**Московский Авиационный Институт**

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Курсовая работа

по курсу “Фундаментальная информатика”

Задание III: Вещественный тип. Приближенные вычисления. Табулирование функций

Студент: Лагуткина М. С.

Группа: М8О-106Б-19

Руководитель: Дубинин А. В.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Задание**

Рассматривается уравнение вида F(x)=0. Предполагается, что функций F(x) достаточно гладкая, монотонная на этом отрезке и существует единственный корень уравнения x\* ∈ [a,b]. На отрезке [a, b] ищется приближенное решение x с точностью ε, т. е. такое, что |x - x\*| < ε.

Составить программу на языке Си с процедурами решения трансцедентных алгебраических уравнений различными численными методам (итераций, Ньютона и половинного деление — дихотомии). Уравнения оформить как функции параметры, разрешив относительно неизвестной величины в случае необходимости. Применить каждую процедуру к решению двух уравнений — заданного вариантом и следующего за ним

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Уравнение | Отрезок | Базовый метод | Приближенное значение корня |
| 23 |  | [2,4] | Ньютона | 3.23 |
| 24 |  | [1,2] | Дихотомии | 1.8756 |

.

## Метод Дихотомии

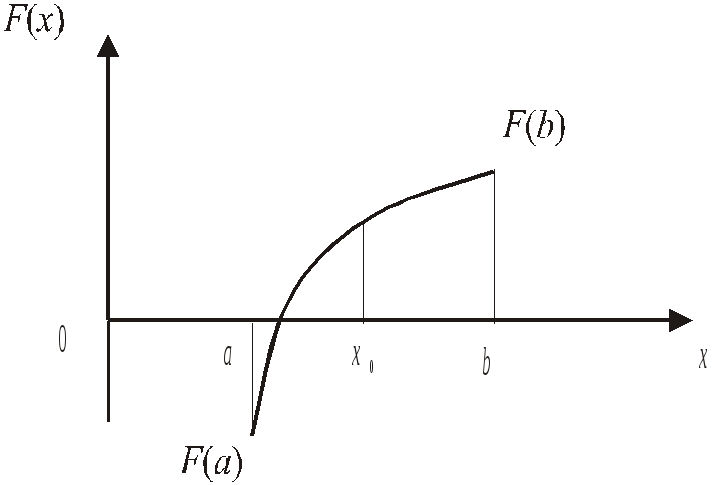
Очевидно, что если на отрезке [a, b] существует корень уравнения, то значения функции на концах отрезка имеют разные знаки: . Метод заключается в делении отрезка пополам и его сужении в два раза на каждом шаге итерационного процесса в зависимости от знака функции в середине отрезка.

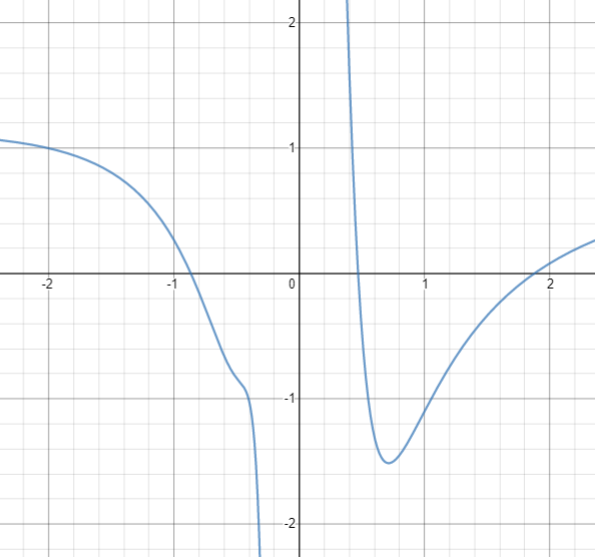
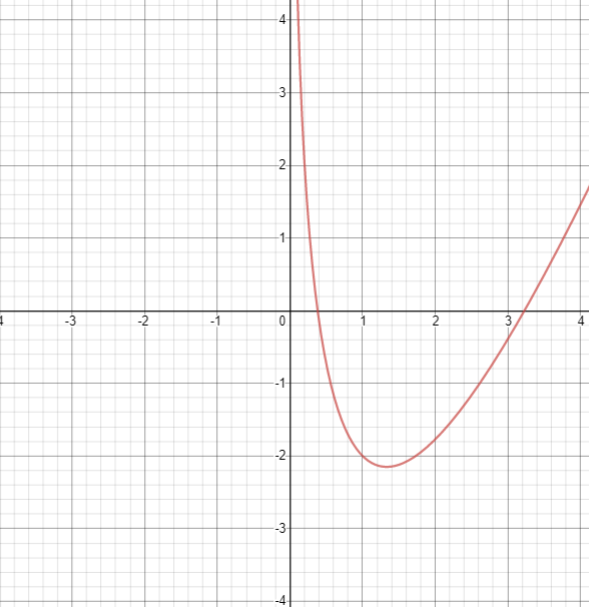
* За начальное приближение принимаются границы исходного отрезка:

.

* Итерационный процесс:
* Условие окончания: .
* Приближенное значение корня:

То есть мы делим наш отрезок пополам и смотрим, на каком из двух получившихся отрезков произведение значений на его концах - отрицательное число. Затем выбираем его первоначальным и повторяем наши действия, пока изменение не будет меньше выбранного ε.





## Метод итераций

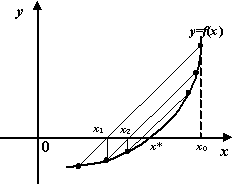
Метод простой итерации — один из простейших численных методов решения уравнений. Метод основан на принципе сжимающего отображения, который применительно к численным методам в общем виде также может называться методом последовательных приближений.

Идея метода заключается в замене исходного уравнения уравнением вида . Такое преобразование можно делать разными способами. В частности, сохраняет корни уравнение вида т.е в . Постоянная l должна быть того же знака, что и производная на отрезке. В программе используем , т.к. для лучшей сходимости необходимо взять производную вблизи корня.

Достаточное условие сходимости метода , . Это условие необходимо проверить перед началом решения задачи, так как функция может быть выбрана неоднозначно, причем в случае неверного выбора указанной функции метод расходится.

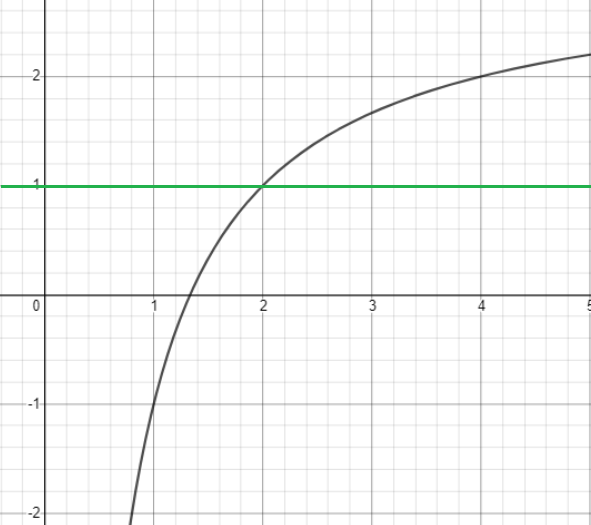
В программе берем

* Начальное приближение корня: .
* Итерационный процесс: .
* Условие окончания: .
* Приближенное значение корня: x\* ≈ xконечное



Для данных функций не выполняется метод итераций, так как не выполняется условие

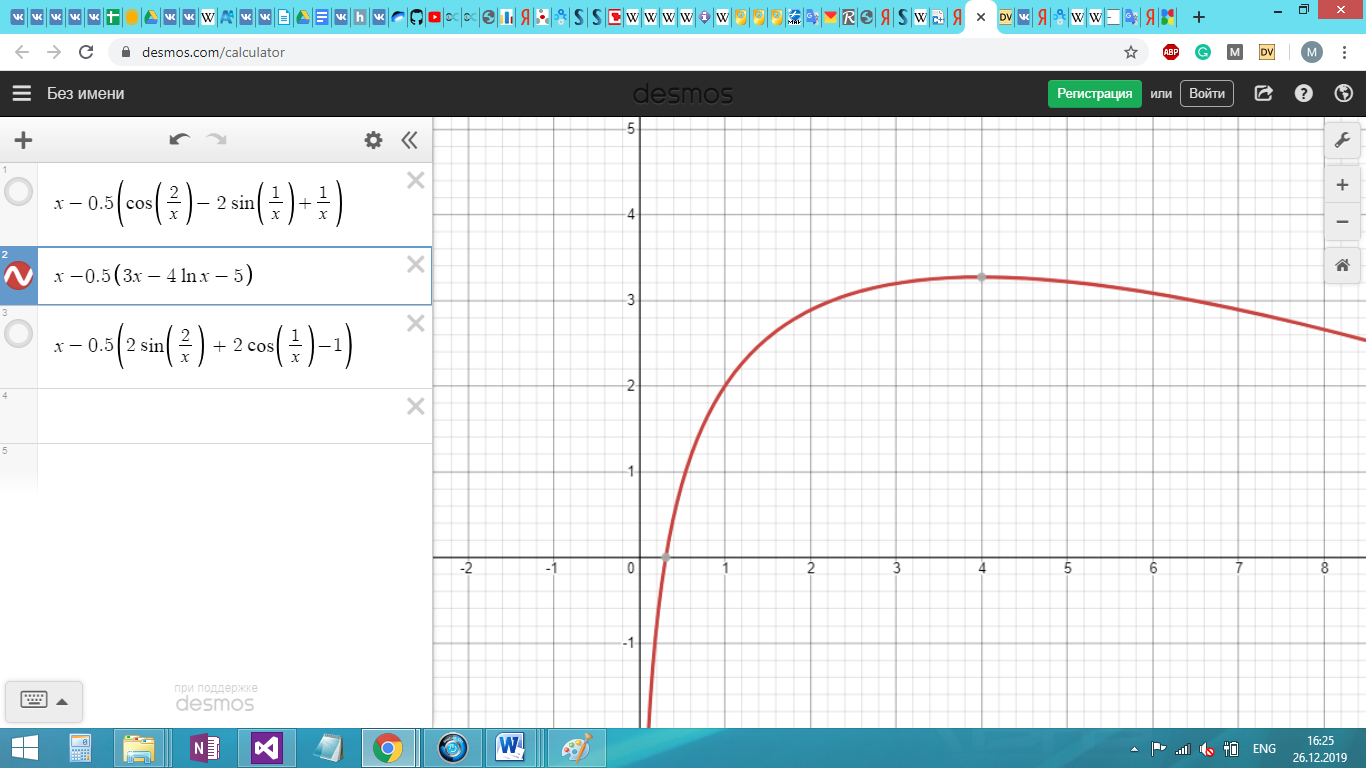
Обоснование:

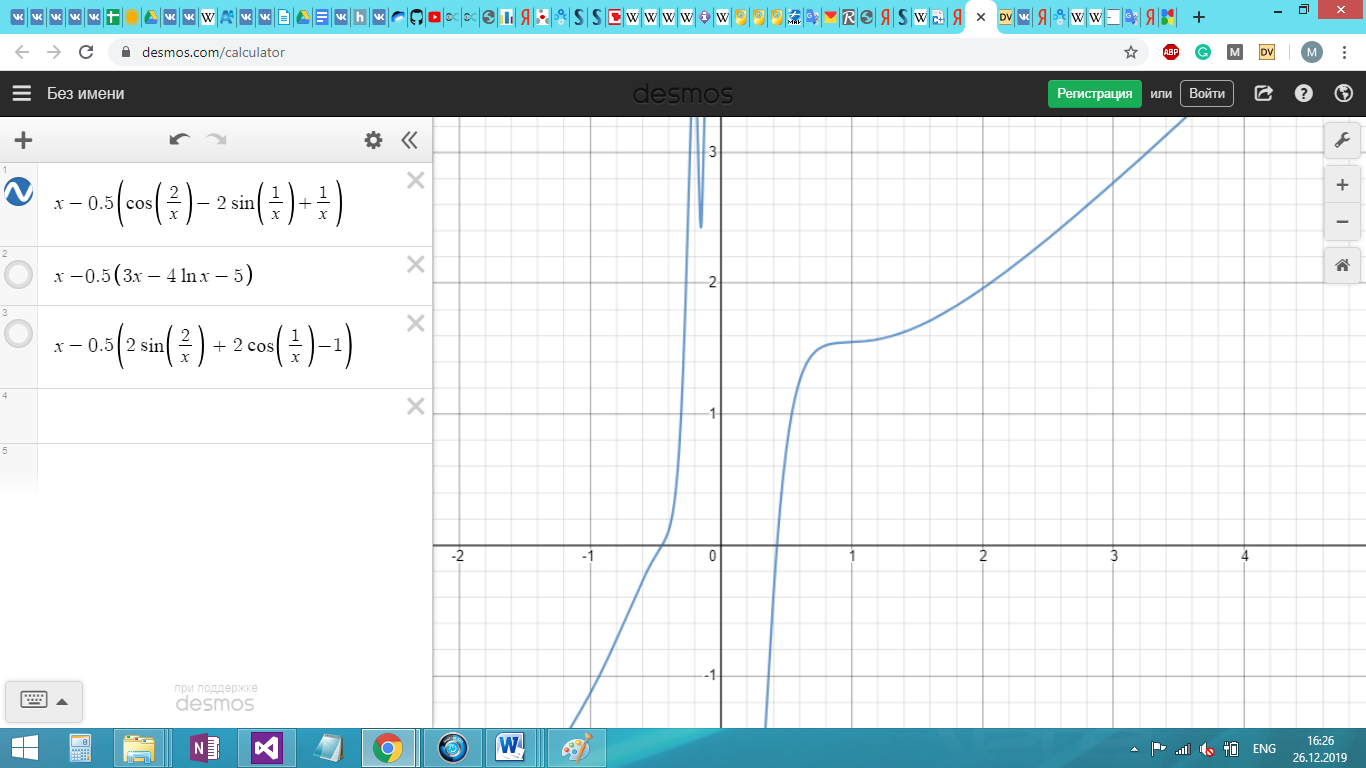


Для того, чтобы можно было применить данный метод, можно заменить уравнения вида уравнениями вида .

Полученные уравнения:

= 0





Для этих функций выполняется метод итераций, так как не выполняется условие

## Метод Ньютона

Метод Ньютона является частным случаем метода итераций. Условие сходимости метода: на отрезке [a, b]. Итерационный процесс:

Поиск решения осуществляется путём построения последовательных приближений и основан на принципах простой итерации. Вместо теперь берем . Метод обладает квадратичной сходимостью.

Изображение выглядит как лазер

Автоматически созданное описание

**Описание программы**

С помощью typedef создаем структурный тип f\_result для функций, который хранит информацию об: успехе вычислений, количестве итераций и приближенном значении корня. Пишем функции: возвращающую значение машинного эпсилон, считающие первую и вторую производные.

На основе теории об раннее описанных методах вычисления корней создаем три функции, соответствующие данным методам. Они принимают функцию, описывающую уравнение, в качестве параметра, границы окрестности корня.

В функции main вызываем данные функции, оформляем результаты вычислений в виде таблицы.

**Вывод**

Использование структурных типов в качестве возвращаемого значения функции позволяет возвращать больше одного значения, что делает программу более модульной и компактной. А передача математических функций в качестве параметра другим функциям позволяет повторно использовать данные функции для разных математических уравнений, что упрощает код самой программы.

Среди рассмотренных методов самым быстрым оказался метод Ньютона, а самым медленным ¬– метод Дихотомии. При этом все методы оказались применимы и их результаты совпадают с точностью до машинного эпсилон.