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 №2**

**«Суммирование рядов»**

**Выполнил**:

студент группы 381903-1

Стребкова К.О.

**Проверил**:

доцент кафедры МОСТ, к.т.н.,

Сысоев А.В.

Нижний Новгород

2019

Содержание

[Введение 3](#_Toc28523789)

[1. Постановка задачи 4](#_Toc28523790)

[2. Руководство пользователя 6](#_Toc28523791)

[3.1. Описание структуры программы 9](#_Toc28523792)

[3.2. Описание алгоритмов 10](#_Toc28523793)

[4. Результаты экспериментов 13](#_Toc28523794)

[4.1. Синус 13](#_Toc28523795)

[4.2. Косинус 14](#_Toc28523796)

[4.3. Экспонента 15](#_Toc28523797)

[4.4. Гиперболический косинус 16](#_Toc28523798)

[4.5. Натуральный логарифм 16](#_Toc28523799)

[4.6. Квадратный корень 17](#_Toc28523800)

[Заключение 19](#_Toc28523801)

[Литература 20](#_Toc28523802)

[Приложение 21](#_Toc28523803)

# Введение

Математика  является  наукой,  которая  широко  используются  на  практике. Любой  производственно-технологический  процесс  не  обходится  без  фундаментальных  математических  закономерностей.  Эффективное  применение  различных  инструментов  математического  аппарата  позволяет  конструировать  устройства  и  автоматизированные  агрегаты,  способные  выполнять  операции  с  высоким  уровнем  точности,  выполнять  сложные  расчеты  и  вычисления  при  проектировании  зданий  и  сооружений,  производить  необходимые  вычисления  при  различных  исследованиях. Одними из таких инструментов являются ряды Тейлора. Благодаря простоте и замечательным свойствам им нашли применение практически во всех разделах математики, физики и других наук.

# Постановка задачи

Разработать программу, позволяющую выполнять расчет значений для некоторого набора функций в заданной точке с заданной погрешностью за счет разложения этих функций в ряд Тейлора.

Программа должна работать в двух режимах:

1. Однократный расчет функции в заданной точке.
2. Серийный эксперимент.

**При работе в режиме 1** пользователь должен иметь возможность:

* выбрать функцию,
* задать точку x, в которой необходимо вычислить значение,
* задать точность вычисления (>= 0.000001),
* задать число элементов ряда для выполнения расчета (N - от 1 до 1000).

Расчет оценки значения выбранной функции выполняется либо до достижения заданной точности, либо до исчерпания введенного числа слагаемых. По результатам расчета программа должна вывести следующую информацию:

* эталонное значение (полученное с использованием встроенных математических функций языка программирования С),
* вычисленную оценку значения функции,
* разницу между оценкой и эталонным значением,
* количество слагаемых, которое было вычислено.

**При работе в режиме 2** пользователь должен иметь возможность:

* выбрать функцию,
* задать точку x, в которой необходимо вычислить значение,
* задать число экспериментов (NMax - от 1 до 25).

По результатам расчета программа должна вывести следующую информацию:

* эталонное значение (полученное с использованием встроенных математических функций языка программирования С),
* таблицу из NMax строк со следующими столбцами: кол-во слагаемых (от 1 до NMax), вычисленную оценку значения функции, разницу между оценкой и эталонным значением.

**Технические требования:**

* Каждый студент должен реализовать расчет функций sin(x), cos(x), exp(x) и одной уникальной функции из списка ниже. Номер требуемой функции совпадает с номером в списке группы.
* По желанию можно выбрать дополнительные функции, кроме 4-х обязательных.
* Расчет функций должен быть реализован в виде функций языка С.
* В программе должен быть использован тип указатель на функцию.
* В программе желательно использовать модули.

**Уникальные функции:**

1. Квадратный корень из 1+х - (1+x)^(1/2)
2. Логарифм от 1+х - ln(1+x)
3. Тангенс - tg(x)
4. Котангенс - ctg(x)
5. Арксинус - arcsin(x)
6. Арккосинус - arccos(x)
7. Арктангенс - arctg(x)
8. Арккотангенс - arcctg(x)
9. Гиперболический синус - sh(x)
10. Гиперболический косинус - ch(x)
11. Гиперболический тангенс - th(x)
12. Гиперболический котангенс - cth(x)
13. Обратный гиперболический синус - arsh(x)
14. Обратный гиперболический тангенс - arth(x)
15. Секанс - sec(x)
16. Косеканс - csc(x)

# Руководство пользователя

В начале программы пользователю необходимо выбрать один из двух предложенных режимов работы программы.

отчет1.1.png

Рис. 1. Выбор режима

Далее на экране будут отображены 6 пронумерованных функций. Пользователь должен выбрать одну из них, введя соответствующий порядковый номер.

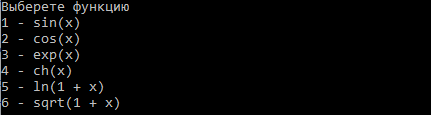


Рис. 2. Выбор функции

Затем требуется ввести значение х (точка, в которой будет производиться расчет значения выбранной функции).

****

Рис. 3. Ввод значения х

Для функций под номерами 5 и 6 на х накладываются ограничения:

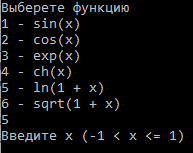


Рис. 4. Ограничения для функции №5

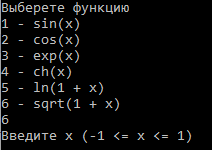


Рис. 5. Ограничения для функции №6

Последующие шаги будут отличаться в зависимости от выбранного режима.

* Режим №1 (однократный расчет функции в точке)

На данном этапе пользователь должен ввести допустимую погрешность (точность) вычисления значения функции. Значение погрешности должно быть больше, чем 0,000001.



Рис. 6. Ввод точности вычисления

Далее пользователь вводит количество слагаемых, необходимое для расчета. Это количество должно находиться в пределах от 1 до 1000 включительно.



Рис. 7. Ввод количества слагаемых

Затем на экране появится следующая информация:

* эталонное значение (полученное с использованием встроенных математических функций языка программирования С)
* вычисленную оценку значения функции,
* разницу между оценкой и эталонным значением,
* количество слагаемых, которое было вычислено.

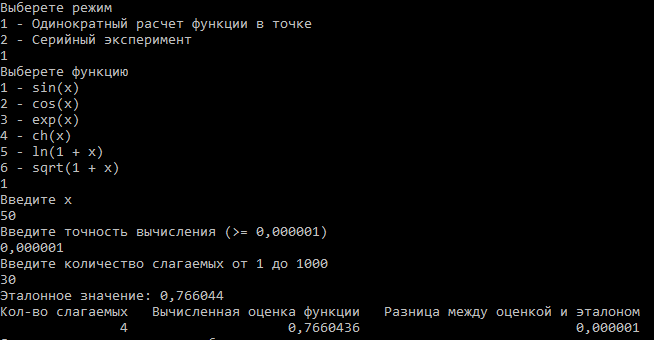


Рис. 8. Пример работы режима №1

* Режим №2 (серийный эксперимент)

Во втором режиме после ввода значения х требуется ввести нужное количество экспериментов в пределах от 1 до 25 включительно.



Рис. 9. Ввод количества экспериментов

Далее для каждого эксперимента пользователю необходимо ввести допустимую погрешность (точность) вычисления значения функции и количество слагаемых, необходимое для расчета. Ограничения для погрешности и количества слагаемых такие же, как в режиме №1.

* После ввода всех необходимых параметров на экране появится эталонное значение функции (полученное с использованием встроенных математических функций языка программирования С) и таблица со следующими столбцами: кол-во слагаемых, вычисленная оценка значения функции, разница между оценкой и эталонным значением. В таблице каждая строка отображает данные соответствующего эксперимента (номер строки равен номеру эксперимента).

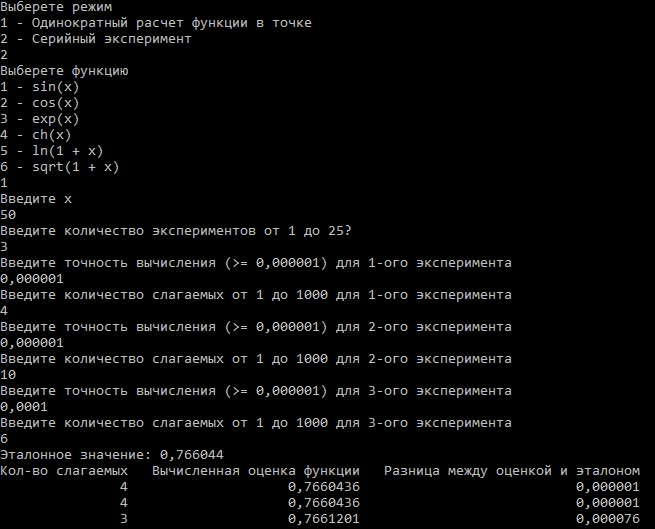


Рис. 10. Пример работы режима №2

1. **Руководство программиста**

## Описание структуры программы

Программа написана на языке СИ и состоит из одного файла Source.c.

В программе реализованы следующие функции:

1. Функция «Choice\_regime»

Описание: осуществляет выбор режима работы программы.

Передаваемые в функцию параметры: отсутствуют.

В результате возвращает значение типа int, соответствующее номеру выбранного пользователем режима.

1. Функция «Choice\_function»

Описание: осуществляет выбор функции.

Передаваемые в функцию параметры: отсутствуют.

В результате возвращает значение типа int, соответствующее номеру выбранной пользователем функции.

1. Функция «My\_sin»

Описание: вычисляет значение функции sin(x) в точке и выводит количество подсчитанных слагаемых, вычисленное значение функции в точке за счет разложения в ряд Тейлора, разницу между вычисленным и эталонным значениями.

Передаваемые в функцию параметры: значение x типа double, значение погрешности E типа float, количество слагаемых k типа int.

Не возвращает значение.

1. Функция «My\_cos»

Описание: вычисляет значение функции cos(x) в точке и выводит количество подсчитанных слагаемых, вычисленное значение функции в точке за счет разложения в ряд Тейлора, разницу между вычисленным и эталонным значениями.

Передаваемые в функцию параметры: значение x типа double, значение погрешности E типа float, количество слагаемых k типа int.

Не возвращает значение.

1. Функция «My\_exp»

Описание: вычисляет значение функции exp(x) в точке и выводит количество подсчитанных слагаемых, вычисленное значение функции в точке за счет разложения в ряд Тейлора, разницу между вычисленным и эталонным значениями.

Передаваемые в функцию параметры: значение x типа double, значение погрешности E типа float, количество слагаемых k типа int.

Не возвращает значение.

1. Функция «My\_ch»

Описание: вычисляет значение функции ch(x) в точке и выводит количество подсчитанных слагаемых, вычисленное значение функции в точке за счет разложения в ряд Тейлора, разницу между вычисленным и эталонным значениями.

Передаваемые в функцию параметры: значение x типа double, значение погрешности E типа float, количество слагаемых k типа int.

Не возвращает значение.

1. Функция «My\_ln»

Описание: вычисляет значение функции ln(1+x) в точке и выводит количество подсчитанных слагаемых, вычисленное значение функции в точке за счет разложения в ряд Тейлора, разницу между вычисленным и эталонным значениями.

Передаваемые в функцию параметры: значение x типа double, значение погрешности E типа float, количество слагаемых k типа int.

Не возвращает значение.

1. Функция «My\_sqrt»

Описание: вычисляет значение функции sqrt(1+x) в точке и выводит количество подсчитанных слагаемых, вычисленное значение функции в точке за счет разложения в ряд Тейлора, разницу между вычисленным и эталонным значениями.

Передаваемые в функцию параметры: значение x типа double, значение погрешности E типа float, количество слагаемых k типа int.

Не возвращает значение.

1. Функция «Output\_one»

Описание: осуществляет однократный эксперимент вычисления значения функции в точке, выводит эталонное значение и информацию об эксперименте.

Передаваемые параметры в функцию: указатель на функцию f, значение х типа double, эталонное значение функции func типа double.

Не возвращает значение.

1. Функция «Output\_two»

Описание: последовательно осуществляет серию экспериментов вычисления значения функции в точке, выводит на экран эталонное значение и таблицу со следующими столбцами: кол-во слагаемых, вычисленная оценка функции, разница между оценкой и эталоном.

Передаваемые в функцию параметры: указатель на функцию f, значение х типа double, эталонное значение функции func типа double.

Не возвращает значение.

## Описание алгоритмов

1. Разложение функции sin(x) в ряд Тейлора

Сначала переменной func присваивается 1, переменной term присваивается х (точка, в которой вычисляется значение синуса).

Далее запускается цикл, в котором на каждом i-ом (i от 2 до k, где k – количество слагаемых) шаге вычисляется i-ое слагаемое (term) и его значение суммируется с предыдущими слагаемыми. Сумма хранится в переменной func. Цикл заканчивается, если посчитаны все слагаемые или достигнута нужная точность оценочного значения функции.

Формула для i-ого слагаемого:

term = (-1) \* term \* x \* x / ((2 \* i - 1) \* (2 \* i - 2))

Общая формула разложения синуса в ряд Тейлора:

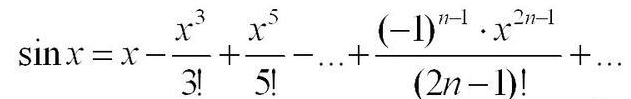


Рис. 11. Разложение синуса

1. Разложение функции cos(x) в ряд Тейлора

Сначала переменным func и term присваивается 1.

Далее запускается цикл, в котором на каждом i-ом (i от 2 до k, где k – количество слагаемых) шаге вычисляется i-ое слагаемое (term) и его значение суммируется с предыдущими слагаемыми. Сумма хранится в переменной func. Цикл заканчивается, если посчитаны все слагаемые или достигнута нужная точность оценочного значения функции.

Формула для i-ого слагаемого:

term = (-1) \* term \* x \* x / ((2 \* i - 2) \* (2 \* i - 3))

Общая формула разложения косинуса в ряд Тейлора:

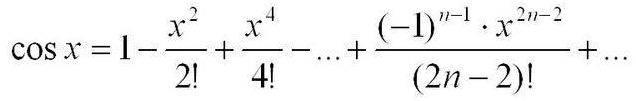


Рис. 12. Разложение косинуса

1. Разложение функции exp(x) в ряд Тейлора

Сначала переменным func и term присваивается 1.

Далее запускается цикл, в котором на каждом i-ом (i от 2 до k, где k – количество слагаемых) шаге вычисляется i-ое слагаемое (term) и его значение суммируется с предыдущими слагаемыми. Сумма хранится в переменной func. Цикл заканчивается, если посчитаны все слагаемые или достигнута нужная точность оценочного значения функции.

Формула для i-ого слагаемого:

term = term \* x / (i - 1)

Общая формула разложения экспоненты в ряд Тейлора:

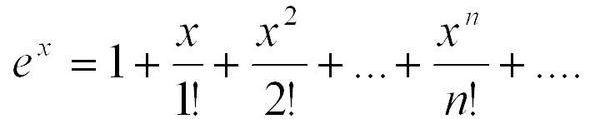


Рис. 13. Разложение экспоненты

1. Разложение функции ch(x) в ряд Тейлора

Сначала переменным func и term присваивается 1.

Далее запускается цикл, в котором на каждом i-ом (i от 2 до k, где k – количество слагаемых) шаге вычисляется i-ое слагаемое (term) и его значение суммируется с предыдущими слагаемыми. Сумма хранится в переменной func. Цикл заканчивается, если посчитаны все слагаемые или достигнута нужная точность оценочного значения функции.

Формула для i-ого слагаемого:

term = term \* x \* x / ((2 \* i - 2) \* (2 \* i - 3))

Общая формула разложения гиперболического косинуса в ряд Тейлора:

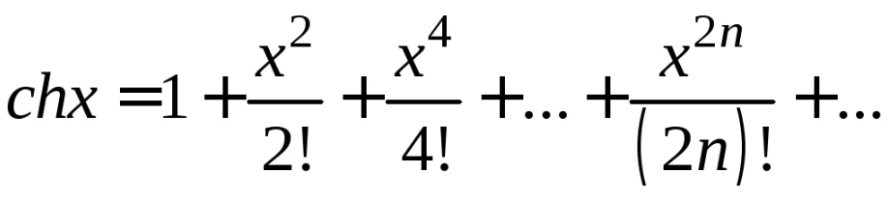


Рис. 14. Разложение гиперболического косинуса

1. Разложение функции ln(1+x) в ряд Тейлора

Сначала переменным func и term присваивается х (точка, в которой вычисляется значение функции).

Далее запускается цикл, в котором на каждом i-ом (i от 2 до k, где k – количество слагаемых) шаге вычисляется i-ое слагаемое (term) и его значение суммируется с предыдущими слагаемыми. Сумма хранится в переменной func. Цикл заканчивается, если посчитаны все слагаемые или достигнута нужная точность оценочного значения функции.

Формула для i-ого слагаемого:

term = (-1) \* term \* x / i

Общая формула разложения натурального логарифма в ряд Тейлора:

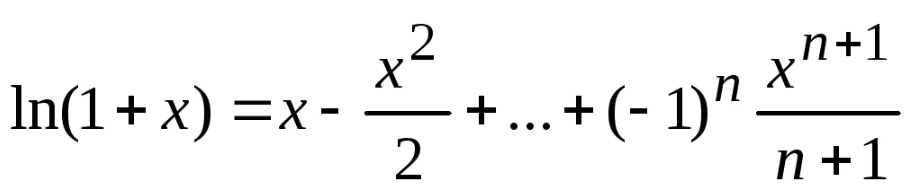


Рис. 15. Разложение натурального логарифма

1. Разложение функции (1+х)^1/2 в ряд Тейлора

Сначала переменным func и term присваивается 1.

Далее запускается цикл, в котором на каждом i-ом (i от 2 до k, где k – количество слагаемых) шаге вычисляется i-ое слагаемое (term) и его значение суммируется с предыдущими слагаемыми. Сумма хранится в переменной func. Цикл заканчивается, если посчитаны все слагаемые или достигнута нужная точность оценочного значения функции.

Формула для i-ого слагаемого:

term = term \* x \* (0.5 - i + 2) / (i - 1)

Общая формула разложения квадратного корня в ряд Тейлора:

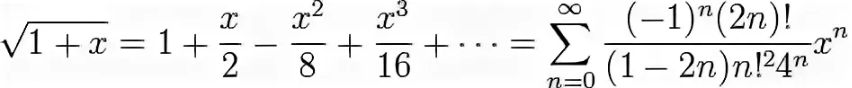


Рис. 16. Разложение квадратного корня

# Результаты экспериментов

## Синус

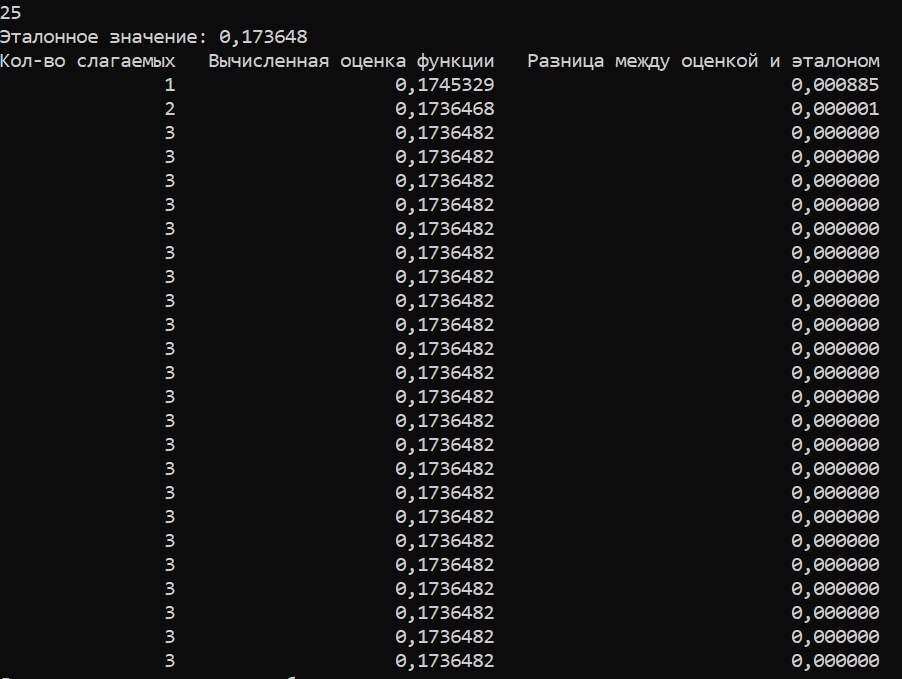


Рис. 17. Эксперимент для sin(х) при х = 10◦

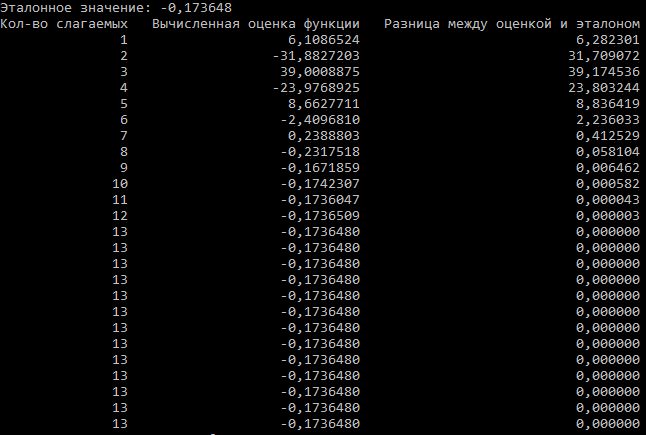


Рис. 18. Эксперимент для sin(х) при х = 350◦

## Косинус

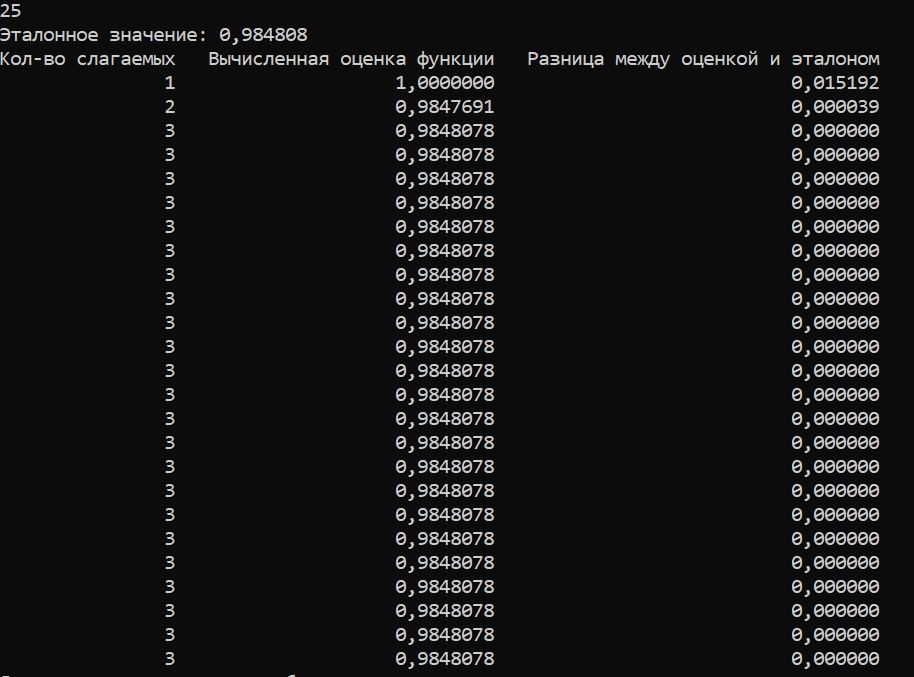


Рис. 19. Эксперимент для cos(x) при х = 10◦

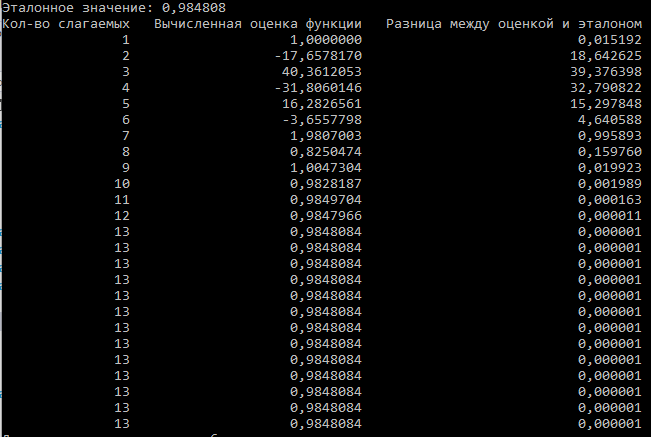


Рис. 20. Эксперимент для cos(x) при х = 350◦

## Экспонента

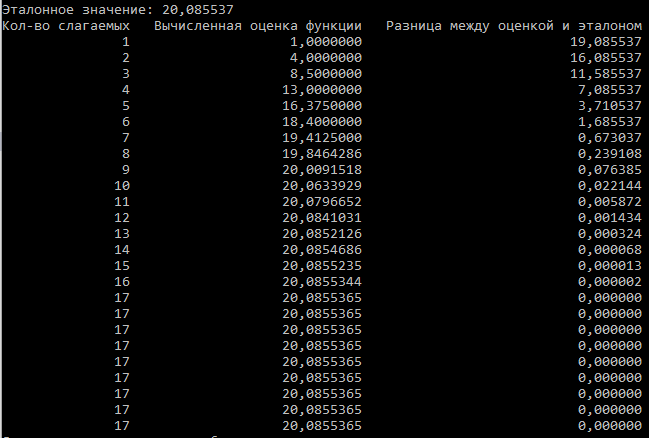
****

Рис. 21. Эксперимент для exp(x) при х = 3

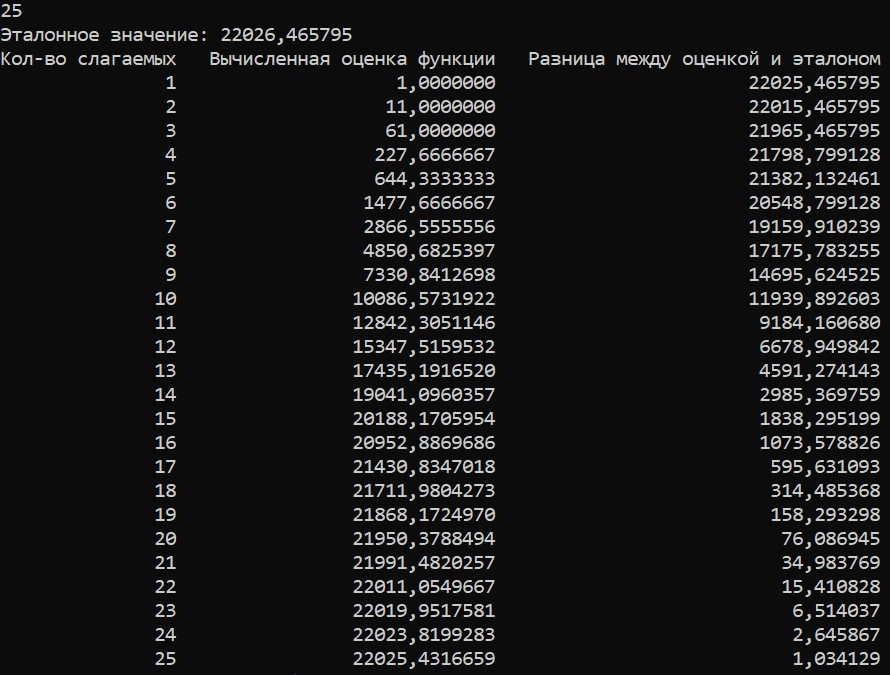
****

Рис. 22. Эксперимент для exp(x) при х = 10

## Гиперболический косинус

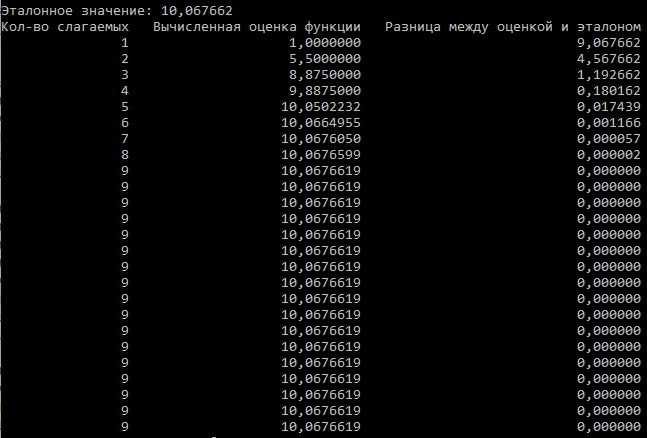
****

Рис. 23. Эксперимент для ch(x) при х = 3

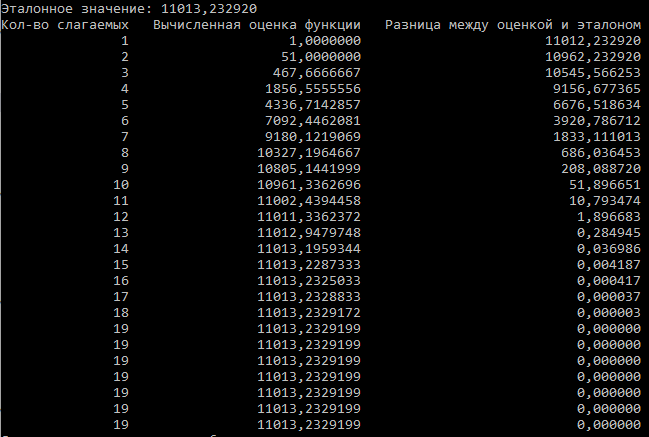


Рис. 24. Эксперимент для ch(x) при х = 10

## Натуральный логарифм

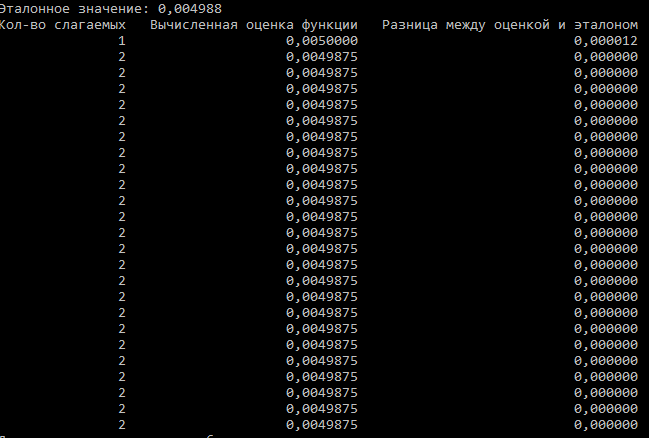
****

Рис. 25. Эксперимент для ln(1 + х) при х = 0,005

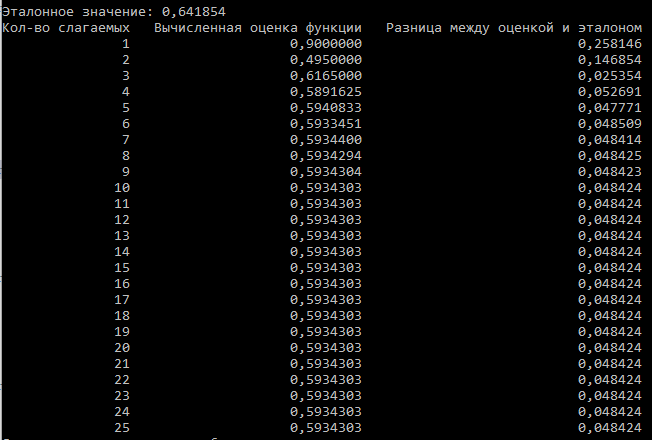


Рис. 26. Эксперимент для ln(1 + x) при х = 0,9

## Квадратный корень

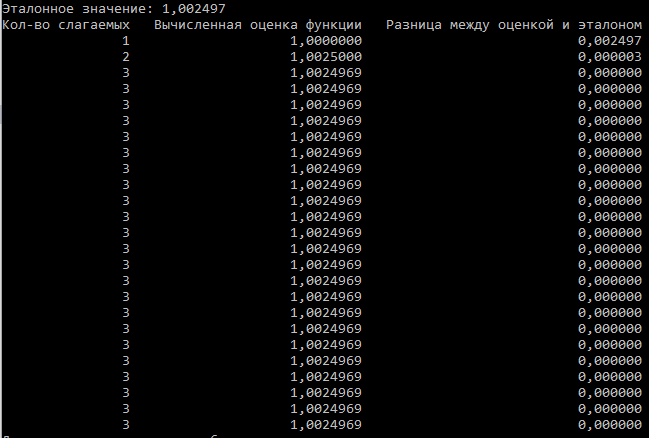
****

Рис. 27. Эксперимент для √(1 + x) для х = 0,005

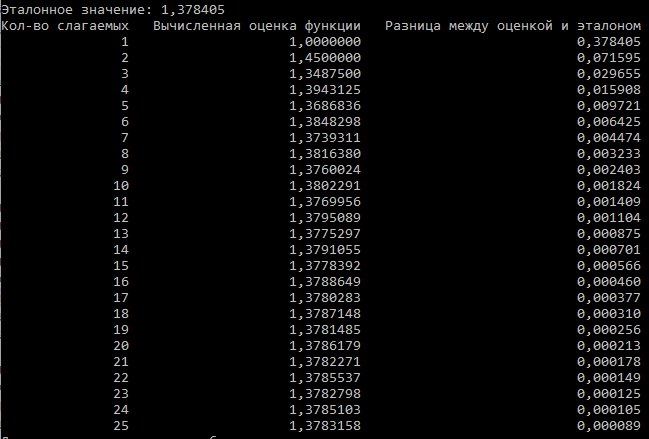


Рис. 28. Эксперимент для √(1 + x) для х = 0,9

# Заключение

Поставленная задача была выполнена. Реализована программа, которая рассчитывает приближенное значение функций sin(x), cos(x), exp(x), ch(x), ln(1+x), √(1+x) в заданной точке. Были проведены серийные эксперименты для определения точности разложений в ряды Тейлора ранее перечисленных функций. Результаты показали, что для достаточно малых аргументов разложение позволяет быстро приблизиться к эталонному значению функции в заданной точке.

# Литература

1. Сайт Studfiles. Разложение функций в степенные ряды. Ряд Тейлора. Ряд Маклорена. - <https://studfile.net/preview/6262840/page:12/>
2. Керниган, Брайан У., Ритчи, Деннис М. Язык программирования C, 2-е изд.

# Приложение

Код программы:

#define \_USE\_MATH\_DEFINES

#define N 25

#include <stdio.h>

#include <math.h>

#include <locale.h>

int choice\_regime()

{

int n;

printf("Выберете режим\n");

printf("1 - Одинократный расчет функции в точке\n");

printf("2 - Серийный эксперимент\n");

scanf\_s("%d", &n);

return n;

}

int choice\_function()

{

int n;

printf("Выберете функцию\n");

printf("1 - sin(x)\n");

printf("2 - cos(x)\n");

printf("3 - exp(x)\n");

printf("4 - ch(x)\n");

printf("5 - ln(1 + x)\n");

printf("6 - sqrt(1 + x)\n");

scanf\_s("%d", &n);

return n;

}

void my\_sin(double x, float E, int k)

{

int i;

double sin\_c = sin(x), func = x, delta = fabs(sin\_c - func), term = x;

for (i = 2; (i <= k) && (delta > E); i++)

{

term = (-1) \* term \* x \* x / ((2 \* i - 1) \* (2 \* i - 2));

func = func + term;

delta = fabs(sin\_c - func);

}

printf("%16d %26.7lf %32.6lf\n", i - 1, func, delta);

}

void my\_cos(double x, float E, int k)

{

int i;

double cos\_c = cos(x), func = 1, delta = fabs(cos\_c - func), term = 1;

for (i = 2; (i <= k) && (delta > E); i++)

{

term = (-1) \* term \* x \* x / ((2 \* i - 2) \* (2 \* i - 3));

func = func + term;

delta = fabs(cos\_c - func);

}

printf("%16d %26.7lf %32.6lf\n", i - 1, func, delta);

}

void my\_exp(double x, float E, int k)

{

int i;

double exp\_c = exp(x), func = 1, delta = fabs(exp\_c - func), term = 1;

for (i = 2; (i <= k) && (delta > E); i++)

{

term = term \* x / (i - 1);

func = func + term;

delta = fabs(exp\_c - func);

}

printf("%16d %26.7lf %32.6lf\n", i - 1, func, delta);

}

void my\_ch(double x, float E, int k)

{

int i;

double ch\_c = cosh(x), func = 1, delta = fabs(ch\_c - func), term = 1;

for (i = 2; (i <= k) && (delta > E); i++)

{

term = term \* x \* x / ((2 \* i - 2) \* (2 \* i - 3));

func = func + term;

delta = fabs(ch\_c - func);

}

printf("%16d %26.7lf %32.6lf\n", i - 1, func, delta);

}

void my\_ln(double x, float E, int k)

{

int i;

double ln\_c = log(1 + x), func = x, delta = fabs(ln\_c - func), term = x;

for (i = 2; (i <= k) && (delta > E); i++)

{

term = (-1) \* term \* x / i;

func = func + term;

delta = fabs(ln\_c - func);

}

printf("%16d %26.7lf %32.6lf\n", i - 1, func, delta);

}

void my\_sqrt(double x, float E, int k)

{

int i;

double sqrt\_c = sqrt(1 + x), func = 1, delta = fabs(sqrt\_c - func), term = 1;

for (i = 2; (i <= k) && (delta > E); i++)

{

term = term \* x \* (0.5 - i + 2) / (i - 1);

func = func + term;

delta = fabs(sqrt\_c - func);

}

printf("%16d %26.7lf %32.6lf\n", i - 1, func, delta);

}

void output\_one(void(\*f)(double, float, int), double x, double func)

{

float E;

int k;

printf("Введите точность вычисления (>= 0,000001)\n");

scanf\_s("%f", &E);

printf("Введите количество слагаемых от 1 до 1000\n");

scanf\_s("%d", &k);

printf("Эталонное значение: %lf\n", func);

printf("Кол-во слагаемых Вычисленная оценка функции Разница между оценкой и эталоном\n");

f(x, E, k);

}

void output\_two(void(\*f)(double, float, int), double x, double func)

{

float E[N];

int k[N], experience, i;

printf("Введите количество экспериментов от 1 до 25?\n");

scanf\_s("%d", &experience);

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

{

printf("Введите точность вычисления (>= 0,000001) для %d-ого эксперимента\n", i + 1);

scanf\_s("%f", &E[i]);

printf("Введите количество слагаемых от 1 до 1000 для %d-ого эксперимента\n", i + 1);

scanf\_s("%d", &k[i]);

}

printf("Эталонное значение: %lf\n", func);

printf("Кол-во слагаемых Вычисленная оценка функции Разница между оценкой и эталоном\n");

for (i = 0; i < k[i]; i++)

f(x, E[i], k[i]);

}

void main()

{

setlocale(LC\_ALL, "Rus");

int n, regime;

void(\*f)(double, float, int) = NULL;

double x, func;

regime = choice\_regime();

n = choice\_function();

if (n == 5)

printf("Введите x (-1 < x <= 1)\n");

else if (n == 6)

printf("Введите x (-1 <= x <= 1)\n");

else

printf("Введите x\n");

scanf\_s("%lf", &x);

switch (n)

{

case 1:

f = my\_sin;

x = x / 180 \* M\_PI;

func = sin(x);

break;

case 2:

f = my\_cos;

x = x / 180 \* M\_PI;

func = cos(x);

break;

case 3:

f = my\_exp;

func = exp(x);

break;

case 4:

f = my\_ch;

func = cosh(x);

break;

case 5:

f = my\_ln;

func = log(1 + x);

break;

case 6:

f = my\_sqrt;

func = sqrt(1 + x);

break;

}

if (regime == 1)

output\_one(f, x, func);

else

output\_two(f, x, func);

system("pause");

}