Федеральное агентство связи  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
 «Сибирский государственный университет  
телекоммуникаций и информатики»

Кафедра ПМиК

Лабораторная работа №3 по дисциплине

«Вычислительная математика» по теме:

«Метод биссекций, метод хорд и метод Ньютона»

Выполнил:

ст. гр. ИВ-823

Шиндель Э. Д.

Проверила:

Ассистент

Кафедры ПМиК

Петухова Я. В.

Новосибирск, 2020

**Содержание:**

1. Постановка задачи…………………………………………………3

2. Метод биссекций………………………………………..…………3

3. Метод хорд…………………………………………………………4

4. Метод Ньютона…………………………………………………….6

5. Пример работы……………………………………………………..9

6. Листинг……………………………………………………………10

**1. Постановка задачи**

Проверка на наличие корней:

f(a) f(b) < 0 (есть хотя бы один корень)

f(a) = f(0,5) = 0,5605

f(b) = f(2) = -2,4298

f(a) f(b) = -1,3619 < 0 → Есть хотя бы 1 корень

**2. Метод биссекций**

a = 0,5 b = 2 = (0,0001) или 3 итерации

Проверка:

Для перехода на следующую итерацию находим интервал, где происходит смена знака:

if (f(a) f(x) < 0) a = a, b = x; [a; x]

else a = x, b = b; [x, b]

1-ая итерация:

a = 0,5 b = 2

Проверка:

f(a) f(x) = f(0,5) f(1,25) = 0,5605 0,7123 = 0,3992 > 0 → a = x

2-ая итерация:

a = 1,25 b = 2

Проверка:

f(a) f(x) = f(1,25) f(1,625) = 0,7123 (-0,8860) = -0,6311 < 0 → b = x

3-я итерация:

a = 1,25 b = 1,625

Проверка:

Так как мы выполнили 3 итерации, и не достигли нужной точности, то запишем ответ, сократив точность

**Ответ:** x = 1,4 с точностью 0,1.

**3. Метод хорд**

a = 0,5 b = 2 = (0,0001) или 3 итерации

Проверка:

Для перехода на следующую итерацию находим интервал, где происходит смена знака:

if (f(a) f(x) < 0) a = a, b = x; [a; x]

else a = x, b = b; [x, b]

1-ая итерация:

a = 0,5 b = 2

f(a) f(x) = f(0,5) f(0,7812) = 0,5605 1,4299 = 0,8402 > 0 → a = x

2-ая итерация:

a = 0,7812 b = 2

Проверка:

f(a) f(x) = f(0,7812) f(1,2327) = 1,4299 0,7721 = 1,1040 > 0 →a = x

3-я итерация:

a = 1,2327 b = 2

Проверка:

Так как мы выполнили 3 итерации, и не достигли нужной точности, то запишем ответ, сократив точность

**Ответ:** x = 1,4 с точностью 0,1.

**4. Метод Ньютона**

= (0,0001) или 3 итерации

Построим график и отметим точку решения.

Теперь выберем начальное приближение . Обычно это один из концов отрезка. Начальное приближение должно удовлетворять условию: .

Найдём 1 и 2 производную функции:

Проверяем левый конец отрезка:

Проверяем правый конец:

В качестве начального приближения выбираем

Проверка:

1-ая итерация:

Проверка:

2-ая итерация

Проверка:

3-я итерация:

1,4501

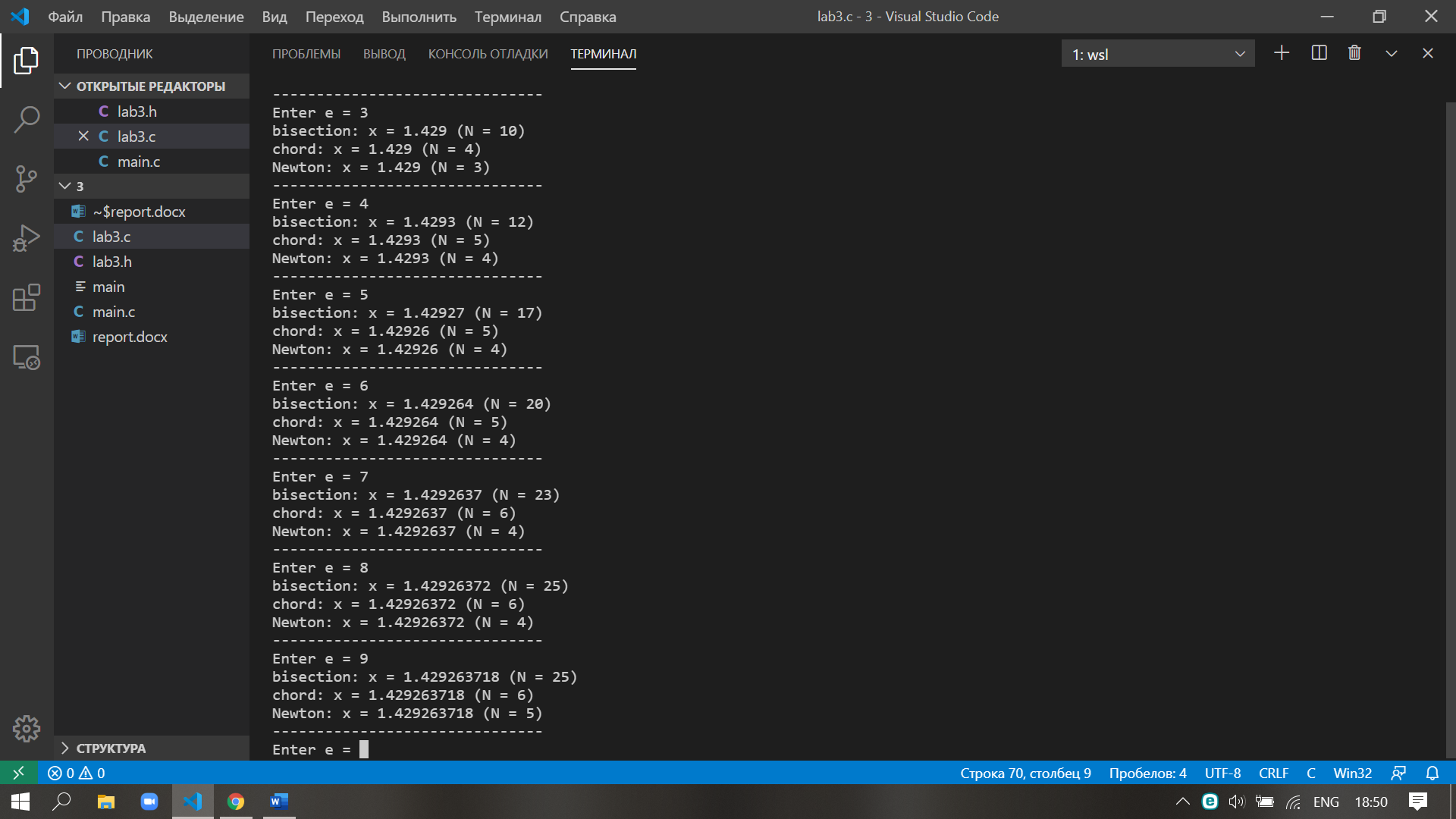
Проверка:

Так как мы выполнили 3 итерации, и не достигли нужной точности, то запишем ответ, сократив точность

**Ответ:** x = 1,4 с точностью 0,1.

**5. Пример работы**

При e = 3, 4, 5, 6, 7, 8, 9. (точность, кол-во знаков после запятой)



**Листинг**

**//lab3.c**

#include "lab3.h"

double aim(int x) {

double exp = 1.0;

for (int i = 0; i < x; i++) exp /= 10;

return exp;

}

double function(double x) {

double f = cos(x) + 2 \* sin(2 \* x) - 1 / x;

return f;

}

int check(double a, double b) {

double f\_a, f\_b;

f\_a = function(a);

f\_b = function(b);

return (((f\_a \* f\_b) < 0) ? 0 : 1);

}

void all\_in(int e) {

double a = 0.5, b = 2.0;

double dif = aim(e);

if (check(a, b)) {

printf("No roots in the interval [%.1f; %.1f]\n", a, b);

return;

}

bisection(e, dif, a, b);

chord(e, dif, a, b);

Newton(e, dif, a, b);

}

void bisection(int e, double dif, double a, double b) {

double x, f\_a, f\_x;

int N = 0;

do {

x = (a + b) / 2;

f\_a = function(a);

f\_x = function(x);

if ((f\_a \* f\_x) < 0) b = x;

else a = x;

N++;

} while (fabs(f\_x) > dif);

printf("bisection: x = %.\*f (N = %d)\n", e, x, N - 1);

}

void chord(int e, double dif, double a, double b) {

double x0 = 0.0, x1 = 0.0;

int N = 0;

do {

x0 = x1;

x1 = (a \* function(b) - b \* function(a)) / (function(b) - function(a));

if ((function(a) \* function(x1)) < 0) b = x1;

else a = x1;

N++;

} while (fabs(x1 - x0) > dif);

printf("chord: x = %.\*f (N = %d)\n", e, x1, N - 1);

}

void Newton(int e, double dif, double a, double b) {

double x0 = 0.0, x1 = 0.0, f1, f2;

int N = 0;

f2 = (function(a + dif) - 2 \* function(a) + function(a - dif)) / (dif \* dif);

if ((function(a) \* f2) > 0) x1 = a;

else x1 = b;

do {

x0 = x1;

f1 = (function(x0 + dif) - function(x0 - dif)) / (2 \* dif);

x1 = x0 - (function(x0) / f1);

N++;

} while (fabs(function(x0) / f1) > dif);

printf("Newton: x = %.\*f (N = %d)\n", e, x0, N - 1);

}