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**

**«Вычисление математических функций с использованием рядов»**

**Выполнил**:

студент группы 3824Б1ПМ1

Ермольев. Т.Д.

**Проверил**:

преподаватель каф. ВВСП,

Волокитин В.Д.

Нижний Новгород

2025

**Содержание**

[Постановка задачи 3](#_Toc26962562)

[Метод решения 4](#_Toc26962563)

[Руководство пользователя 5](#_Toc26962564)

[Описание программной реализации 6](#_Toc26962565)

[Подтверждение корректности 7](#_Toc26962566)

[Результаты экспериментов 8](#_Toc26962567)

[Заключение 9](#_Toc26962568)

[Приложение 10](#_Toc26962569)

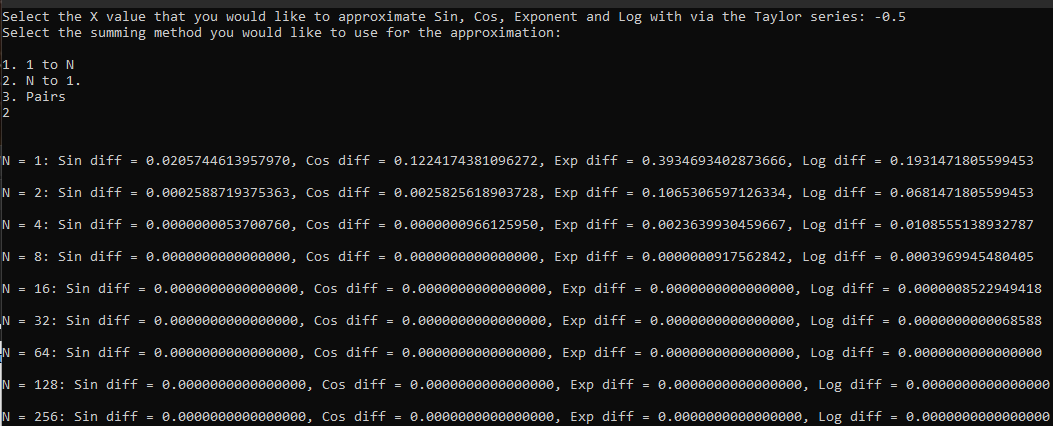
# Постановка задачи

Посчитать ошибку в разложении Маклорена функций , , , , до -го члена для разных . При вычислении значений, сравнить методы суммирования последовательности чисел c плавающей запятой (прямое, обратное, попарное).

# Метод решения

Созданы функции для вычисления -го члена функций , , , с помощью -го члена и . Одна из этих функций передаётся в одну из трёх функций для вычисления и суммирования последовательности, которые возвращает сумму всех членов. Далее выводится модуль разницы между значением и библиотечной функцией.

# Руководство пользователя

Пользователь вводит , и то какой метод суммирования он хочет использовать (прямой, обратный или попарный). Программа выводит модуль разницы между возратным значением функции и библиотечными функциями для

Модуль разницы функции логарифма выводится только в случае

# Описание программной реализации

mathFuncs.cpp, для создания использовался IDE Visual Studio и компилятор MSVC.

- Функции NthTaylorSin, NthTaylorCos, NthTaylorExp, NthTaylorLog для вычисления -го члена последовательности Маклорена функций , , , .

- Функции OneToN, NToOne, PairSum для суммирования последовательности.

- Функция PairSumFunc использованная PairSum для рекурсии.

- Функция main - главная функция программы, которая запрашивает X и метода суммирования последовательности, и которая выводит ошибку

# Подтверждение корректности

Для подтверждения корректности в программе распечатываются результаты библиотечных функций, и разница значения написанной функции с ними.

# Результаты экспериментов

По данным экспериментов видно, что:

- Для значений все функции “сходятся” к одному и тому же значению, и ошибка перестаёт убывать.

Таблица средних ошибок каждого алгоритма при и разных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Обычная | Обратная | Попарная | Вывод |
|  | S – 0, C – 1e-15, E – 2e-15, L – 1e-15 | S – 0, C – 0, E – 0, L – 1e-15 | S – 0, C – 0, E – 0, L – 1e-15 | Обратная и попарная выигрывают |
|  | S – 1e-15, C – 1e-15, E – 3e-15, L – 2e-15 | S – 0, C – 0, E – 0, L – 1e-15 | S – 0, C – 0, E – 1e-15, L – 1e-15 | Обратная выигрывает, попарная лучше обычной |
|  | S – 5e-15, C – 5e-15, E – 5e-14 | S – 7e-15, C – 3e-15, E – 2e-14 | S – 7e-15, C – 6e-15, E – 4e-14 | Обычная для синуса, обратная для остальных. |
|  | S – 3e-13, C – 1e-12, E – 9e-12 | S – 2e-13, C – 5e-13, E – 5e-12 | S – 2e-13, C – 5e-13, E – 7e-12 | Обратная выигрывает, попарная лучше обычной |
|  | S – 3e-11, C – 2e-11, E – 1e-9 | S – 5e-11, C – 6e-11, E – 3e-9 | S – 4e-11, C – 7e-11, E – 4e-9 | Обычная выигрывает для всех функций, попарная лучше обратной для синуса и косинуса |
|  | S – 3e-8, C – 5e-9, E –9e-7 | S – 5e-9, C – 5e-10, E –2e-7 | S – 7e-8, C – 4e-8, E –1e-6 | Обратная выигрывает, обычная лучше попарной |
|  | S – 4e-5, C – 8e-5, E –6e-3 | S – 3e-6, C – 9e-6, E –4e-3 | S – 8e-6, C – 1e-5, E –5e-3 | Обратная выигрывает, попарная лучше обычной |

Вывод: Чаще всего обратный алгоритм суммирования выдаёт самый точный результат. В некоторых случаях лучше использовать обычный, в частности для синуса. Попарная является хорошим вариантом лежащим между двумя остальными алгоритмами.

# Заключение

Была написана программа считающая ошибку в разложении Маклорена функций , , , , до -го члена для разных . При вычислении значений, были сравнены методы суммирования последовательности чисел c плавающей запятой (прямое, обратное, попарное).

# Приложение

#include "stdio.h"

#include "stdlib.h"

#include "math.h"

double NthTaylorSin(double x, double prev, int n)

{

return n == 0 ? x : (-(prev \* x \* x) / ((2 \* n)\*(2 \* n + 1)));

}

double NthTaylorCos(double x, double prev, int n)

{

return n == 0 ? 1 : (-(prev \* x \* x) / ((2 \* n) \* (2 \* n -1)));

}

double NthTaylorExp(double x, double prev, int n)

{

return n == 0 ? 1 : prev\*x/n;

}

double NthTaylorLog(double x, double prev, int n)

{

return n == 0 ? x : -prev\*x\*n/(n+1);

}

double OneToN(int n, double T(double, double, int), double x )

{

double ret = 0, val = 0;

int i;

for (i = 0; i < n; i++)

{

val = T(x, val, i);

ret += val;

}

return ret;

}

double NToOne(int n, double T(double, double, int), double x)

{

double ret = 0;

int i;

double\* arr = (double\*)malloc(n \* sizeof(double));

arr[0] = T(x, 0, 0);

for (i = 1; i < n; i++)

{

arr[i] = T(x, arr[i - 1], i);

}

for (i = n - 1; i >= 0; i--)

{

ret += arr[i];

}

free(arr);

return ret;

}

double PairSumFunc(double\* arr, int l, int r)

{

double ret = 0;

int i;

if (r - l <= 5)

{

for (i = r-1; i >= l; i--) ret += arr[i];

return ret;

}

return PairSumFunc(arr, l, (l + r) / 2) + PairSumFunc(arr, (l + r) / 2, r);

}

double PairSum(int n, double T(double, double, int), double x)

{

int i;

double ret = 0;

double\* arr = (double\*)malloc(n \* sizeof(double));

arr[0] = T(x, 0, 0);

for (i = 1; i < n; i++)

{

arr[i] = T(x, arr[i - 1], i);

}

ret = PairSumFunc(arr, 0, n);

free(arr);

return ret;

}