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)»**

**Выполнила**:

студентка группы 3821Б1ПМ2

Анисимова Ю. Д.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2022

**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 11](#_Toc26962568)

[Приложение 12](#_Toc26962569)

# Постановка задачи

Задачей моей лабораторной работы является реализация вычисления значения функций sin, cos, exp, ln(1+x) в точке с помощью ряда Маклорена. Необходимо реализовать методы прямого, попарного и обратного суммирования элементов. Нужно описать реализацию и алгоритмы работы программы. Необходимо подтвердить корректность реализации вычисления данных функций. Провести эксперименты по замеру точности различных методов суммирования и результаты сравнить с библиотечным значением.

# Метод решения

Ряд Тейлора – разложение функции в бесконечную сумму степенных функций. С помощью этого ряда можно найти приближённое значение функции в окрестности заданной точки.

Разложения по формуле Тейлора для заданных функций выглядят так:

Задача выполняется с помощью прямого, обратного и попарного суммирования.

**Прямое суммирование**

Слагаемые складываются в порядке увеличения их порядкового номера (каждое слагаемое находится с использованием предыдущего)

**Обратное суммирование**

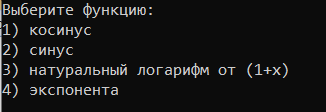
Слагаемые складываются в порядке уменьшения их порядкового номера (каждое слагаемое находится с использованием предыдущего)

**Попарное суммирование**

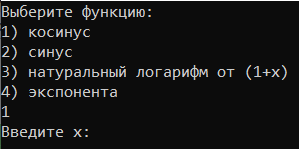
Попарное суммирование представляет собой метод суммирования последовательности чисел конечной точности с плавающей запятой. Попарное суммирование последовательности из n чисел x n работает посредством рекурсивно разбиения последовательности на две половины, суммирования каждой половины и сложения двух.

# Руководство пользователя

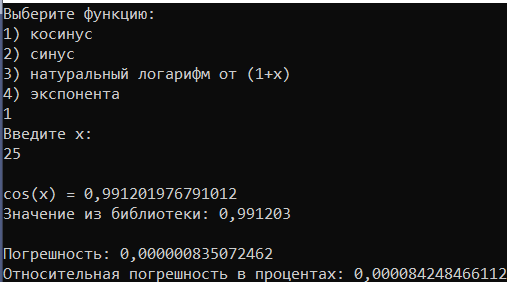
При запуске программы появляется окно ввода, где предлагается выбрать функцию для подсчета.



Далее предлагается ввести число x, от которого будет считаться выбранная функция:



В результате работы программы будет выводится значение выбранной функции, которое посчитала программа, значение этой же функции из библиотеки math.h, погрешность вычислений и относительная погрешность, выраженная в %.



# Описание программной реализации

double direct\_summ(double\* arr, int m)

Функция прямого суммирования. Принимает на вход массив слагаемых и длину.

double revers\_summ(double\* arr, int m)

Функция обратного суммирования. Принимает на вход массив слагаемых и длину.

double paired\_summ(double\* arr, int m)

Функция попарного суммирования. Принимает на вход массив слагаемых и длину.

double cos\_1(double pred, double n, int i)

Функция вычисления косинуса.

double sin\_1(double pred, double n, int i)

Функция вычисления синуса.

double ln\_1(double pred, double n, int i)

Функция вычисления натурального логарифма от 1+x

double exp\_1(double pred, double n, int i)

Функция вычисления экспоненты.

void made\_array(double\* arr, int m, option result, double p)

# Подтверждение корректности

Для подтверждения корректности в программе вычисляется относительная и абсолютная погрешности.

Абсолютная погрешность вычисляет разницу между библиотечным значением и значением, которое посчитала программа. Можно наглядно увидеть, насколько точно работает программа.

Относительная погрешность вычисляется в процентах. Абсолютная погрешность делится на значение, которое посчитала программа и все это умножается на 100.

# Результаты экспериментов

**Прямое суммирование:**

SIN(X)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 1 | 5 | 20 | 30 | 40 | 50 |
| Абс. погрешность | **0** | **-0,000000000000002** | -0,000000002195830 | -0,000058433734858 | 2,560623082529272 | 39110,196105539434939 |
| Относит. погрешность | **0** | **0,000000000000208** | -0,000000240521526 | 0,005914506125074 | -141,041536123252286 | **-99,999329144009309** |

COS(X)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 1 | 5 | 20 | 30 | 40 | 50 |
| Абс. погрешность | **0** | 0,000000000000002 | **-0,000000002508622** | 0,000025759462948 | 0,290050237001436 | 29641,527992883478873 |
| Относит. погрешность | **0** | 0,000000000000568 | **-0,000000614734736** | 0,016702446185697 | -30,308650315733420 | -100,003255559037868 |

EXP(X)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 1 | 5 | 20 | 30 | 40 | 50 |
| Абс. погрешность | **0** | 0,000000000000057 | 0,000000059604645 | **-0,003906250000000** | **288,0** | **1659137687552,0** |
| Относит. погрешность | **0** | 0,000000000000038 | 0,000000000000012 | **-0,000000000000037** | **0,000000000000122** | **0,000000032000616** |

LN(1+X) -1<x<=1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | -0,9 | -0,5 | 0 | 0,5 | 0,9 | 1,0 |
| Абс. погрешность | **-0,000002187354984** | **0** | **0** | **0** | 0,000000125155339 | **0,004975001249750** |
| Относит. погрешность | 0,000094995710185 | 0,000000000000032 | **0** | 0,000000000000027 | 0,000019499042725 | **0,722929725920748** |

**Обратное суммирование:**

SIN(X)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 1 | 5 | 20 | 30 | 40 | 50 |
| Абс. погрешность | **0** | **-0,000000000000002** | -0,000000001317719 | -0,000054106324121 | **2,461328157629367** | 24025,127005790214753 |
| Относит. погрешность | **0** | 0,000000000000220 | -0,000000144337162 | **0,005476473214023** | **-143,416073261024934** | -99,998907926737218 |

COS(X)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 1 | 5 | 20 | 30 | 40 | 50 |
| Абс. погрешность | **0** | **0,000000000000001** | -0,000000001972122 | **0,000021037791799** | 0,542250115903654 | 22526,040604811976664 |
| Относит. погрешность | **0** | **0,000000000000489** | -0,000000483266052 | **0,013640495096285** | **-44,844146342853158** | -100,004283963543415 |

EXP(X)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 1 | 5 | 20 | 30 | 40 | 50 |
| Абс. погрешность | **0** | **0** | **-0,000000119209290** | 0,003906250000000 | 352,0 | 1659139784704,0 |
| Относит. погрешность | **0** | **0** | **-0,000000000000025** | 0,000000000000037 | 0,000000000000150 | 0,000000032000656 |

LN(1+X) -1<x<=1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | -0,9 | -0,5 | 0 | 0,5 | 0,9 | 1,0 |
| Абс. погрешность | -0,000002187354985 | **0** | **0** | **0** | **0,000000125155338** | **0,004975001249750** |
| Относит. погрешность | 0,000094995710243 | **0** | **0** | **0** | **0,000019499042604** | 0,722929725920764 |

**Попарное суммирование:**

SIN(X)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 1 | 5 | 20 | 30 | 40 | 50 |
| Абс. погрешность | **0** | **-0,000000000000002** | **-0,000000003663772** | **-0,000068997400838** | 2,506831910479349 | **20479,737625146295613** |
| Относит. погрешность | **0** | 0,000000000000243 | **-0,000000401313386** | 0,006983806773014 | -142,294671636965262 | -99,998718872784636 |

COS(X)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 1 | 5 | 20 | 30 | 40 | 50 |
| Абс. погрешность | **0** | **0,000000000000001** | -0,000000001495434 | 0,000054760603876 | **0,055718188347738** | **20480,964966028492199** |
| Относит. погрешность | 0,000000000000021 | **0,000000000000489** | -0,000000366454234 | 0,035513475762882 | -7,710192549741059 | **-100,004711748185997** |

EXP(X)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 1 | 5 | 20 | 30 | 40 | 50 |
| Абс. погрешность | **0** | 0,000000000000028 | 0,000002026557922 | 497381,77148437500 | 67453479028576,0 | 165098868494293794816,0 |
| Относит. погрешность | 0,000000000000016 | 0,000000000000019 | 0,000000000000418 | 0,000004654311306 | 0,028664841843468 | 3,289079795970353 |

LN(1+X) -1<x<=1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | -0,9 | -0,5 | 0 | 0,5 | 0,9 | 1,0 |
| Абс. погрешность | **-0,000145367772748** | **0** | **0** | **0** | 0,000008654593361 | 0,007751472290696 |
| Относит. погрешность | 0,006313640750536 | **0** | **0** | **0** | 0,001348392561779 | 1,130948472127189 |

Таким образом, глядя на полученные результаты и сравнивая их между собой и со значениями из библиотеки, можно сделать вывод, что для вычисления синуса больше подходит попарное суммирование, для вычисления косинуса больше подходит обратное суммирование, для вычисления экспоненты прямое суммирование, для вычисления логарифма обратное суммирование.

# Заключение

В ходе лабораторной работы я реализовала вычисления значений функций sin, cos, exp, ln(1+x) в точке с помощью ряда Маклорена. Реализовала методы прямого, попарного и обратного суммирования элементов. Описала реализацию и алгоритмы работы программы. Подтвердила корректность реализации вычисления данных функций. Провела эксперименты по замеру точности различных методов суммирования и результаты сравнила с библиотечным значением.

# Приложение

double direct\_summ(double\* arr, int m)

{

double summ = 0;

int i;

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

{

summ += arr[i];

}

return summ;

}

double revers\_summ(double\* arr, int m)

{

double summ = 0;

int i;

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

{

summ += arr[m - i];

}

return summ;

}

double paired\_summ(double\* arr, int m)

{

int i, j;

while (m > 1)

{

for (i = 1, j = 0; i < m; i += 2, ++j)

{

arr[j] = arr[i] + arr[i - 1];

}

m /= 2;

}

return arr[0];

}

double cos\_1(double pred, double n, int i)

{

i = 2 \* i;

return (pred \* (-1.0) \* n \* n) / (i \* (i - 1.0));

}

double sin\_1(double pred, double n, int i)

{

i = 2 \* i + 1;

return (pred \* (-1.0) \* n \* n) / (i \* (i - 1.0));

}

double ln\_1(double pred, double n, int i)

{

return (pred \* (-1.0) \* n \* i) / (i + 1.0);

}

double exp\_1(double pred, double n, int i)

{

return (pred \* n) / ((double)i);

}

void made\_array(double\* arr, int m, option result, double p)

{

int i;

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

{

arr[i] = result(arr[i - 1], p, i);

}

}