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 функций sin(x), cos(x), exp^(x), ln(1+x) в окрестности 0»**

**Выполнил**:

студент/ка группы 3821Б1ПМ2

Василевский А.П.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2022

**Содержание**

[Постановка задачи 3](#_Toc99227367)

[Метод решения 4](#_Toc99227368)

[Руководство пользователя 6](#_Toc99227369)

[Описание программной реализации 8](#_Toc99227370)

[Подтверждение корректности 11](#_Toc99227371)

[Результаты экспериментов 12](#_Toc99227372)

[Заключение 17](#_Toc99227373)

# Постановка задачи

Задачами лабораторной работы являются реализация алгоритмов вычисления функций sin(x), cos(x), exp^(x), ln(1+x) в окрестности 0, суммирование в прямом и обратном порядке, а также попарное суммирование на языке программирования Си, измерение абсолютной и относительной ошибки вычислений.

# Метод решения

**Сортировка выбором**

Суть всех алгоритмов заключается в вычислении приблизительного значения функции при помощи разложения её в ряд Маклорена.

Функцию , имеющую (n+1) производных в точке x = 0, можно представить по формуле Маклорена:

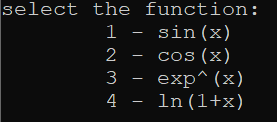
Ряд Маклорена для функций sin(x), cos(x), exp^(x), ln(1+x):

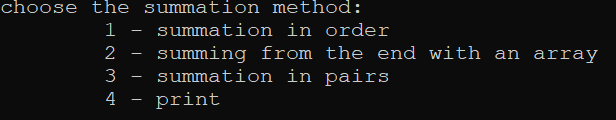
Точность вычисления этих функций повышается с увеличением количества элементов в ряде Маклорена, однако в знаменателе элементов присутствует факториал, которой быстро возрастает и поэтому может принимать значения больше чем может хранить float даже при небольших х, из-за этого при вычислении значения элементов по формуле может возникнуть большая погрешность.

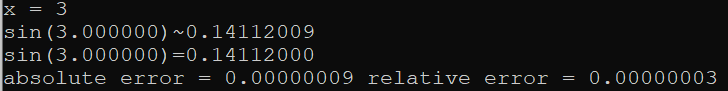
Для того чтобы появилась возможность вычислить как можно больше элементов с сохранением наибольшей точности прибегнем к следующему методу, который рассмотрим на примере функции exp^(x).

Первый элемент равняется 1, а каждый последующий можно получить из предыдущего для этого достаточно поделить предыдущий элемент на порядковый номер текущего и умножить на значение х. За счёт такого способа вычисления элементов удаётся избежать переполнения факториалом переменной типа float. Однако точность вычислений этим способом зависит также и от порядка суммирования, например, при суммировании по порядку по рядку точность будет меньше чем при суммировании с конца.

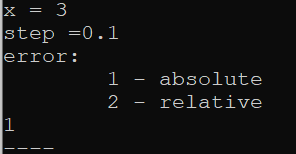
# Руководство пользователя

При запуске программы будет выводиться сообщение с предложением выбрать функцию, значение которой нужно вычислить. 

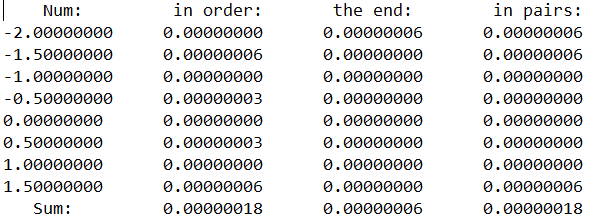
После выбора функции необходимо выбрать способ суммирования (1 – прямое суммирование, 2 – суммирование с конца, 3 – попарное суммирование, 4 – суммирование всеми способами на интервале с записью полученных значений в файл)

При выборе способа суммирования от 1 до 3 программа запросит число х и выведет значение функции от этого числа вычисленное при помощи разложения в ряд Маклорена и значение библиотечной функции. Далее выведет абсолютную и относительную погрешность.

При выборе способа суммирования 4 программа запросит помимо самого числа х ещё и шаг, с которым будет вычисляться ошибка. Далее нужно будет выбрать ошибку, которую нужно посчитать.



После чего в файл “table.txt” (при отсутствии будет создан автоматически) будут записаны ошибки выбранного типа на интервале от -|х| до |х| с шагом |step| и записаны в следующем формате: в первой столбце записано число от которого вычисляется функция, во втором столбце ошибка прямого суммирования, в третьем столбце ошибка обратного суммирования, в четвёртом столбце ошибка попарного суммирования. В последней строке будет записана суммарная ошибка каждого из столбцов.



# Описание программной реализации

void sinchoose() – вызывается при выборе вычисления . Предоставляет выбор числа и способа суммирования на основе введённых данных вызывает функцию для подсчёта и выводит приблизительное значение, значение библиотечной функции и ошибку (абсолютную и относительную).

void coschoose() – вызывается при выборе вычисления . Предоставляет выбор числа и способа суммирования на основе введённых данных вызывает функцию для подсчёта и выводит приблизительное значение, значение библиотечной функции и ошибку (абсолютную и относительную).

void expchoose() – вызывается при выборе вычисления . Предоставляет выбор числа и способа суммирования на основе введённых данных вызывает функцию для подсчёта и выводит приблизительное значение, значение библиотечной функции и ошибку (абсолютную и относительную).

void lnchoose() – вызывается при выборе вычисления . Предоставляет выбор числа и способа суммирования на основе введённых данных вызывает функцию для подсчёта и выводит приблизительное значение, значение библиотечной функции и ошибку (абсолютную и относительную).

float \_1sin(float num) - на вход функция принимает число типа float и возвращает примерное значение синуса от этого числа используя метод прямого суммирования.

float \_2sin(float num) - на вход функция принимает число типа float и возвращает примерное значение синуса от этого числа используя метод суммирования с конца при помощи массива.

float \_3sin(float num) - на вход функция принимает число типа float и возвращает примерное значение синуса от этого числа используя метод попарного суммирования.

float \_1cos(float num) - на вход функция принимает число типа float и возвращает примерное значение косинуса от этого числа используя метод прямого суммирования.

float \_2cos(float num) - на вход функция принимает число типа float и возвращает примерное значение косинуса от этого числа используя метод суммирования с конца при помощи массива.

float \_3cos(float num) - на вход функция принимает число типа float и возвращает примерное значение косинуса от этого числа используя метод попарного суммирования.

float \_1exp(float num) - на вход функция принимает число типа float и возвращает примерное значение от этого числа используя метод прямого суммирования.

float \_2exp(float num) - на вход функция принимает число типа float и возвращает примерное значение от этого числа используя метод суммирования с конца при помощи массива.

float \_3exp(float num) - на вход функция принимает число типа float и возвращает примерное значение от этого числа используя метод попарного суммирования.

float \_1ln(float num) - на вход функция принимает число типа float и возвращает примерное значение ln(1+num) от этого числа используя метод прямого суммирования.

float \_2ln(float num) - на вход функция принимает число типа float и возвращает примерное значение ln(1+num) от этого числа используя метод суммирования с конца при помощи массива.

float \_3ln(float num) - на вход функция принимает число типа float и возвращает примерное значение ln(1+num) от этого числа используя метод попарного суммирования.

float next(float pred, int i, float x) – на вход получает предыдущий член в ряде Маклорена номер текущего и число для которого вычисляется функция. Возвращает следующий член в ряде для функций ln(1+num) и .

float next\_cos(float pred, int i, float x) – на вход получает предыдущий член в ряде Маклорена номер текущего и число для которого вычисляется функция. Возвращает следующий член в ряде для функции cos(x).

float next\_sin(float pred, int i, float x) – на вход получает предыдущий член в ряде Маклорена номер текущего и число для которого вычисляется функция. Возвращает следующий член в ряде для функции sin(x).

# Подтверждение корректности

Для подтверждения корректности во время работы программы при выборе метода суммирования выберем четвёртый вариант, введём “x” и “step”, а также выберем интересующую нас ошибку. Программой будет создан файл в котором будет записана ошибка для трёх способов суммирования на интервале от -|x| до |x| с шагом в step, а в последней строчке будет записана суммарная погрешность для каждого из способов суммирования. На основе суммарной погрешности можно сделать вывод о точности вычислений.

Абсолютная ошибка вычисляется как модуль разности между значением реализованной функции и её библиотечным аналогом, корректность которого подтверждена. Относительная вычисляется как абсолютная делённая на число. Поэтому при х = 0 относительная ошибка в программе принимает значение “nan”

# Результаты экспериментов

Для каждой функции построена таблица, в которой каждому значению числа х соответствует ошибка вычислений для каждого типа суммирования.

Абсолютная ошибка для sin(x):

|  |  |  |  |
| --- | --- | --- | --- |
| Х | Прямое суммирование | Суммирование с конца | Попарное суммирование |
| -3.00000000 | 0.00000009 | 0.00000004 | 0.00000004 |
| -2.70000005 | 0.00000024 | 0.00000024 | 0.00000000 |
| -2.40000010 | 0.00000006 | 0.00000006 | 0.00000006 |
| -2.10000014 | 0.00000006 | 0.00000006 | 0.00000006 |
| -1.80000019 | 0.00000006 | 0.00000006 | 0.00000000 |
| -1.50000024 | 0.00000006 | 0.00000000 | 0.00000006 |
| -1.20000029 | 0.00000000 | 0.00000000 | 0.00000000 |
| -0.90000027 | 0.00000006 | 0.00000000 | 0.00000000 |
| -0.60000026 | 0.00000000 | 0.00000000 | 0.00000000 |
| -0.30000025 | 0.00000000 | 0.00000000 | 0.00000000 |
| -0.00000024 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.29999977 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.59999979 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.89999980 | 0.00000000 | 0.00000000 | 0.00000000 |
| 1.19999981 | 0.00000000 | 0.00000000 | 0.00000000 |
| 1.49999976 | 0.00000006 | 0.00000000 | 0.00000000 |
| 1.79999971 | 0.00000000 | 0.00000006 | 0.00000012 |
| 2.09999967 | 0.00000012 | 0.00000018 | 0.00000018 |
| 2.39999962 | 0.00000000 | 0.00000000 | 0.00000012 |
| 2.69999957 | 0.00000003 | 0.00000006 | 0.00000006 |
| 2.99999952 | 0.00000007 | 0.00000004 | 0.00000004 |
| Сумма | 0.00000091 | 0.00000080 | 0.00000075 |

Относительная ошибка для sin(x):

|  |  |  |  |
| --- | --- | --- | --- |
| Х | Прямое суммирование | Суммирование с конца | Попарное суммирование |
| -3.00000000 | 0.00000003 | 0.00000001 | 0.00000001 |
| -2.70000005 | 0.00000009 | 0.00000009 | 0.00000000 |
| -2.40000010 | 0.00000002 | 0.00000002 | 0.00000002 |
| -2.10000014 | 0.00000003 | 0.00000003 | 0.00000003 |
| -1.80000019 | 0.00000003 | 0.00000003 | 0.00000000 |
| -1.50000024 | 0.00000004 | 0.00000000 | 0.00000004 |
| -1.20000029 | 0.00000000 | 0.00000000 | 0.00000000 |
| -0.90000027 | 0.00000007 | 0.00000000 | 0.00000000 |
| -0.60000026 | 0.00000000 | 0.00000000 | 0.00000000 |
| -0.30000025 | 0.00000000 | 0.00000000 | 0.00000000 |
| -0.00000024 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.29999977 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.59999979 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.89999980 | 0.00000000 | 0.00000000 | 0.00000000 |
| 1.19999981 | 0.00000000 | 0.00000000 | 0.00000000 |
| 1.49999976 | 0.00000004 | 0.00000000 | 0.00000000 |
| 1.79999971 | 0.00000000 | 0.00000003 | 0.00000007 |
| 2.09999967 | 0.00000006 | 0.00000009 | 0.00000009 |
| 2.39999962 | 0.00000000 | 0.00000000 | 0.00000005 |
| 2.69999957 | 0.00000001 | 0.00000002 | 0.00000002 |
| 2.99999952 | 0.00000002 | 0.00000001 | 0.00000001 |
| Сумма | 0.00000044 | 0.00000034 | 0.00000035 |

Абсолютная ошибка для cos(x):

|  |  |  |  |
| --- | --- | --- | --- |
| Num: | in order: | the end: | in pairs: |
| -4.00000000 | 0.00000024 | 0.00000000 | 0.00000012 |
| -3.59999990 | 0.00000012 | 0.00000036 | 0.00000012 |
| -3.19999981 | 0.00000000 | 0.00000006 | 0.00000018 |
| -2.79999971 | 0.00000000 | 0.00000012 | 0.00000000 |
| -2.39999962 | 0.00000012 | 0.00000018 | 0.00000018 |
| -1.99999964 | 0.00000003 | 0.00000000 | 0.00000012 |
| -1.59999967 | 0.00000004 | 0.00000005 | 0.00000005 |
| -1.19999969 | 0.00000003 | 0.00000006 | 0.00000006 |
| -0.79999971 | 0.00000006 | 0.00000000 | 0.00000000 |
| -0.39999971 | 0.00000006 | 0.00000000 | 0.00000000 |
| 0.00000030 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.40000030 | 0.00000006 | 0.00000000 | 0.00000000 |
| 0.80000031 | 0.00000006 | 0.00000000 | 0.00000000 |
| 1.20000029 | 0.00000003 | 0.00000000 | 0.00000006 |
| 1.60000026 | 0.00000003 | 0.00000005 | 0.00000007 |
| 2.00000024 | 0.00000000 | 0.00000006 | 0.00000006 |
| 2.40000033 | 0.00000006 | 0.00000006 | 0.00000006 |
| 2.80000043 | 0.00000006 | 0.00000000 | 0.00000012 |
| 3.20000052 | 0.00000006 | 0.00000006 | 0.00000018 |
| 3.60000062 | 0.00000006 | 0.00000006 | 0.00000006 |
| Sum: | 0.00000111 | 0.00000112 | 0.00000143 |

Относительная ошибка для cos(x):

|  |  |  |  |
| --- | --- | --- | --- |
| Num: | in order: | the end: | in pairs: |
| -4.00000000 | 0.00000006 | 0.00000000 | 0.00000003 |
| -3.59999990 | 0.00000003 | 0.00000010 | 0.00000003 |
| -3.19999981 | 0.00000000 | 0.00000002 | 0.00000006 |
| -2.79999971 | 0.00000000 | 0.00000004 | 0.00000000 |
| -2.39999962 | 0.00000005 | 0.00000007 | 0.00000007 |
| -1.99999964 | 0.00000001 | 0.00000000 | 0.00000006 |
| -1.59999967 | 0.00000002 | 0.00000003 | 0.00000003 |
| -1.19999969 | 0.00000002 | 0.00000005 | 0.00000005 |
| -0.79999971 | 0.00000007 | 0.00000000 | 0.00000000 |
| -0.39999971 | 0.00000015 | 0.00000000 | 0.00000000 |
| 0.00000030 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.40000030 | 0.00000015 | 0.00000000 | 0.00000000 |
| 0.80000031 | 0.00000007 | 0.00000000 | 0.00000000 |
| 1.20000029 | 0.00000002 | 0.00000000 | 0.00000005 |
| 1.60000026 | 0.00000002 | 0.00000003 | 0.00000004 |
| 2.00000024 | 0.00000000 | 0.00000003 | 0.00000003 |
| 2.40000033 | 0.00000002 | 0.00000002 | 0.00000002 |
| 2.80000043 | 0.00000002 | 0.00000000 | 0.00000004 |
| 3.20000052 | 0.00000002 | 0.00000002 | 0.00000006 |
| 3.60000062 | 0.00000002 | 0.00000002 | 0.00000002 |
| Sum: | 0.00000077 | 0.00000044 | 0.00000060 |

Абсолютная ошибка для :

|  |  |  |  |
| --- | --- | --- | --- |
| Num: | in order: | the end: | in pairs: |
| -3.00000000 | 0.00000006 | 0.00000002 | 0.00000008 |
| -2.70000005 | 0.00000002 | 0.00000004 | 0.00000004 |
| -2.40000010 | 0.00000003 | 0.00000003 | 0.00000009 |
| -2.10000014 | 0.00000004 | 0.00000002 | 0.00000002 |
| -1.80000019 | 0.00000006 | 0.00000003 | 0.00000003 |
| -1.50000024 | 0.00000001 | 0.00000000 | 0.00000000 |
| -1.20000029 | 0.00000006 | 0.00000000 | 0.00000006 |
| -0.90000027 | 0.00000003 | 0.00000000 | 0.00000006 |
| -0.60000026 | 0.00000000 | 0.00000000 | 0.00000000 |
| -0.30000025 | 0.00000006 | 0.00000000 | 0.00000000 |
| -0.00000024 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.29999977 | 0.00000012 | 0.00000000 | 0.00000000 |
| 0.59999979 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.89999980 | 0.00000000 | 0.00000000 | 0.00000000 |
| 1.19999981 | 0.00000024 | 0.00000000 | 0.00000024 |
| 1.49999976 | 0.00000000 | 0.00000000 | 0.00000000 |
| 1.79999971 | 0.00000048 | 0.00000000 | 0.00000048 |
| 2.09999967 | 0.00000095 | 0.00000095 | 0.00000095 |
| 2.39999962 | 0.00000191 | 0.00000000 | 0.00000095 |
| 2.69999957 | 0.00000095 | 0.00000000 | 0.00000000 |
| 2.99999952 | 0.00000000 | 0.00000000 | 0.00000191 |
| Sum: | 0.00000502 | 0.00000110 | 0.00000491 |

Относительная ошибка для :

|  |  |  |  |
| --- | --- | --- | --- |
| Num: | in order: | the end: | in pairs: |
| -3.00000000 | 0.00000002 | 0.00000001 | 0.00000003 |
| -2.70000005 | 0.00000001 | 0.00000001 | 0.00000001 |
| -2.40000010 | 0.00000001 | 0.00000001 | 0.00000004 |
| -2.10000014 | 0.00000002 | 0.00000001 | 0.00000001 |
| -1.80000019 | 0.00000003 | 0.00000002 | 0.00000002 |
| -1.50000024 | 0.00000001 | 0.00000000 | 0.00000000 |
| -1.20000029 | 0.00000005 | 0.00000000 | 0.00000005 |
| -0.90000027 | 0.00000003 | 0.00000000 | 0.00000007 |
| -0.60000026 | 0.00000000 | 0.00000000 | 0.00000000 |
| -0.30000025 | 0.00000020 | 0.00000000 | 0.00000000 |
| -0.00000024 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.29999977 | 0.00000040 | 0.00000000 | 0.00000000 |
| 0.59999979 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.89999980 | 0.00000000 | 0.00000000 | 0.00000000 |
| 1.19999981 | 0.00000020 | 0.00000000 | 0.00000020 |
| 1.49999976 | 0.00000000 | 0.00000000 | 0.00000000 |
| 1.79999971 | 0.00000026 | 0.00000000 | 0.00000026 |
| 2.09999967 | 0.00000045 | 0.00000045 | 0.00000045 |
| 2.39999962 | 0.00000079 | 0.00000000 | 0.00000040 |
| 2.69999957 | 0.00000035 | 0.00000000 | 0.00000000 |
| 2.99999952 | 0.00000000 | 0.00000000 | 0.00000064 |
| Sum: | 0.00000284 | 0.00000051 | 0.00000217 |

Абсолютная ошибка для ln(1+num):

|  |  |  |  |
| --- | --- | --- | --- |
| Num: | in order: | the end: | in pairs: |
| -0.89999998 | 0.00000167 | 0.00000024 | 0.00000000 |
| -0.76999998 | 0.00000036 | 0.00000000 | 0.00000036 |
| -0.63999999 | 0.00000000 | 0.00000012 | 0.00000000 |
| -0.50999999 | 0.00000000 | 0.00000006 | 0.00000000 |
| -0.38000000 | 0.00000000 | 0.00000000 | 0.00000000 |
| -0.25000000 | 0.00000000 | 0.00000003 | 0.00000000 |
| -0.12000000 | 0.00000001 | 0.00000000 | 0.00000000 |
| 0.00999999 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.13999999 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.26999998 | 0.00000001 | 0.00000000 | 0.00000000 |
| 0.39999998 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.52999997 | 0.00000003 | 0.00000000 | 0.00000006 |
| 0.65999997 | 0.00000012 | 0.00000000 | 0.00000006 |
| 0.78999996 | 0.00000024 | 0.00000000 | 0.00000006 |
| Sum: | 0.00000244 | 0.00000045 | 0.00000054 |

Относительная ошибка для ln(1+num):

|  |  |  |  |
| --- | --- | --- | --- |
| Num: | in order: | the end: | in pairs: |
| -0.89999998 | 0.00000185 | 0.00000026 | 0.00000000 |
| -0.76999998 | 0.00000046 | 0.00000000 | 0.00000046 |
| -0.63999999 | 0.00000000 | 0.00000019 | 0.00000000 |
| -0.50999999 | 0.00000000 | 0.00000012 | 0.00000000 |
| -0.38000000 | 0.00000000 | 0.00000000 | 0.00000000 |
| -0.25000000 | 0.00000000 | 0.00000012 | 0.00000000 |
| -0.12000000 | 0.00000012 | 0.00000000 | 0.00000000 |
| 0.00999999 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.13999999 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.26999998 | 0.00000006 | 0.00000000 | 0.00000000 |
| 0.39999998 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.52999997 | 0.00000006 | 0.00000000 | 0.00000011 |
| 0.65999997 | 0.00000018 | 0.00000000 | 0.00000009 |
| 0.78999996 | 0.00000030 | 0.00000000 | 0.00000008 |
| Sum: | 0.00000304 | 0.00000069 | 0.00000074 |

По данным предоставленным в таблицах относительной ошибки можно сделать следующий вывод: наибольшей точностью обладает обратное суммирование, после него идет попарное суммирование, а самой худшей точностью обладает прямое суммирование. Причём для exp^x разница между попарным и прямым суммированием мала по сравнению с остальными функциями. Это вызвано тем что разложения в ряд Маклорена для sin(x), cos(x), ln(1+x) знакопеременные, в то время как для exp^(x) знакопостоянное.

# Заключение

В ходе лабораторной работы были реализованы алгоритмы вычисления функций sin(x), cos(x), exp^(x), ln(1+x) в окрестности 0 по ряду Маклорена с использованием прямого, обратного и попарного суммирования на языке программирования Си и проведена проверка корректности. Были описаны алгоритмы работы вычисления функций sin(x), cos(x), exp^(x), ln(1+x) в окрестности 0 и проведено измерение погрешности для разных способов суммирования.