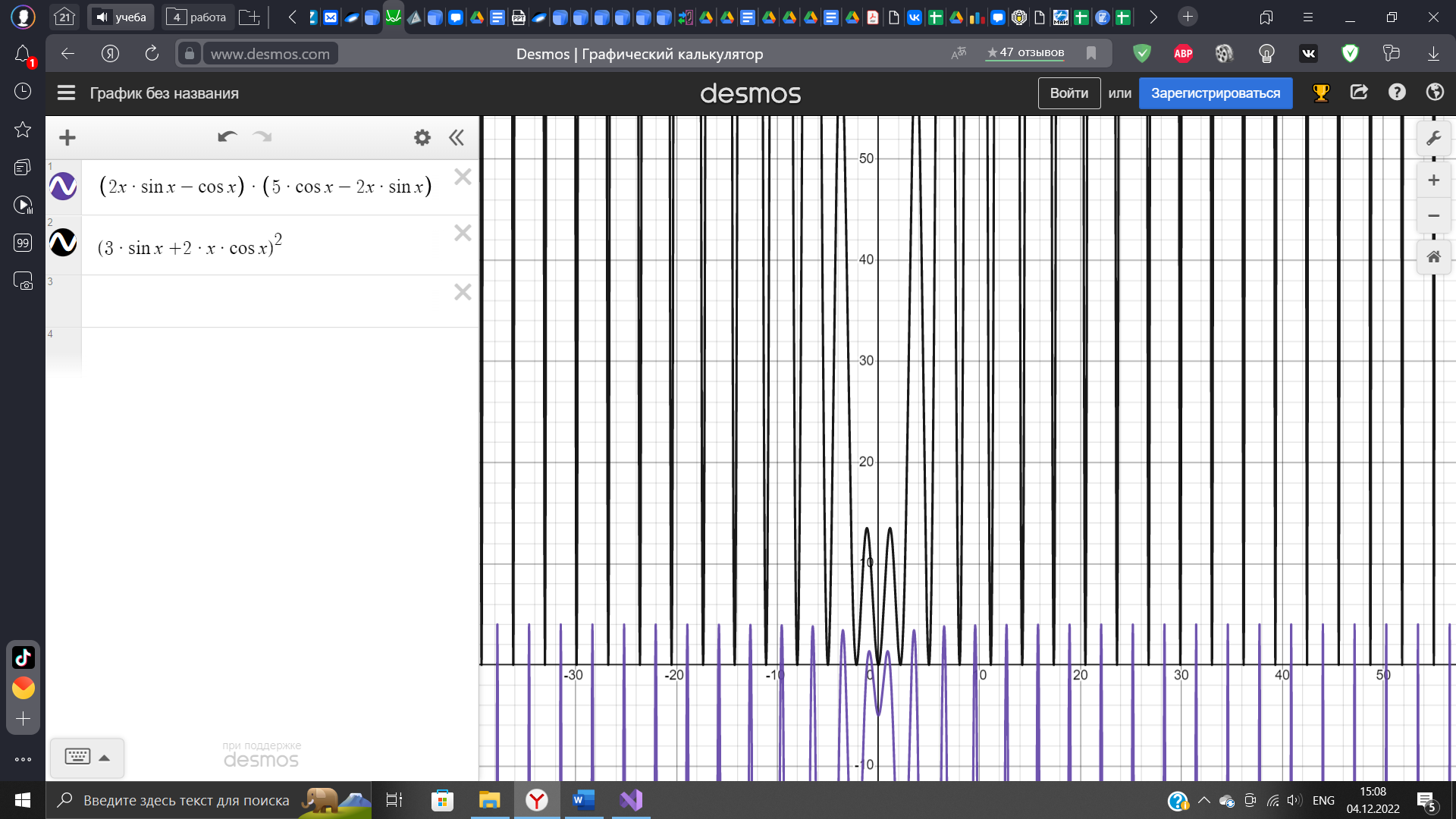
В этом методе за d0 берется значение производной в каждой новой точке. Тогда итерационный процесс имеет вид . Условия окончания итераций и начальное значение такие же, как в методе итерации. Условие сходимости метода выглядит следующим образом: |f(x) \* f ’’(x)| < .

1. Рассмотрим, можно ли использовать метод Ньютона на уравнении №10 () на отрезке [0.4, 1].

Найдем функции y1 = |f(x) \* f ’’(x)| и y2 = :

y1 =

y2 =

Проверим условия сходимости для метода Ньютона, построив графики функций y1 и y2, проверяя условие |f(x) \* f ’’(x)| < :

Условие выполняется, значит, метод Ньютона может быть применен к этой функции.

1. Рассмотрим, можно ли использовать метод Ньютона на уравнении №11 () на отрезке [-1, 0].

Найдем функции y1 = |f(x) \* f ’’(x)| и y2 = :

y1 =

y2 =

Проверим условия сходимости для метода Ньютона, построив графики функций y1 и y2, проверяя условие |f(x) \* f ’’(x)| < :

Изображение выглядит как текст, снимок экрана, компьютер

Автоматически созданное описание

Условие выполняется, значит, метод Ньютона может быть применен к этой функции.

**Метод дихотомии**

Метод дихотомии (метод половинного деления) – численный метод для решения нелинейных уравнений вида f(x) = 0. Предполагается непрерывность функции f(x). Из непрерывности следует, что на отрезке существует хотя бы один корень уравнения.

Для начала итераций необходимо знать отрезок [xL, xR], на концах которого функция принимает значения противоположных знаков. Это проверяется неравенством вида f(xL)\*f(xR) <0.

Далее находится значение xM - середины отрезка: xM = .  Если значения функции в середине отрезка и в начале отрезка разные, то нужно переместить правую границу в середину отрезка. В противном случае нужно переместить левую границу в середину отрезка. Затем действия повторяются, начиная с вычисления значения xM. Алгоритм заканчивается, когда f(xM) = 0 или xL = xR.

1. Проверим, удовлетворяет ли отрезок уравнения №10 данному условию:

уравнение №10: ; отрезок [0.4, 1].

f(0.4) = 0,31 – 0,92 = -0,61

f(1) = 1,68 – 0,54 = 1,14

f(0.4)\*f(1) = -0,61 \* 1,14 = -0,6954 < 0. Значит, отрезок удовлетворяет условию.

1. Проверим, удовлетворяет ли отрезок уравнения №11 данному условию:

уравнение №11: отрезок [-1, 0].

f(-1) = 0,37 + 0,93 – 2 = -0,7

f(0) = 1 + 0 – 2 = -1

f(0.4)\*f(1) = -0,61 \* 1,14 = -0,6954 < 0. Значит, отрезок удовлетворяет условию.

**Практическая часть**

**Описание переменных**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Описание** |
| fabs | double | Модуль числа |
| f10 | Исходное уравнение №10 |
| F10 | Уравнение №10, приведенное к методу итерации |
| fp10 | Производная от уравнения №10 |
| f11 | Исходное уравнение №11 |
| fp11 | Производная от уравнения №11 |
| iteration | Вычисление корня методом итерации |
| newt | Вычисление корня методом Ньютона |
| dichotomy | Вычисление корня методом дихотомии |
| a | Начало отрезка |
| b | Конец отрезка |

**Описание алгоритма**

В начале определяются используемые функции. Функция fabs возвращает модуль числа, функции f10 (исходная функция №10), F10 (функция №10, приведенная к виду x = f(x)), fp10 (производная от исходной функции №10), f11 (исходная функция №11), F11 (производная от исходной функции №11) возвращают соответствующую функцию. Далее идут функции, вычисляющие корень уравнения.

1. Метод итерации.

Сохраняем приближенное значение корня в переменной px (). Переменная x принимает значение итерационного процесса, равного f(px). Далее начинается основной цикл, который длится, пока модуль разности px и x больше эпсилон. В цикле px принимает значение x, а x – f(x).

1. Метод Ньютона.

Сохраняем приближенное значение корня в переменной px (). Переменная x принимает значение итерационного процесса, равного , где F(px) – производная от f(px). Далее начинается основной цикл, который длится, пока модуль разности px и x больше эпсилон. В цикле px принимает значение x, а x – .

1. Метод дихотомии.

Сохраняем приближенное значение корня в переменной x (). Переменной px присваиваем значение конца отрезка (b). Далее начинается основной цикл, который длится, пока модуль разности px и x больше эпсилон. Если произведение f(x) и f(a) больше нуля, то a получает значение x. В противном случае b приравнивается к x. Затем px приравнивается к x, а x к .

**Код программы**

#include <stdio.h>

#include <locale.h>

#include <math.h>

#define eps 0.0000001

double fabs(double x) {

if (x > 0) { return x; }

if (x < 0) { return x \* (-1); }

if (x == 0) { return 0; }

}

double f10(double x) {

return 2 \* x \* sin(x) - cos(x);

}

double F10(double x) {

return (((1 / (tan(x))) / 2.));

}

double fp10(double x) {

return 3 \* sin(x) + 2 \* x \* cos(x);

}

double f11(double x) {

return exp(x) + sqrt(1 + exp(2 \* x)) - 2;

}

double fp11(double x) {

return (exp(x) \* sqrt(exp(2 \* x) + 1) + exp(2 \* x)) / (sqrt(exp(2 \* x) + 1));

}

double iteration(double f(double), double a, double b) {

double px = (a + b) / 2., x = f(px);

while (fabs(x - px) > eps) {

px = x;

x = f(x);

}

return x;

}

double newt(double f(double), double F(double), double a, double b) {

double px = (a + b) / 2., x = px - f(px) / F(px);

while (fabs(px - x) > eps) {

px = x;

x = px - f(px) / F(px);

}

return x;

}

double dichotomy(double f(double), double a, double b) {

double px = b, x = (a + b) / 2.;

while (fabs(px - x) > eps) {

if (f(x) \* f(a) > 0) {

a = x;

}

else { b = x; }

px = x;

x = (a + b) / 2.;

}

return x;

}

int main() {

setlocale(LC\_ALL, "rus");

printf(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("| Уравнение | Отрезок | Решение методом итерации | Решение методом Ньютона | Решение методом дихотомии |\n");

printf("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n");

printf("|\t 2\*x\*sin(x) - cos(x) | [0.4, 1] | %.4lf | %.4lf | %.4lf |\n", iteration(F10, 0.4, 1.0), newt(f10, fp10, 0.4, 1.0), dichotomy(f10, 0.4, 1.0));

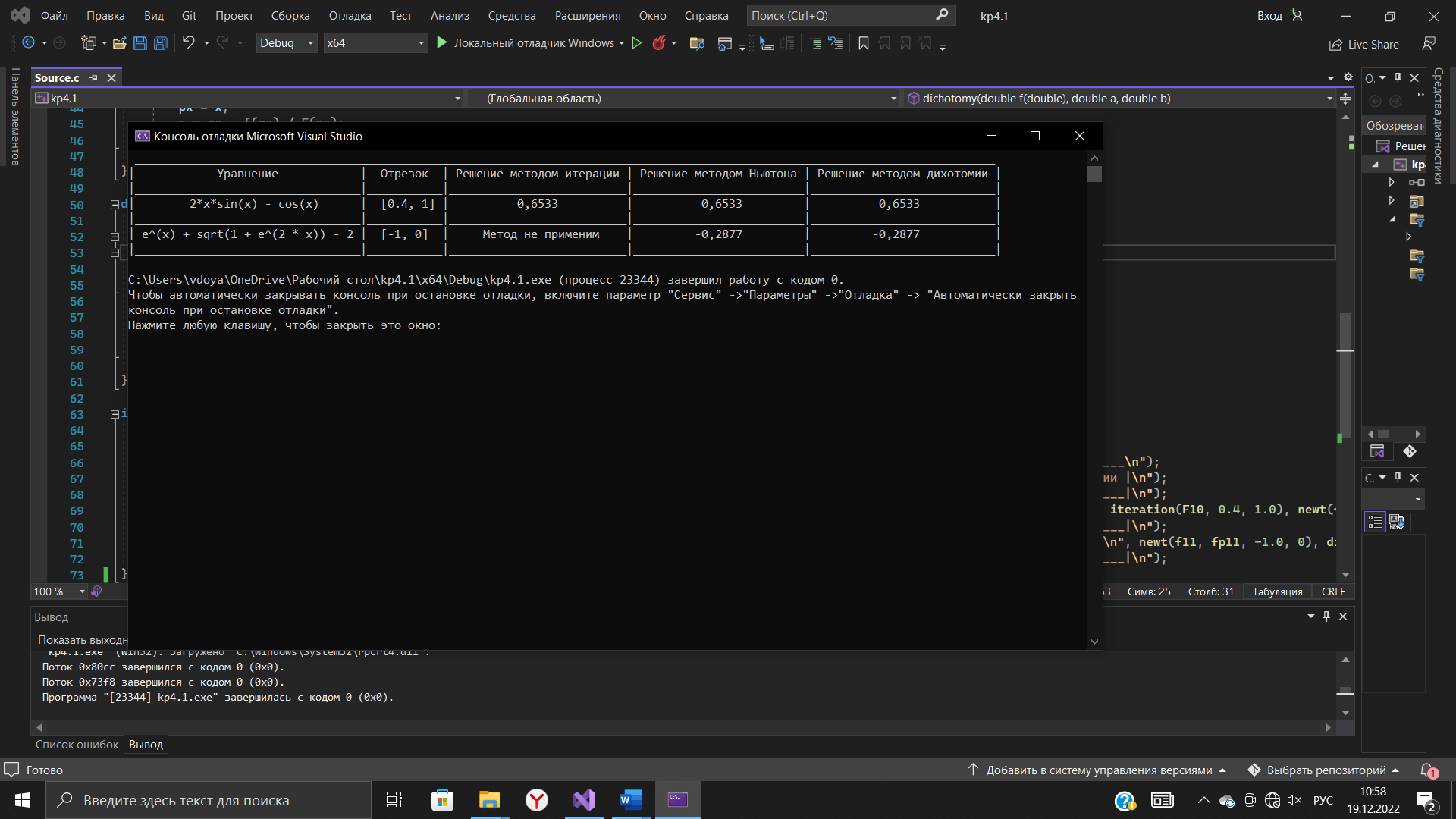
printf("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n");

printf("| e^(x) + sqrt(1 + e^(2 \* x)) - 2 | [-1, 0] | Метод не применим | %.4lf | %.4lf |\n", newt(f11, fp11, -1.0, 0), dichotomy(f11, -1.0, 0));

printf("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n");

}

**Выходные данные**



**Заключение**

В ходе выполнения данного задания курсового проекта я научилась реализовывать решения трансцендентных алгебраических уравнений различными численными методами (методом итерации, методом Ньютона, методом дихотомии) и доказывать невозможность решения определенным методом.

Используя сведения о численных методах, я составила программу с процедурами решения уравнений. Математически, используя графики функций и их производных, я доказала применимость и неприменимость тех или иных методов к решению уравнений.