Министерство науки и высшего образования Российской Федерации

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Наименование учебного структурного подразделения

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

по дисциплине «Наименование дисциплины»

НАИМЕНОВАНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

Фамилия Имя Отчество обучающегося

Направление подготовки Код Наименование направления подготовки

Направленность (профиль) «Наименование образовательной программы»

Руководитель работы

ученая степень, звание

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_И.О. Фамилия

*подпись*

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_ г.

Автор работы

студент группы № \_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.О. Фамилия

*подпись*

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_ г.

Томск – 2023

[Оглавление:](#_Toc151071306)

[ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 1](#_Toc151071439)

[Оглавление 2](#_Toc151071440)

[Постановка задачи: 3](#_Toc151071441)

[Теоретическая часть 4](#_Toc151071442)

[Программная реализация 5](#_Toc151071443)

[Вывод результатов тестирования в виде таблицы: 7](#_Toc151071444)

[Заключение: 8](#_Toc151071445)

# ****Постановка задачи:****

* **Цель работы****:** Целью данной лабораторной работы является изучение и реализация метода хорд для численного решения уравнений. Метод хорд — это один из численных методов решения уравнений, который использует приближенные значения функции для нахождения их корней.
* **Задачи:**
  + **Определение теоретических основ метода хорд:** в рамках этой задачи будет проведен анализ теоретических основ метода хорд, включая математические выкладки и объяснения принципов работы метода.
  + **Разработка программы, реализующей метод хорд:** для выполнения этой задачи будет создана программная реализация метода хорд, использующая выбранную среду разработки Python.
  + **Проведение численных экспериментов для оценки эффективности метода:** в рамках этой задачи будут проведены численные эксперименты с использованием разработанной программы для оценки эффективности метода хорд при решении различных уравнений.
* **Среда разработки:** для выполнения лабораторной работы будет использована среда разработки Python.

Теоретическая часть**:**

**Описание метода хорд:** Метод хорд (или метод секущих) — это численный метод для приближенного нахождения корней уравнений. Он использует линейную интерполяцию между двумя точками на графике функции, чтобы приближенно найти корень. Основная идея заключается в следующем:

1. Выбираются две начальные точки, которые находятся с разных сторон от искомого корня.
2. Строится секущая линия (хорда), проходящая через эти две точки.
3. Точка пересечения секущей линии с осью x становится новой аппроксимацией корня.
4. Процесс повторяется до достижения заданной точности или максимального числа итераций.

**Математические выкладки:** Давайте рассмотрим более подробно математические выкладки метода хорд. Пусть *f*(*x*) - функция, у которой мы ищем корень, и заданы две начальные точки *x*0​ и *x*1​ (где *f*(*x*0​) и *f*(*x*1​) имеют разные знаки). Тогда:

Сначала мы строим уравнение секущей линии (хорды) между точками

Затем мы приравниваем *y* к нулю и решаем уравнение для *x*, чтобы найти новую аппроксимацию корня:

Это новое значение *x* используется в следующей итерации, и процесс повторяется до достижения заданной точности или максимального числа итераций.

**Основные понятия:**

* **Точность:** Точность метода хорд зависит от выбора начальных точек и числа итераций. Чем больше итераций выполняется, тем ближе к истинному корню мы приближаемся. Точность можно контролировать путем задания критерия остановки, например, минимальной разницы между последовательными приближениями.
* **Сходимость:** Сходимость метода хорд зависит от формы графика функции и расположения начальных точек. Если начальные точки близки к корню и функция не слишком "плоская" в окрестности корня, метод может сходиться быстро. Однако, в случае множественных корней или "плоских" участков графика, метод может сходиться медленно или даже расходиться. *Условием сходимости метода хорд является то, что между начальными точками и корнем функции не должно быть двух перегибов и более* (точек, где производная меняет свой знак)
* **Условия применимости:** Метод хорд может быть применен для нахождения корней непрерывных функций, если начальные точки выбраны так, что они находятся с разных сторон от корня и функция непрерывна в интервале между ними. Если функция имеет разрывы или вертикальные асимптоты, метод хорд может не сойтись или дать некорректный результат.

Программная реализация**:**

Алгоритм метода хорд в программной реализации может быть описан следующим образом:

1. Задать начальные приближения *x*0​ и *x*1​, где *f*(*x*0​) и *f*(*x*1​) имеют разные знаки.
2. Для каждой итерации выполнить следующие шаги:
   * Вычислить новое приближение корня *x* с использованием формулы метода хорд:
   * Проверить условие остановки, например, если достигнута заданная точность или выполнено максимальное число итераций.
   * Если условие остановки не выполнено, обновить �0*x*0​ и �1*x*1​ для следующей итерации и продолжить.

**Код программы:** В программной реализации метода хорд вам потребуется написать код на языке программирования, который будет выполнять описанный выше алгоритм. Код может выглядеть следующим образом на языке Python:

def chord\_method(f, a, b, eps=1e-6, max\_iter=1000):

    import math

    h = 0.1

    d\_f = lambda x: (f(x + h) - f(x - h)) / (2 \* h)

    dd\_f = lambda x: (d\_f(x + h) - d\_f(x - h)) / (2 \* h)

    # проверяем условие существования корня на отрезке [a, b]

    delt = (a + b) / 100

    u = delt + a

    for i in range(100):

        try:

            if dd\_f(u) \* dd\_f(u - delt) < 0:

                return (

                    "Условие существования корня не выполнено на отрезке [{}, {}]".format(

                        a, b

                    )

                )

        except:

            print("деление на ноль")

    if f(a) \* f(b) >= 0:

        return "Условие существования корня не выполнено на отрезке [{}, {}]".format(

            a, b

        )

    # Инициализация начальных значений

    fa, fb = f(a), f(b)

    iter\_count = 0

    last\_iter = -99999

    c=0

    # Итерационный цикл

    if d\_f(a) \* dd\_f(a) >= 0:

        while abs(a - last\_iter) > eps and iter\_count < max\_iter:

            # Вычисление значения функции в точке c

            c = a - (fa \* (b - a)) / (fb - fa)

            fc = f(c)

            last\_iter = 0 + a

            a, fa = c, fc

            iter\_count += 1

# Итерационный цикл

    if d\_f(a) \* dd\_f(a) < 0:

        while abs(b - last\_iter) > eps and iter\_count < max\_iter:

            # Вычисление значения функции в точке c

            c = b - (fb \* (b - a)) / (fb - fa)

            fc = f(c)

            last\_iter = 0 + b

            b, fb = c, fc

            iter\_count += 1

    # Возвращение приближенного значения корня или сообщение об ошибке

    if iter\_count == max\_iter:

        return f"Метод не сошелся за {max\_iter} итераций"

    else:

        return (c + last\_iter) / 2

if \_\_name\_\_ == "\_\_main\_\_":  # тесты

    f = lambda x: math.log(5, 5 \* x)

    test = chord\_method(f, a=0.5, b=1)

    print(test)

**Входные и выходные данные:** Входными данными для программы являются функция *f*(*x*), начальные приближения *x*0​ и *x*1​, заданная точность *eps*, и максимальное число итераций *max*\_*iter*. Выходными данными является найденный корень или информация о том, что метод не сошелся.

# ****Вывод результатов тестирования в виде таблицы:****

| **Уравнение** | **Метод хорд** | **Метод половинного деления** |
| --- | --- | --- |
|  | Найденный корень:  2.0 | Найденный корень:  2.0 |
|  | Кол-во итераций:  6 | Кол-во итераций:  3 |

|  |  |  |
| --- | --- | --- |
|  | Найденный корень:  0.5236 | Найденный корень:  0.5236 |
|  | Кол-во итераций:  4 | Кол-во итераций:  4 |

|  |  |  |
| --- | --- | --- |
|  | Найденный корень:  0.6931 | Найденный корень:  0.6931 |
|  | Кол-во итераций:  7 | Кол-во итераций:  17 |

# Заключение:

**Подведение итогов:** в ходе выполнения лабораторной работы был изучен и реализован метод хорд для численного решения уравнений. Программа успешно находит корни уравнений с учетом заданной точности и ограниченного числа итераций.

**Оценка метода:** Метод хорд является эффективным численным методом для нахождения корней уравнений. Его эффективность зависит от правильного выбора начальных приближений и свойств функции. В данной работе метод хорд успешно справился с разными уравнениями.

**Пути улучшения:** для дальнейших исследований можно рассмотреть адаптивные методы выбора начальных приближений и более сложные стратегии остановки, чтобы повысить эффективность метода хорд.