МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

Направление подготовки: «Программная инженерия»

**ОТЧЕТ**

по лабораторной работе

**ПРИБЛИЖЕННОЕ ВЫЧИСЛЕНИЕ**

**ПРОСТЕЙШИХ МАТЕМАТИЧЕСКИХ ФУНКЦИЙ**

**Выполнил:** студент группы

3822Б1ПР2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.А.Ермолаев

Подпись

**Проверил:** д.ф.-м.н., доц.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.А. Баркалов

Подпись

Нижний Новгород  
2022

**Содержание**

[1.ВВЕДЕНИЕ 3](#_Toc122689023)

[2.ПОСТАНОВКА ЗАДАЧИ 4](#_Toc122689024)

[3.ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc122689025)

[4.ОПИСАНИЕ ПРОГРАММЫ 6](#_Toc122689026)

[4.1 Пользовательские типы данных 6](#_Toc122689027)

[4.2 Сервисные функции 6](#_Toc122689028)

[4.3 Математические функции 6](#_Toc122689029)

[5.РЕЗУЛЬТАТЫ 8](#_Toc122689030)

[6.ЗАКЛЮЧЕНИЕ 12](#_Toc122689031)

[8.ЛИТЕРАТУРА 13](#_Toc122689032)

[9.ПРИЛОЖЕНИЕ 1 14](#_Toc122689033)

[10.ПРИЛОЖЕНИЕ 2 16](#_Toc122689034)

[11.ПРИЛОЖЕНИЕ 3 17](#_Toc122689035)

[12.ПРИЛОЖЕНИЕ 4 18](#_Toc122689036)

# 1.ВВЕДЕНИЕ

Во время обучения программированию, обучающемуся полезно знать, как работает та или иная фишка языка, на котором он пишет, “под капотом”. Поэтому мы попробуем реализовать некоторые математические функции с помощью разложения в ряд Маклорена.

# 2.ПОСТАНОВКА ЗАДАЧИ

Используя разложение функции в ряд Маклорена, реализовать математические функции: экспоненты, синуса, косинуса, натурального логарифма. Сравнить точность реализованных нами функций с функциями, встроенными в модуль math.h.

# 3.ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Ряд Маклорена – частный случай ряда Тейлора, имеет вид:

Разложение экспоненты в ряд Маклорена:

Разложение синуса в ряд Маклорена:

Разложение косинуса в ряд Маклорена:

Разложение натурального логарифма в ряд Маклорена:

# 4.ОПИСАНИЕ ПРОГРАММЫ

## 4.1 Пользовательские типы данных

{

double a, b, h;

double\* x, \* fx, \* fprx;

double (\*func1) (double, int);

double (\*func2) (double);

int n;

int size;

} programInfo – пользовательская структура, которая хранит всю необходимую информацию для функционирования программы. Передаётся в качестве аргумента в сервисные функции.

## 4.2 Сервисные функции

void print(programInfo\* info) – отвечает за вывод результатов в виде таблицы и вычисления погрешности вычислений.

void setArrayValues(programInfo\* info) – заполняет массивы значениями.

void initSegment(programInfo\* info) – инициирует ввод необходимых значений и инициализирует массивы(отрезки). При повторном вызове удаляет старый отрезок.

## 4.3 Математические функции

unsigned long long my\_factorial(unsigned long long n) – функция, итеративно вычисляющая факториал.

double my\_pow(double num, int power) – функция, итеративно возводящая num в степень power.

double my\_exp(double num, int n) – функция, вычисляющая экспоненту в лоб.

double my\_exp2(double num, int n) – функция, вычисляющая экспоненту через рекуррентную формулу.

double fast\_exp(double y, int n) – функция, вычисляющая экспоненту через разложение. (Использует функцию my\_exp2)

double my\_sin(double num, int n) - функция, вычисляющая синус через рекуррентную формулу.

double fast\_sin(double y, int n) - функция, вычисляющая синус без периода. (Использует функцию my\_sin)

double my\_cos(double num, int n) - функция, вычисляющая косинус через рекуррентную формулу.

double fast\_cos(double y, int n) - функция, вычисляющая косинус без периода. (Использует функцию my\_cos)

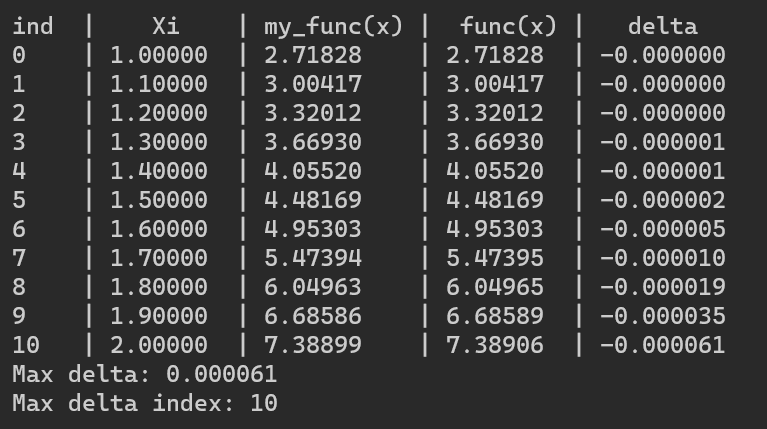
double my\_ln(double y, int n) – функция, рекуррентно вычисляющая натуральный логарифм .

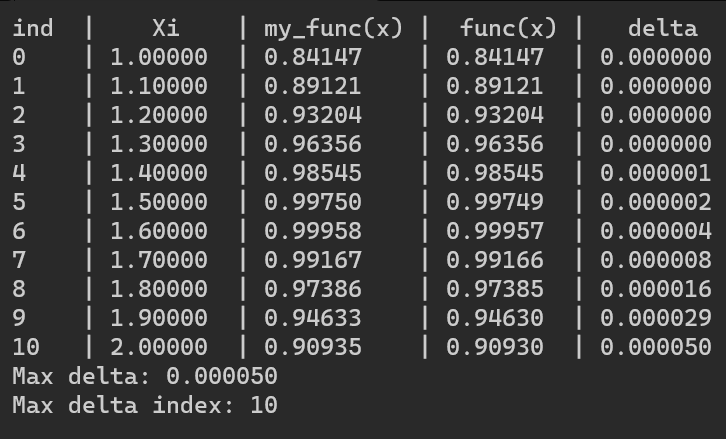
double my\_ln2(double y, int n) – функция, рекуррентно вычисляющая натуральный логарифм .

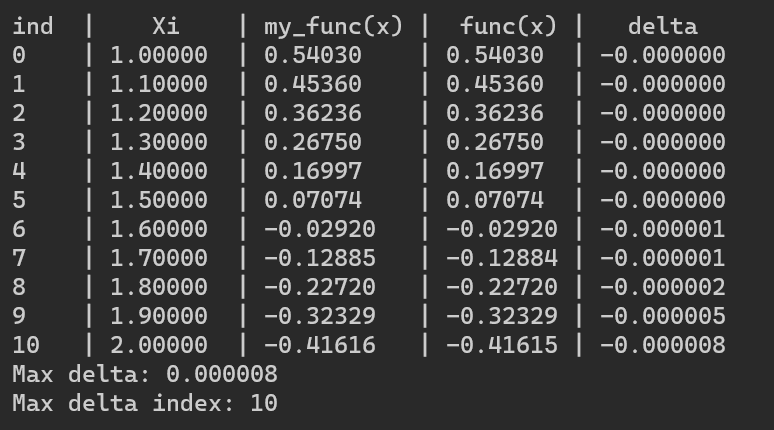
# 5.РЕЗУЛЬТАТЫ

На отрезке с шагом 0.1 и количеством слагаемых 10 были получены следующие результаты:

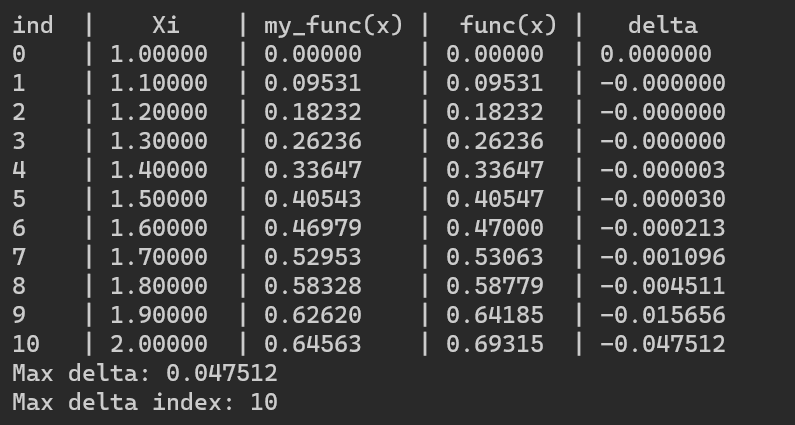
* Для экспоненты:



* Для синуса: 
* Для косинуса:

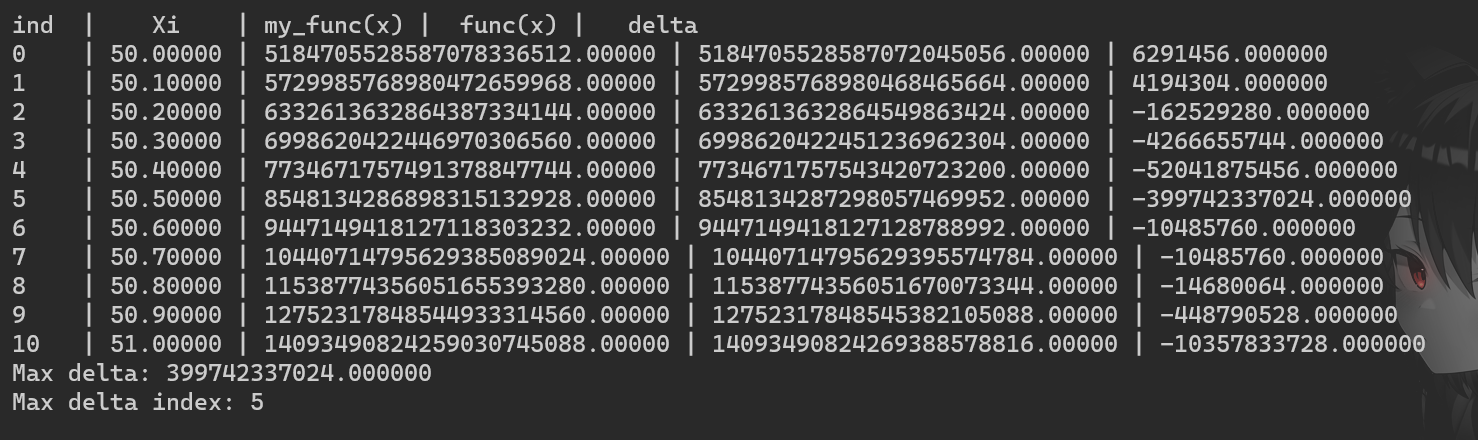


* Натуральный логарифм:

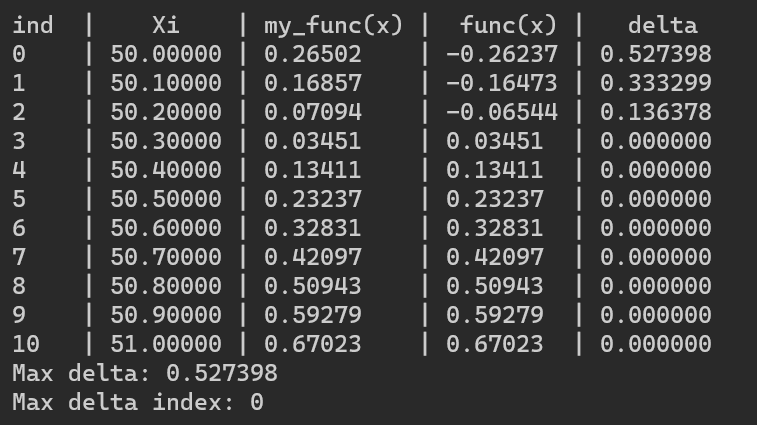


На отрезке [50, 51] и с шагом 0.1 и количеством слагаемых 10 были получены следующие результаты:

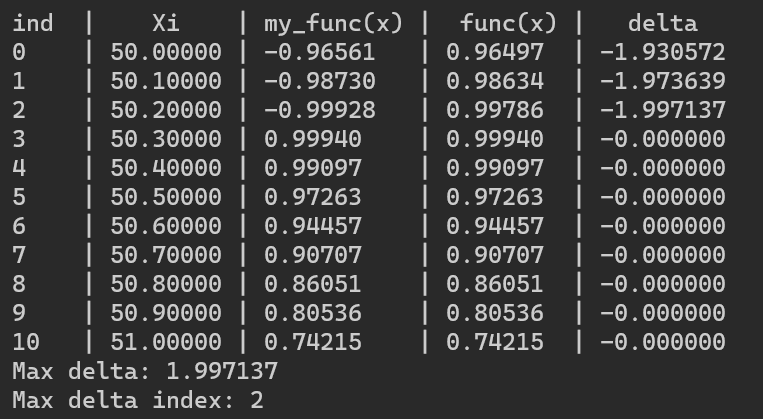
* Для экспоненты:



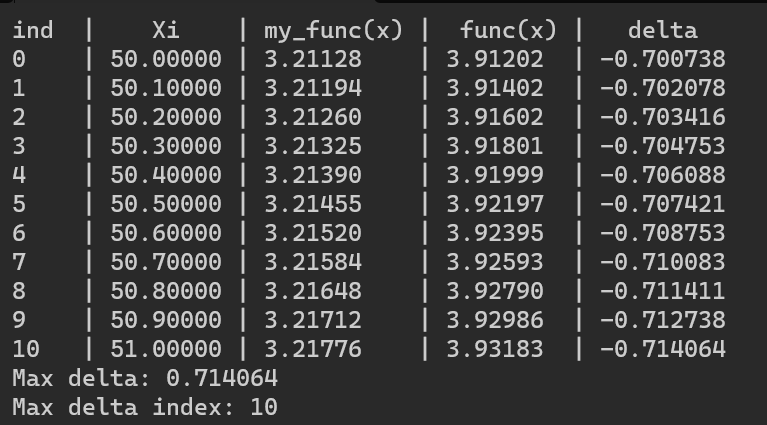
* Для синуса:



* Для косинуса:



* Для натурального логарифма:



# 6.ЗАКЛЮЧЕНИЕ

Мы, используя разложение функции в ряд Маклорена, реализовали математические функции: экспоненты, синуса, косинуса, натурального логарифма. Сравнили точность реализованных нами функций с функциями, встроенными в модуль math.h.

# 8.ЛИТЕРАТУРА

1. Дубограй И.В. Степенные ряды. Методические указания к выполнению типового расчёта. / И.В. Дубограй, Л.Н. Дьяков, О.В. Скуднева – Москва: Издательство МГТУ им. Н.Э. Баумана, 2005. – 43 с.

# 9.ПРИЛОЖЕНИЕ 1

Математические функции:

#define \_USE\_MATH\_DEFINES

#include <math.h>

unsigned long long my\_factorial(unsigned long long n)

{

unsigned long long result = 1;

for (int i = 2; i <= n; i++)

result \*= i;

return result;

}

double my\_pow(double num, int power)

{

double result = 1;

for (int i = 0; i < power; i++)

result \*= num;

return num;

}

double my\_exp(double num, int n)

{

double result = 0;

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

result += my\_pow(num, i) / my\_factorial(i);

return result;

}

double my\_exp2(double num, int n)

{

double a = 1;

double result = 1;

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

{

a \*= num / i;

result += a;

}

return result;

}

double my\_sin(double num, int n)

{

double a = num;

double result = a;

for (int i = 3; i <= n; i += 2)

{

a \*= -(num \* num) / ((i - 1) \* i);

result += a;

}

return result;

}

double my\_cos(double num, int n)

{

double a = 1;

double result = a;

for (int i = 2; i <= n; i += 2)

{

a \*= -num \* num / ((i - 1) \* i);

result += a;

}

return result;

}

// y in (0, 2]

double my\_ln(double y, int n)

{

double x = y - 1;

double result = x, a = x;

for (int i = 2; i <= n; i++)

{

a \*= -x;

result += a / i;

}

return result;

}

//ln(y) = 2(x + x^3/3 + x^5/5 + x^7/7 + ...) for all y = (1+x)/(1-x)

double my\_ln2(double y, int n)

{

double x = -(1 - y) / (y + 1);

double a = x;

double result = 0;

for (int i = 1; i <= n; i += 2)

{

result += a / i;

a \*= x \* x;

}

return 2 \* result;

}

// sin(y) = sin(2pik+x) = sin(x)

double fast\_sin(double y, int n)

{

if (y > 0)

y -= floor(y / (M\_PI)) \* M\_PI;

else

y += -floor(-y / (M\_PI)) \* M\_PI;

return my\_sin(y, n);

}

// cos(y) = cos(2pik+x) = cos(x)

double fast\_cos(double y, int n)

{

if (y > 0)

y -= floor(y / (M\_PI)) \* M\_PI;

else

y += -floor(-y / (M\_PI)) \* M\_PI;

return my\_cos(y, n);

}

// e^y = 2^k\*e^x

// y = k\*ln(2) + x

double fast\_exp(double y, int n)

{

double k = floor(y / M\_LN2);

return pow(2, k) \* my\_exp2(y - k \* M\_LN2, n);

}

# 10.ПРИЛОЖЕНИЕ 2

Сервисные функции:

#include <stdio.h>

#include <malloc.h>

#include <math.h>

#include "my\_module.h"

void initSegment(programInfo\* info)

{

if (info->x != NULL)

free(info->x);

if (info->fx != NULL)

free(info->fx);

if (info->fprx != NULL)

free(info->fprx);

printf("Enter a: ");

scanf\_s("%lf", &info->a);

printf("Enter b: ");

scanf\_s("%lf", &info->b);

printf("Enter h: ");

scanf\_s("%lf", &info->h);

printf("Enter n: ");

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

info->size = (info->b - info->a) / info->h + 1;

info->x = (double\*)malloc(sizeof(double) \* info->size);

info->fx = (double\*)malloc(sizeof(double) \* info->size);

info->fprx = (double\*)malloc(sizeof(double) \* info->size);

}

void print(programInfo \* info)

{

double delta, maxDelta = 0;

int maxDeltaIndex = 0;

printf("ind |\t Xi\t| my\_func(x) |\tfunc(x)\t| delta\n");

for (int i = 0; i < info->size; i++)

{

delta = info->fprx[i] - info->fx[i];

printf("%-\*i| %-\*.5lf | %-\*.5lf | %-\*.5lf | %-lf\n", 5, i, 8,

info->x[i], 10, info->fprx[i], 8, info->fx[i], delta);

if (fabs(delta) > maxDelta)

{

maxDelta = fabs(delta);

maxDeltaIndex = i;

}

}

printf("Max delta: %lf\n", maxDelta);

printf("Max delta index: %i\n\n", maxDeltaIndex);

}

void setArrayValues(programInfo \* info)

{

for (int i = 0; i < info->size; i++)

{

info->x[i] = info->a + i \* info->h;

info->fprx[i] = info->func1(info->x[i], info->n);

info->fx[i] = info->func2(info->x[i]);

}

}

# 11.ПРИЛОЖЕНИЕ 3

Заголовочный файл:

#pragma once

struct programInfo

{

double a, b, h;

double\* x, \* fx, \* fprx;

double (\*func1) (double, int);

double (\*func2) (double);

int n;

int size;

};

// Math Functions

unsigned long long my\_factorial(unsigned long long n);

double my\_pow(double num, int power);

double my\_exp(double num, int n);

double my\_exp2(double num, int n);

double fast\_exp(double y, int n);

double my\_sin(double num, int n);

double fast\_sin(double y, int n);

double my\_cos(double num, int n);

double fast\_cos(double y, int n);

double my\_ln(double y, int n);

double my\_ln2(double y, int n);

// Service Functions

void print(programInfo \* info);

void setArrayValues(programInfo\* info);

void initSegment(programInfo\* info);

# 12.ПРИЛОЖЕНИЕ 4

Основной файл:

#include <stdio.h>

#include <windows.h>

#include <math.h>

#include "my\_module.h"

int main()

{

programInfo myInfo;

int op, op1, loop\_flag;

myInfo.x = NULL;

myInfo.fx = NULL;

myInfo.fprx = NULL;

initSegment(&myInfo);

system("cls");

while (1)

{

printf("Choose function/operation:\n"

"1. Exp\n"

"2. Sin\n"

"3. Cos\n"

"4. Ln\n"

"9. Resize segment\n"

"0. Exit\n"

"Enter operation: ");

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

switch (op)

{

case 1:

system("cls");

myInfo.func2 = exp;

loop\_flag = 1;

while (loop\_flag)

{

printf("Choose a realization of Exp:\n"

"1. Simple Exp\n"

"2. Static Exp\n"

"3. Fast Exp\n"

"0. To main menu\n"

"Enter operation: ");

scanf\_s("%d", &op1);

switch (op1)

{

case 1:

system("cls");

myInfo.func1 = my\_exp;

break;

case 2:

system("cls");

myInfo.func1 = my\_exp2;

break;

case 3:

system("cls");

myInfo.func1 = fast\_exp;

break;

case 0:

system("cls");

loop\_flag = 0;

break;

default:

system("cls");

printf("Operation not found!\n\n");

continue;

}

if (loop\_flag)

{

setArrayValues(&myInfo);

print(&myInfo);

}

}

break;

case 2:

system("cls");

myInfo.func2 = sin;

loop\_flag = 1;

while (loop\_flag)

{

printf("Choose a realization of Sin:\n"

"1. Sin\n"

"2. Sin without period\n"

"0. To main menu\n"

"Enter operation: ");

scanf\_s("%d", &op1);

switch (op1)

{

case 1:

system("cls");

myInfo.func1 = my\_sin;

break;

case 2:

system("cls");

myInfo.func1 = fast\_sin;

break;

case 0:

system("cls");

loop\_flag = 0;

break;

default:

system("cls");

printf("Operation not found!\n\n");

continue;

}

if (loop\_flag)

{

setArrayValues(&myInfo);

print(&myInfo);

}

}

break;

case 3:

system("cls");

myInfo.func2 = cos;

loop\_flag = 1;

while (loop\_flag)

{

printf("Choose a realization of Cos:\n"

"1. Cos\n"

"2. Cos without period\n"

"0. To main menu\n"

"Enter operation: ");

scanf\_s("%d", &op1);

switch (op1)

{

case 1:

system("cls");

myInfo.func1 = my\_cos;

break;

case 2:

system("cls");

myInfo.func1 = fast\_cos;

break;

case 0:

system("cls");

loop\_flag = 0;

break;

default:

system("cls");

printf("Operation not found!\n\n");

continue;

}

if (loop\_flag)

{

setArrayValues(&myInfo);

print(&myInfo);

}

}

break;

case 4:

system("cls");

myInfo.func2 = log;

loop\_flag = 1;

while (loop\_flag)

{

printf("Choose a realization of Ln:\n"

"1. Ln(1+x), x in (-1;1]\n"

"2. ln((1+x)/(1-x))\n"

"0. To main menu\n"

"Enter operation: ");

scanf\_s("%d", &op1);

switch (op1)

{

case 1:

system("cls");

myInfo.func1 = my\_ln;

break;

case 2:

system("cls");

myInfo.func1 = my\_ln2;

break;

case 0:

system("cls");

loop\_flag = 0;

break;

default:

system("cls");

printf("Operation not found!\n\n");

continue;

}

if (loop\_flag)

{

setArrayValues(&myInfo);

print(&myInfo);

}

}

break;

case 9:

system("cls");

initSegment(&myInfo);

break;

case 0:

system("cls");

printf("@Vladislav Ermolaev");

return 0;

default:

system("cls");

printf("Operation not found!\n\n");

}

}

free(myInfo.x);

free(myInfo.fx);

free(myInfo.fprx);

}