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**

**«Подсчет sin(x), cos(x), exp(x), ln(x+1)»**

**Выполнил**:

студент группы 3821Б1ПМ2

Головин Р.М.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2021

**Оглавление**

[Постановка задачи 3](#_Toc104405810)

[Методы решения 4](#_Toc104405811)

[Вычисление следующего члена ряда 4](#_Toc104405812)

[Обратное суммирование 4](#_Toc104405813)

[Прямое суммирование 4](#_Toc104405814)

[Попарное суммирование 4](#_Toc104405815)

[Руководство пользователя 6](#_Toc104405816)

[Описание программной реализации 7](#_Toc104405817)

[Подтверждение корректности 10](#_Toc104405818)

[Результаты экспериментов 11](#_Toc104405819)

[Синус 11](#_Toc104405820)

[Косинус 12](#_Toc104405821)

[Экспонента 13](#_Toc104405822)

[Логарифм 14](#_Toc104405823)

[Заключение 15](#_Toc104405824)

# 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.

Метод должен быть наименее точным, поскольку суммируемые числа имеют сильно различающийся порядок.

## 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.

Метод должен быть средним по точности.

## 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.

Метод должен наиболее точным, поскольку суммируемые числа близки по порядку.

# Руководство пользователя

При запуске программы нужно выбрать вычилсяемую функцию,введя цифру от одного до 4.(1-косинус,2-синус,3-экспонента,4-логорифм).После чего на экране сначала выведется измерения для абсолютной ошибки в некотором диапазоне. В каждой строке написано значение икс, ошибка обратного суммирования, ошибка прямого суммирования, ошибка попарного суммирования. Затем в аналогичной форме выведутся результаты измерения относительной ошибки.

# Описание программной реализации

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

Функция задает начало для рекурсивного суммирования.

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

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

Функция выполняет рекурсивное суммирование , начиная с конца ряда.

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

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

Функция выполняет последовательное сложение элементов ряда , начиная с первого.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Функция вычисляет абсолютную разницу между value и standard.

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

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

Функция вычисляет относительную разницу между value и standard.

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

# Подтверждение корректности

Корректность вычислений проверялась за счет сравнения результатов с функциями из стандартной библиотеки math.h, абсолютная и относительная ошибка должны быть малы при тестировании в области определения функций.

# Результаты экспериментов

midRev – средняя ошибка рекурсивного суммирования.

midStr – средняя ошибка прямого суммирования.

midNear – средняя ошибка попарного суммирования.

Таблицы, по которым высчитывались средние значения отклонений, находятся в файле “Tables.xlsx” на листах с соответствующим названием.

## Синус

Таблица1.

|  |  |  |
| --- | --- | --- |
|  | Абсолютная ошибка | Относительная ошибка |
| midRev | 4,40952E-07 | 0,000133356% |
| midStr | 0,00000042 | 0,000141975% |
| midNear | 4,47857E-07 | 0,000133491% |

Как видно из первого столбца таблицы 1 наиболее точным по абсолютной ошибке является обратное суммирование, затем идет попарное и наименее точное прямое суммирование.

Как видно из второго столбца таблицы 1 наиболее точным по относительной ошибке является обратное суммирование, затем идет попарное и наименее точное прямое суммирование.

Делаем вывод, что наиболее точным методом вычисления суммы разложения синуса является обратное суммирование, затем по точности идет попарное и прямое суммирование.

## Косинус

Таблица 2.

|  |  |  |
| --- | --- | --- |
|  | Абсолютная ошибка | Относительная ошибка |
| midRev | 0,000000410 | 0,000142068% |
| midStr | 0,000000457 | 0,000155962% |
| midNear | 0,000000417 | 0,000150299% |

Как видно из первого столбца таблицы 3 наиболее точным по абсолютной ошибке является обратное суммирование, затем идет попарное и наименее точное прямое суммирование.

Как видно из второго столбца таблицы 3 наиболее точным по относительной ошибке является обратное суммирование, затем идет попарное и наименее точное прямое суммирование.

Делаем вывод, что наиболее точным методом вычисления суммы разложения косинуса является обратное суммирование, затем по точности идет попарное и прямое суммирование.

## Экспонента

Таблица 3.

|  |  |  |
| --- | --- | --- |
|  | Абсолютная ошибка | Относительная ошибка |
| midRev | 0,63323177 | 1,02859E-05% |
| midStr | 0,76848058 | 0,00001159% |

Как видно из первого столбца таблицы 3 наиболее точным по абсолютной ошибке является обратное суммирование, затем по точности идет прямое суммирование.

Как видно из второго столбца таблицы 3 наиболее точным по абсолютной ошибке является обратное суммирование, затем по точности идет прямое суммирование.

Делаем вывод, что наиболее точным методом вычисления суммы разложения экспоненты является обратное суммирование.

## Логарифм

Таблица 4.

|  |  |  |
| --- | --- | --- |
|  | Абсолютная ошибка | Относительная ошибка |
| midRev | 0,01032099 | 0,00011259 |
| midStr | 0,01032523 | 0,00011263 |

Как видно из первого столбца таблицы 4 наиболее точным по абсолютной ошибке является обратное суммирование, затем по точности идет прямое суммирование.

Как видно из второго столбца таблицы 4 наиболее точным по абсолютной ошибке является обратное суммирование, затем по точности идет прямое суммирование.

Делаем вывод, что наиболее точным методом вычисления суммы разложения логорифма является обратное суммирование.

# Заключение

В ходе лабораторной работы было реализовано вычисление значения функций sin, cos, exp, ln в точке с помощью разложения в ряд Маклорена. Были реализованы методы прямого, попарного и обратного суммирования элементов разложения. Была описана реализация и алгоритмы работы программы. Проведено подтверждение корректность реализации вычисления данных функций . Проведены эксперименты по замеру точности различных методов суммирования. По результатам экспериментов было подтверждено, что наиболее точным методом вычисления суммы ряда является обратное суммирование.