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**

**«Поиск значений функций при помощи рядов Тейлора»**

**Выполнил**:

Студент группы 3824Б1ПМ1

Сорокин Д.Д.

**Проверила**:

Бусько П.В.

Нижний Новгород

2025

**Содержание**

[Введение 3](#_Toc182419046)

[Постановка задачи 4](#_Toc182419047)

[Руководство пользователя 5](#_Toc182419048)

[Описание программной реализации 7](#_Toc182419049)

[Результаты экспериментов 10](#_Toc182419050)

[Заключение 13](#_Toc182419051)

[Литература 14](#_Toc182419052)

[Приложение 15](#_Toc182419053)

# Введение

В данной лабораторной работе рассматривается задача вычисления значений элементарных математических функций с использованием разложения в ряд Тейлора. Ряд Тейлора позволяет представить функцию в виде бесконечной суммы членов, вычисляемых на основе производных этой функции в заданной точке. В работе реализованы функции для вычисления синуса, косинуса, натурального логарифма и экспоненты с использованием разложения в ряд Тейлора.

Целью работы является изучение принципов работы с рядами Тейлора, а также анализ точности вычислений при различном количестве членов ряда. В процессе выполнения работы будут рассмотрены два метода суммирования членов ряда: прямое суммирование (от первого члена к последнему) и обратное суммирование (от последнего члена к первому). Это позволит оценить влияние порядка суммирования на точность результата.

В работе также проводится сравнение результатов, полученных с использованием рядов Тейлора, со значениями, вычисленными с помощью стандартных функций из библиотеки math.h. Это позволит оценить точность аппроксимации и определить, при каком количестве членов ряда достигается достаточная точность.

# Постановка задачи

В рамках этой лабораторной работы нужно выполнить вычисление функций sin(x), cos(x), ln(1+x) и e^x, применяя разложение в ряд Тейлора. Также необходимо сравнить результаты, полученные с использованием различных методов суммирования, с данными, вычисленными стандартными функциями из библиотеки math.h.

# Руководство пользователя

Пользователь выбирает какую функцию он хочет проверить (Рис.1). Затем выбирает количество элементов ряда (Рис. 2). После он выбирает значение аргумента функции (Рис.2). Программа выводит значение функции, используя разные методы суммирования (до значения, указанного пользователем), а также библиотечное значение. Как только сумма совпадает с библиотечным значением программа выводит, что значение совпало (Рис. 3).

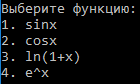


Рис. 1 – Выбор функции



Рис. 2 – Выбор кол-ва элементов и x

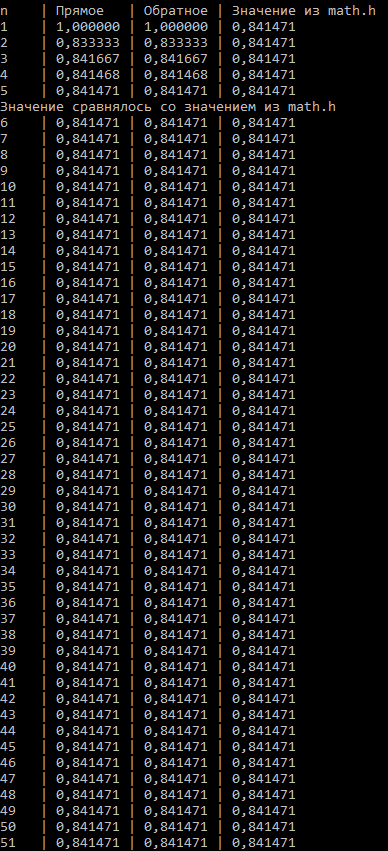


Рис. 3 – Вывод значений

# Описание программной реализации

Вся программа находится в одном файле, но содержит множество функций. Их можно разбить на группы.

1. sinx, cosx, lnx, expx

В них реализованы математические функции через рекурсию (Рис. 4)

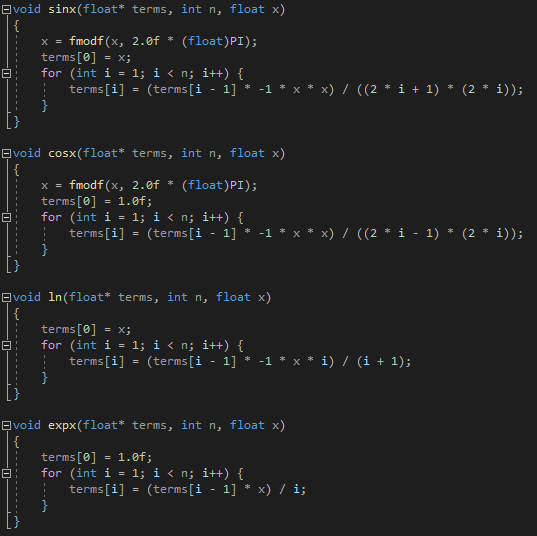


Рис. 4 – Реализация мат. функций

2/ Способы суммирования (direct\_sum, reverse\_sum)

**direct\_sum** (Рис. 5)

Функция вычисляет сумму элементов массива terms в прямом порядке (от первого до последнего элемента).

* 1. Инициализируется переменная fx значением 0.0f, которая будет хранить накапливаемую сумму.
  2. В цикле for перебираются все элементы массива terms от индекса 0 до n-1.
  3. На каждой итерации значение текущего элемента массива terms[i] добавляется к fx.
  4. После завершения цикла функция возвращает значение fx, которое является суммой всех элементов массива.

**reverse\_sum** (Рис. 6)

Функция вычисляет сумму элементов массива terms в обратном порядке (от последнего до первого элемента).

* 1. Инициализируется переменная fx значением 0.0f, которая будет хранить накапливаемую сумму.
  2. В цикле for перебираются все элементы массива terms от индекса n-1 до 0.
  3. На каждой итерации значение текущего элемента массива terms[i] добавляется к fx.
  4. После завершения цикла функция возвращает значение fx, которое является суммой всех элементов массива.

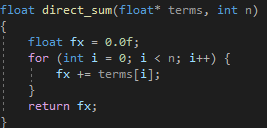


Рис. 5 – Реализация direct\_sum

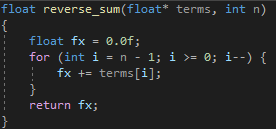


Рис. 6 – Реализация reverse\_sum

3/ filling (Рис. 7)

Функция filling заполняет массив terms значениями, которые зависят от выбранной математической функции (sin, cos, ln, exp), вычисляет ожидаемый результат (expected\_res) для выбранной функции, а также вычисляет сумму элементов массива terms двумя способами: в прямом порядке (direct\_fx) и в обратном порядке (reverse\_fx).

**Логика работы:**

В зависимости от значения func вызывается одна из функций (sinx, cosx, ln, expx), которая заполняет массив terms значениями, соответствующими выбранной математической функции.

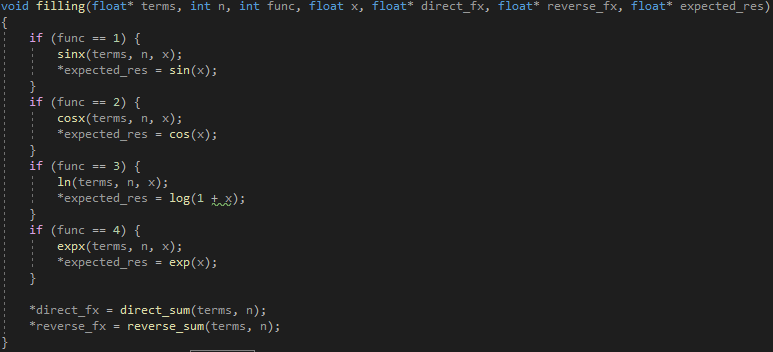


Рис. 7 – Функция filling

4/ printres (Рис. 8)

Функция printres выводит результаты вычислений для различных размеров массива terms (от 1 до n). Она сравнивает результаты, полученные с помощью функций direct\_sum и reverse\_sum, с ожидаемым результатом, взятым из библиотеки math.h. Если результаты совпадают, выводится соответствующее сообщение.

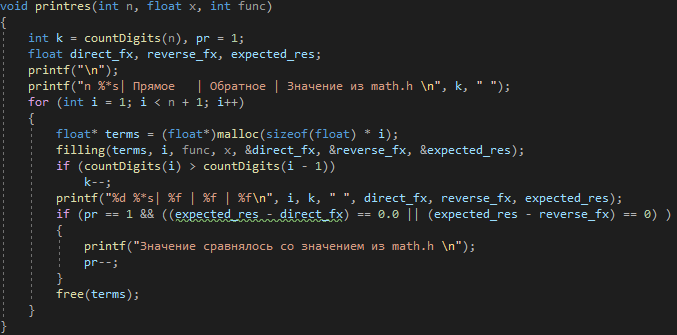


Рис. 8 – Функция printres

# Результаты экспериментов

По данным экспериментов видно, что при увеличении числа слагаемых погрешность уменьшается. С увеличением количества членов ряда (n) точность вычислений возрастает. При достаточно большом n значения, полученные прямым и обратным суммированием, совпадают с эталонными значениями, взятыми из math.h.

Например, x=1. Тогда:

Значение в sin(x) совпало с эталонным при n=5 (Рис. 9)

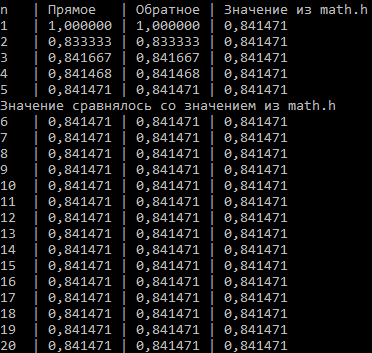


Рис. 9 – sin(x), при x=1

Значение в cos(x) совпало с эталонным при n=6 (Рис. 10)

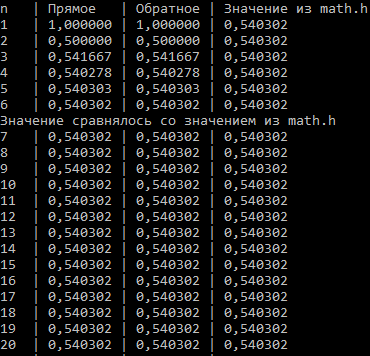


Рис. 10 – cos(x), при x=1

При проверке ln(1+x) при x=1 точности измерения не хватало, чтоб достичь эталонного значения, поэтому я взял x=0,5. Значение совпало с эталонным при n=19 (Рис. 11)

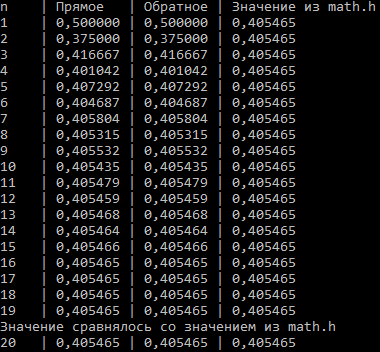


Рис. 11 – ln(1+x), при x=0,5

Значение в e^x совпало с эталонным при n=10 (Рис. 12)

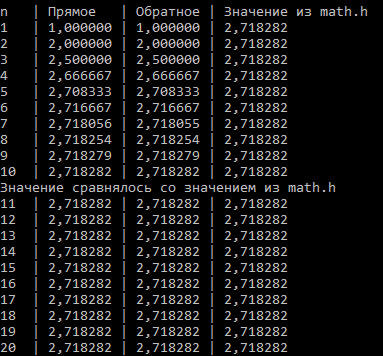


Рис. 12 – e^x, при x=1

# Заключение

В ходе выполнения лабораторной работы были исследованы методы вычисления значений функций (sin(x), cos(x), ln(1+x), e^x) с использованием рядов Тейлора. Основной целью работы было сравнение точности вычислений, полученных прямым и обратным суммированием членов ряда, с эталонными значениями, предоставленными библиотекой math.h.

Основные результаты:

1/ Точность вычислений: с увеличением количества членов ряда (n) точность вычислений возрастает. При достаточно большом n значения, полученные как прямым, так и обратным суммированием, практически совпадают с эталонными значениями. Это подтверждает корректность реализации алгоритмов на основе рядов Тейлора.

2/ Сравнение методов суммирования: прямое и обратное суммирование дают схожие результаты, однако обратное суммирование может быть более устойчивым к накоплению ошибок округления, особенно при больших значениях n. Это делает его предпочтительным в случаях, где требуется высокая точность вычислений.

# Литература

1. Керниган Б., Ритчи Д., Фьюэр А. Язык программирования СИ //М.: Финансы и статистика. – 1992.
2. Кнут Д. Э. Искусство программирования: Сортировка и поиск. – Издательский дом Вильямс, 2000. – Т. 3.

# Приложение

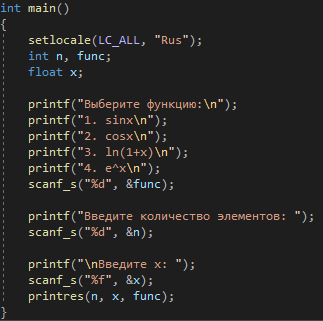


Рис. 12 – Функция main