Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное   
учреждение высшего образования

Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

**Отчет по лабораторной работе**

**«?»**

**Выполнил**:

студент группы 3821Б1ПМ2

Головин Р.М.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2021

**Оглавление**

[Постановка задачи 3](#_Toc98681861)

[Методы решения 4](#_Toc98681862)

[Вычисление следующего члена ряда 4](#_Toc98681863)

[Обратное суммирование 4](#_Toc98681864)

[Прямое суммирование 4](#_Toc98681865)

[Попарное суммирование 4](#_Toc98681866)

[Руководство пользователя 5](#_Toc98681867)

[Описание программной реализации 6](#_Toc98681868)

[Подтверждение корректности 9](#_Toc98681869)

[Результаты экспериментов 10](#_Toc98681870)

[Синус 10](#_Toc98681871)

[Косинус 10](#_Toc98681872)

[Экспонента 10](#_Toc98681873)

[Логарифм 10](#_Toc98681874)

[Заключение 11](#_Toc98681875)

# Постановка задачи

Целью лабораторной работы являлась: реализовать вычисление значения функций sin, cos, exp, ln в точке с помощью разложения в ряд Маклорена. Писать программу необходимо на языке Си. Реализовать методы прямого, попарного и обратного суммирования элементов разложения. Нужно описать реализацию и алгоритмы работы программы. Необходимо подтвердить корректность реализации вычисления данных функций . Провести эксперименты по замеру точности различных методов суммирования.

# Методы решения

## Вычисление следующего члена ряда

Для каждой математической функции sin, cos, exp, ln реализована функция для вычисления следующего члена ряда через предыдущий.

## Обратное суммирование

Реализовано рекурсивно. Рекурсия продолжается до момента пока вычисляемый элемент не станет меньше заданного значения. При достижении указанного порога функция возвращает следующий элемент и происходит сложение.

## Прямое суммирование

Сложение данным методом происходит последовательно, начиная с первого элемента. К первому элементу прибавляется второй, за тем вычисляется третий элемент и прибавляется к результату сложения предыдущих и так далее. Процесс продолжается до момента пока вычисляемый элемент не станет меньше заданного значения. После достижения указанного порога функция возвращает результат.

## Попарное суммирование

Сложение данным методом происходит последовательно, начиная с первого элемента. К сумме двух первых элементов прибавляется сумма двух следующих, за тем вычисляется сумма следующей пары элементов и прибавляется к результату сложения предыдущих и так далее. Процесс продолжается до момента пока вычисляемый элемент не станет меньше заданного значения. После достижения указанного порога функция возвращает результат.

# Руководство пользователя

# Описание программной реализации

**double revSum**(double (\*nextEl)(double x, double pref, int i),double first, double x)

Функция задает начало для рекурсивного суммирования.

* double (\*nextEl)(double x,double pref, int i) – ссылка на функцию для вычисления следующего члена последовательности.
* double first – первый элемент последовательности.
* double x – точка, в которой вычисляется значение функции.

**static double recSum**(double (\*nextEl)(double x, double pref,int i),double x, double pref, int i)

Функция выполняет рекурсивное суммирование , начиная с конца ряда.

* double (\*nextEl)(double x,double pref, int i) – ссылка на функцию для вычисления следующего члена последовательности.
* double first – первый элемент последовательности.
* double x – точка, в которой вычисляется значение функции.
* int i – номер шага.

**double straightSum**(double (\*nextEl)(double x, double pref, int i),double first, double x)

Функция выполняет последовательное сложение элементов ряда , начиная с первого.

* double (\*nextEl)(double x,double pref, int i) – ссылка на функцию для вычисления следующего члена последовательности.
* double first – первый элемент последовательности.
* double x – точка, в которой вычисляется значение функции.

**double nearbySum**(double (\*nextEl)(double x, double pref, int i), double first, double x)

Функция выполняет последовательное попарное сложение элементов ряда , начиная с начала.

* double (\*nextEl)(double x,double pref, int i) – ссылка на функцию для вычисления следующего члена последовательности.
* double first – первый элемент последовательности.
* double x – точка, в которой вычисляется значение функции.

**double nextSin**(double x, double pref, int i)

Функция вычисляет следующий член ряда Макларена через предыдущий для математической функции sin(x).

* double x – точка, в которой вычисляется значение функции.
* double pref – значение предыдущего члена последовательности.
* int i – номер вычисляемого элемента.

**double nextCos**(double x, double pref, int i)

Функция вычисляет следующий член ряда Макларена через предыдущий для математической функции cos(x).

* double x – точка, в которой вычисляется значение функции.
* double pref – значение предыдущего члена последовательности.
* int i – номер вычисляемого элемента.

**double nextExp**(double x, double pref, int i)

Функция вычисляет следующий член ряда Макларена через предыдущий для математической функции e^x.

* double x – точка, в которой вычисляется значение функции.
* double pref – значение предыдущего члена последовательности.
* int i – номер вычисляемого элемента.

**double nextLn**(double x, double pref, int i)

Функция вычисляет следующий член ряда Макларена через предыдущий для математической функции ln(x+1).

* double x – точка, в которой вычисляется значение функции.
* double pref – значение предыдущего члена последовательности.
* int i – номер вычисляемого элемента.

**double absoluteDeviation**(double value, double standard)

Функция вычисляет абсолютную разницу между value и standard.

* double value – некоторое значение.
* double standard – значение с которым происходит сравнение.

**double relativeDeviation**(double value, double standard)

Функция вычисляет относительную разницу между value и standard.

* double value – некоторое значение.
* double standard – значение с которым происходит сравнение.

# Подтверждение корректности

**?**

# Результаты экспериментов

## Синус

## Косинус

## Экспонента

## Логарифм

# Заключение