

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное

образовательное учреждение высшего образования

Санкт-Петербургский университет аэрокосмического приборостроения

КАФЕДРА № 2

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц., канд. техн. наук |  |  |  | С.Л. Козенко |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ № 1 |
| Нелинейные уравнения Вариант 5 |
| по дисциплине: ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4128 |  | 25.02.2023 |  | В.А. Воробьёв |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт – Петербург, 2023

|  |
| --- |
| Нелинейные уравнения Вариант 5 |
| по дисциплине: ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4128 |  | 25.02.2023 |  | В.А. Воробьёв |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт – Петербург, 2023

**СОДЕРЖАНИЕ**

[1 Цели и постановка задачи 3](#_Toc127557461)

[1.1 Цели работы 3](#_Toc127557462)

[1.2 Задание 3](#_Toc127557463)

[2 Описание метода решения 4](#_Toc127557464)

[3 Аналитические расчеты 5](#_Toc127557465)

[4 Схема алгоритма решения задачи 9](#_Toc127557466)

[5 Текст программы на C++ 13](#_Toc127557467)

[6 Результаты программных расчетов 14](#_Toc127557468)

[7 Сравнение программных и аналитических расчётов 16](#_Toc127557469)

[8 Выводы 18](#_Toc127557470)

1. **Цели и постановка задачи**

**1.1 Цели работы**

а) Освоение методов решения нелинейных уравнений;

б) Совершенствование навыков по алгоритмизации и программированию вычислительных задач.

**1.2 Задание**

Составить схему алгоритма и программу на языке C/C++ решения задачи по теме «Нелинейные уравнения» в соответствии с индивидуальным заданием.

|  |  |  |
| --- | --- | --- |
| ***Уравнение*** | ***Метод численного решения, точность*** | ***Параметры*** |
| =*0* | Дихотомии  ε = 4·10-5 | *a = 1.23; b = - 3.14* |

Рисунок 1 – Вариант задания

**2 Описание метода решения**

**Метод половинного деления (метод дихотомии)**

Пусть в уравнении функция **(*x*) является непрерывной (первое требование ”a”) на интервале *a*, *b*, в котором расположен один искомый корень *x*. Для нахождения этого корня разделим отрезок *a*, *b* пополам точкой



Если теперь **(**1) =0, то **1 и является корнем уравнения. В противном случае выбираем тот из отрезков *a*, **1 или **1, *b*, на концах которого функция **(*x*) имеет разные знаки. В пределах этого отрезка согласно предыдущим рассуждениям лежит искомый корень. Таким образом, оказывается определенным интервал *a*1, *b*1, (где *a*1=*a*, *b*1=**1 или *a*1=**1, *b*1= *b*), меньший первоначального *a*, *b* и содержащий *x**.* Повторяя подобные построения, получаем последовательность уменьшающихся интервалов *an*, *bn* таких, что

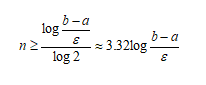


и в каждом из которых заключен корень *x*. Точность вычисления корня *x* определяется размерами интервала *an*, *bn* после *n*-го деления исходного интервала *a*, *b*, так как ошибка определения корня *x* не превышает величины *bn* – *an* .

Следовательно, если *ε* есть заданная точность вычисления, то должно выполняться условие



Отсюда можно определить и необходимое число шагов половинного деления интервала *a*, *b*, если задано *ε*:



**3 Аналитические расчеты**

Сначала строим график нашей функции (рис. 2).

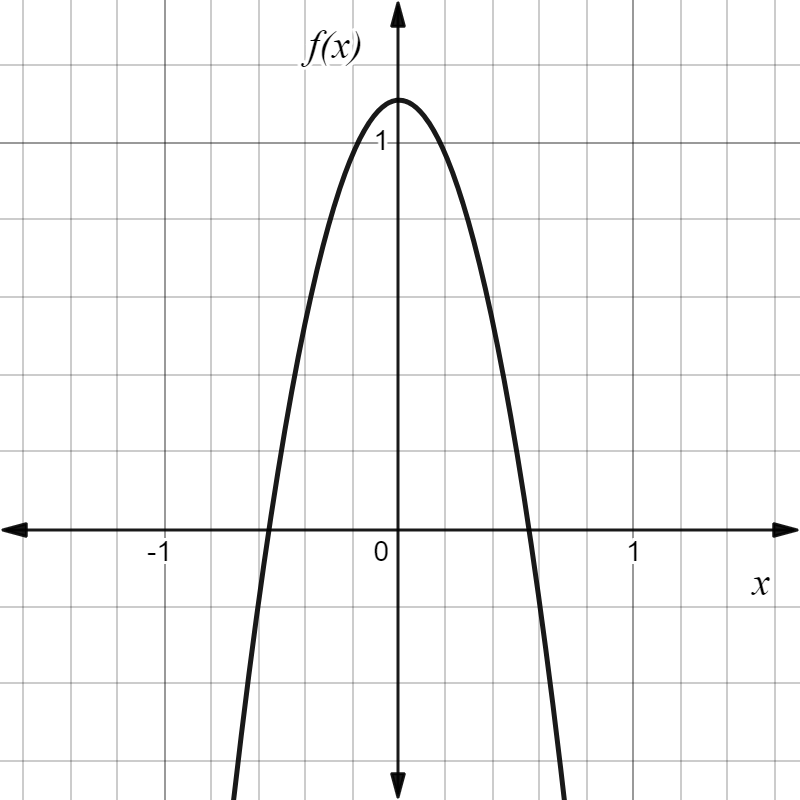


Рисунок 2 – График исходной функции

На графике видно, что f(x) пересекает ось X с в двух промежутках: [-1;0] и [0;1]. Проведем расчеты для этих промежутков с помощью онлайн калькулятора:

**3.1 Расчеты для корня на промежутке [-1;0]**

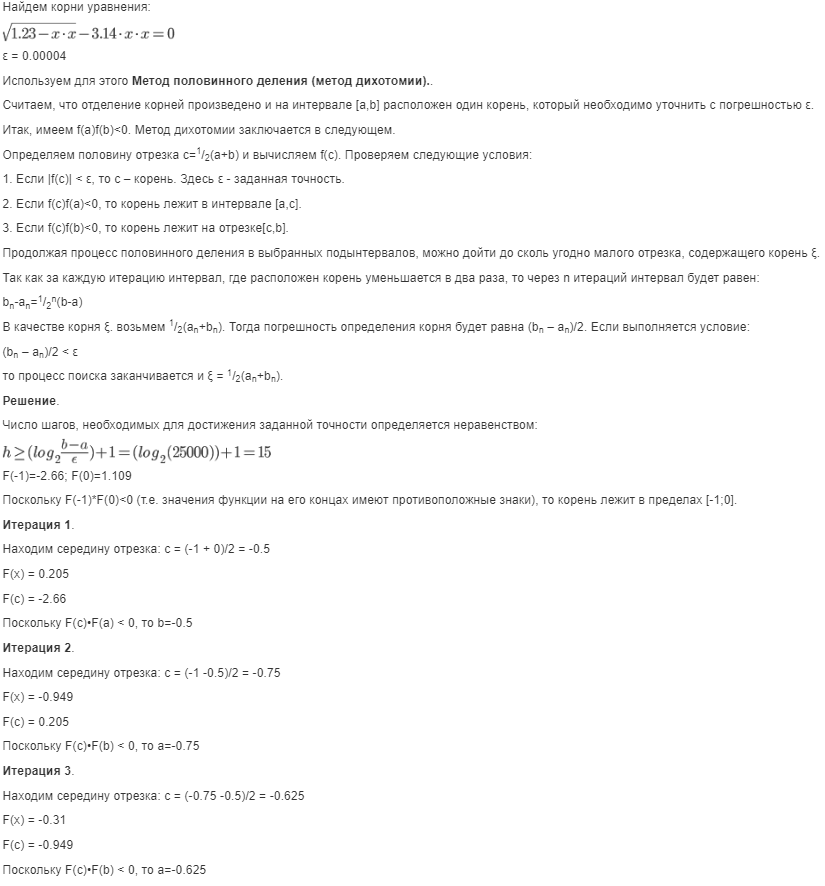


Рисунок 3 – решение с помощью онлайн-калькулятора

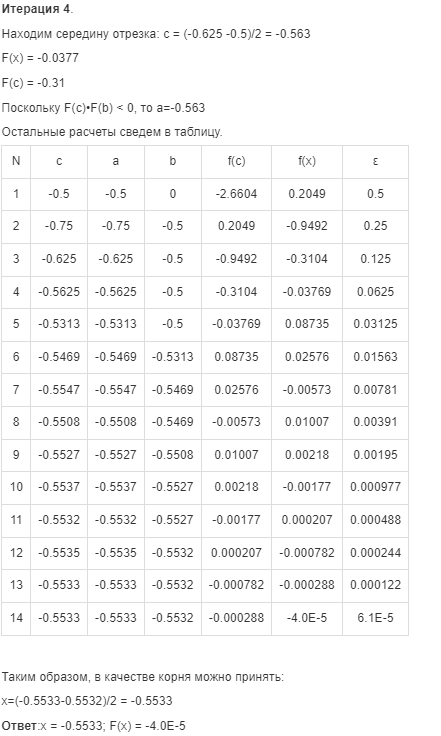


Рисунок 4 – решение с помощью онлайн-калькулятора

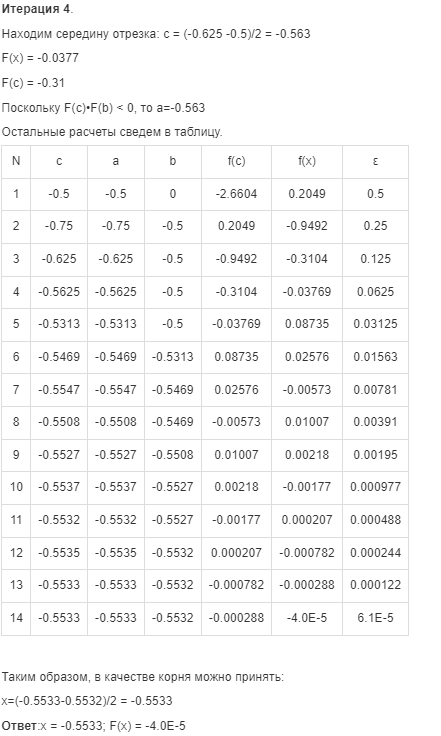


Рисунок 5 – решение с помощью онлайн-калькулятора

**3.2 Расчеты для корня на промежутке [0;1]**

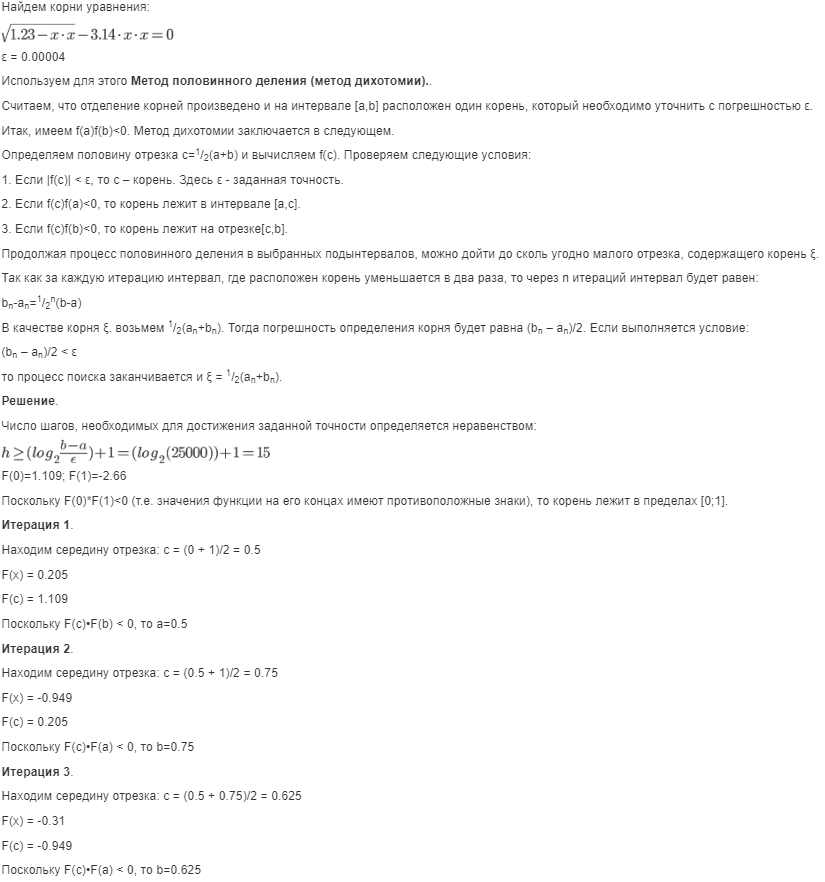
****

Рисунок 6 – решение с помощью онлайн-калькулятора

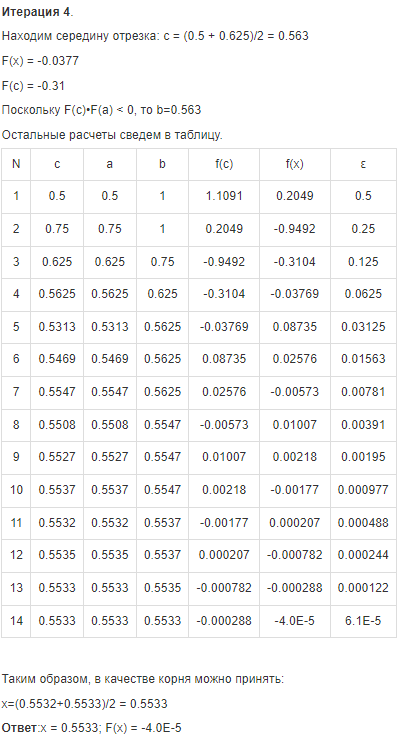
****

Рисунок 7 – решение с помощью онлайн-калькулятора

**4 Схема алгоритма решения задачи**

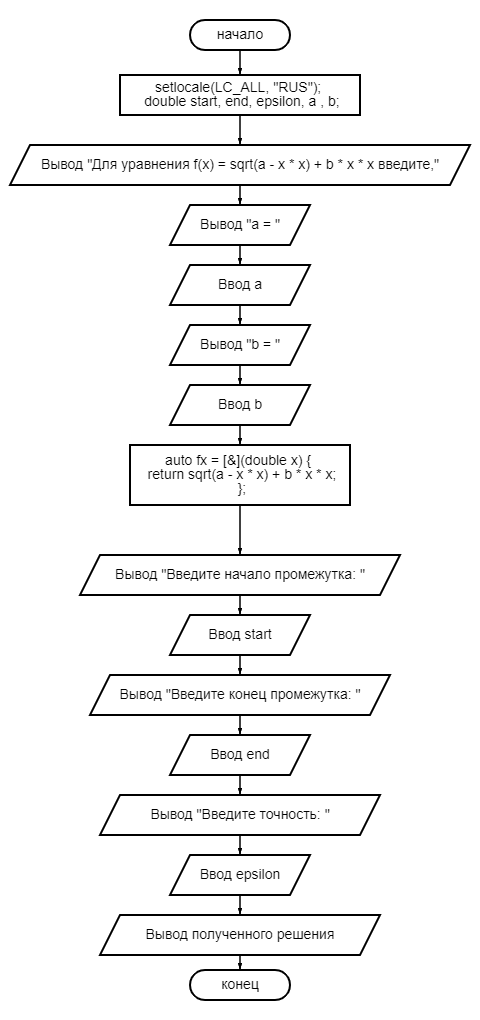
****

Рисунок 8 – Блок-схема основной функции



Рисунок 9 – Блок-схема dichotomy функции

**5 Текст программы на C++**

Исходный код доступен на GitHub: https://github.com/vladcto/SUAI\_homework/blob/a635fedd450ca83705278b949d5cc37749177647/4\_semester/ComputationalMathematics/program.cpp

#include <iostream>

#include <cmath>

#include <functional>

using namespace std;

double dichotomy(std::function<double(double)> f ,double a, double b, double epsilon) {

double c;

int step = 0;

cout << "Начало нахождения решения методом половинного деления." << endl;

while (abs(a - b) > epsilon) {

c = (a + b) / 2;

cout << "Шаг " << step++ << ": f(" << c << ") = " << f(c) << endl;

if (f(c) == 0) return c;

f(a)\* f(c) < 0 ? b = c : a = c;

}

return (a + b) / 2;

}

int main() {

setlocale(LC\_ALL, "RUS");

double start, end, epsilon, a , b;

cout << "Для уравнения f(x) = sqrt(a - x \* x) + b \* x \* x введите," << endl;

cout << "a = ";

cin >> a;

cout << "b = ";

cin >> b;

auto fx = [&](double x) {

return sqrt(a - x \* x) + b \* x \* x;

};

cout << "Введите начало промежутка: ";

cin >> start;

cout << "Введите конец промежутка: ";

cin >> end;

cout << "Введите точность: ";

cin >> epsilon;

cout << "Решением является: " << dichotomy(fx, start, end, epsilon) << endl;

return 0;

}

**6 Результаты программных расчетов**

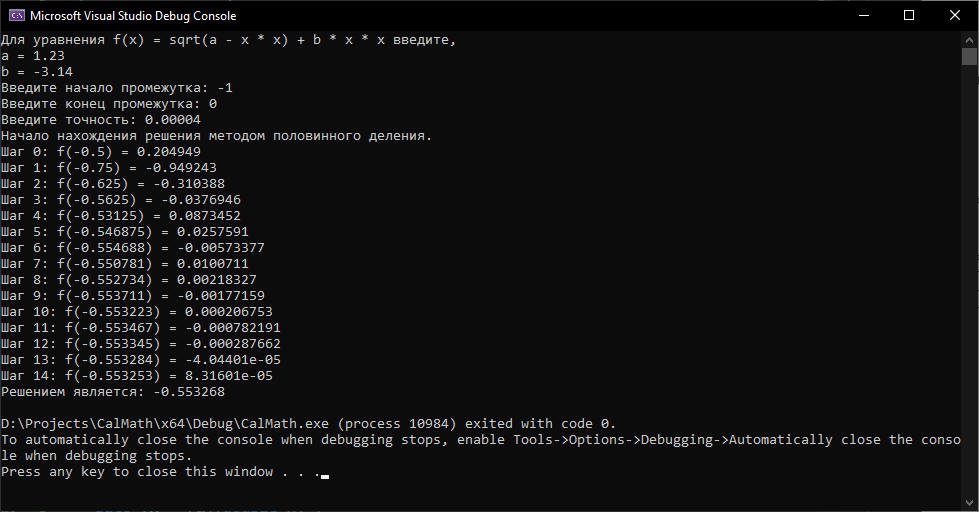


Рисунок 10 – Результат работы программы с предоставленными исходными данными

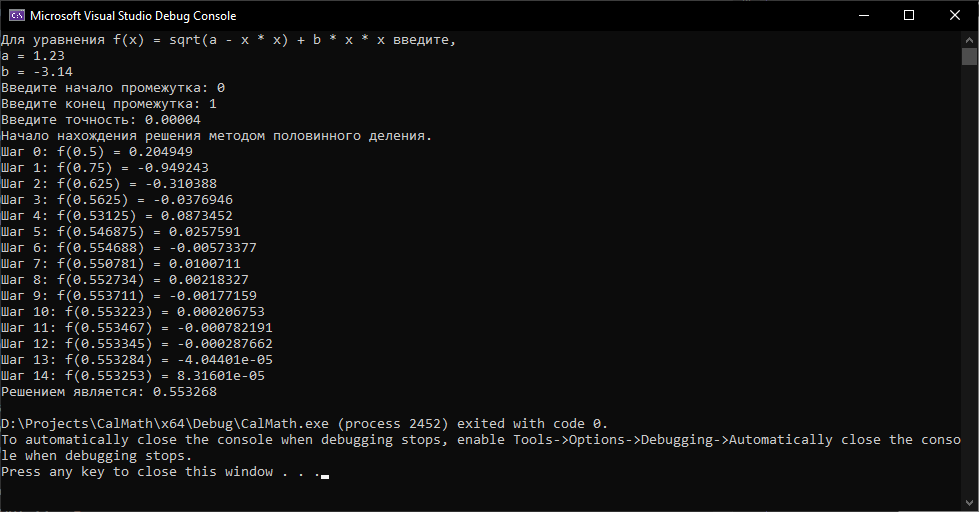


Рисунок 11 – Результат работы программы с произвольными исходными данными

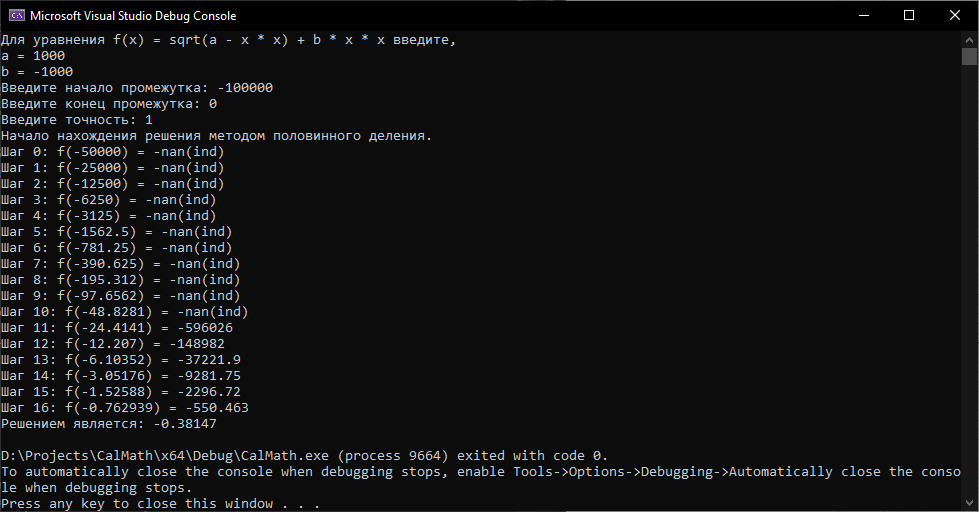


Рисунок 12 – Результат работы программы с произвольными исходными данными



Рисунок 13 – Пример работы алгоритма для корня -0.5533

**7 Сравнение программных и аналитических расчётов**

Результаты сравниваемых программных и аналитических расчетов были представлены в прошлых разделах.

Сейчас сравним нахождения корня для промежутка [-1;0] (см. Рис. 14). Все шаги совпадают с решением онлайн-калькулятора, тем не менее наш метод вследствие отсутствия ограничения на количество выводимых символов типа double, более точно показывает шаги. Полученные значения совпадают с графиком (см. рис. 2).

|  |  |  |
| --- | --- | --- |
| **№** | **x** | **x\*** |
| **1** | **-0.5** | **-0.5** |
| **2** | **-0.75** | **-0.75** |
| **3** | **-0.625** | **-0.625** |
| **4** | **-0.5625** | **-0.5625** |
| **5** | **-0.5313** | **-0.5313** |
| **6** | **-0.5469** | **-0.5469** |
| **7** | **-0.5547** | **-0.5547** |
| **8** | **-0.5508** | **-0.5508** |
| **9** | **-0.5527** | **-0.5527** |
| **10** | **-0.5537** | **-0.5537** |
| **11** | **-0.5532** | **-0.5532** |
| **12** | **-0.5535** | **-0.5535** |
| **13** | **-0.5533** | **-0.5533** |
| **14** | **-0.5533** | **-0.5533** |

Рисунок 14 – Таблица сравнения для корня в промежутке [-1;0]

Теперь сравним нахождения корня для промежутка [0;1] (см. рис. 15). Аналогично сопоставлению промежутка [-1;0], на промежутке [0;1] все шаги совпадают. Мы можем утверждать, что реализованный нами алгоритм является верным.

|  |  |  |
| --- | --- | --- |
| **N** | **x** | **x\*** |
| **1** | **0.5** | **0.5** |
| **2** | **0.75** | **0.75** |
| **3** | **0.625** | **0.625** |
| **4** | **0.5625** | **0.5625** |
| **5** | **0.5313** | **0.5313** |
| **6** | **0.5469** | **0.5469** |
| **7** | **0.5547** | **0.5547** |
| **8** | **0.5508** | **0.5508** |
| **9** | **0.5527** | **0.5527** |
| **10** | **0.5537** | **0.5537** |
| **11** | **0.5532** | **0.5532** |
| **12** | **0.5535** | **0.5535** |
| **13** | **0.5533** | **0.5533** |
| **14** | **0.5533** | **0.5533** |

Рисунок 15 – Таблица сравнения для корня в промежутке [0;1]

**8 Выводы**

В ходе выполнения лабораторной работы были составлены схема алгоритма и программа на языке C++, решающая поставленную задачу в соответствии с вариантом. Результаты при аналитических и программных расчётах оказались одинаковы – итог и все промежуточные значения оказались равны.

В результате вычислений были получены два корня, что соответствует графику:

х1 = -0.5533;

х2 = 0.5533.

В результате выполнения работы был освоен метод дихотомии для решения нелинейных уравнений. Этот метод основан на последовательном разделение отрезка на два одинаковых, что схоже с алгоритмом бинарного поиска. Также в результате выполнения работы были усовершенствованы навыки по алгоритмизации и программированию вычислительных задач.