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**

**«Математические функции»**

**Выполнил**:

студентка группы 3822Б1-ПМ1

Сосновская У.А.

**Проверил**:

преподаватель каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2023

**Содержание**

[Постановка задачи 3](#_Toc132940237)

[Метод решения 4](#_Toc132940238)

[Sin(x) 4](#_Toc132940239)

[Cos(x) 4](#_Toc132940240)

[Exp 4](#_Toc132940241)

[Ln(x+1) 4](#_Toc132940242)

[Руководство пользователя 6](#_Toc132940243)

[Описание программной реализации 7](#_Toc132940244)

[Результаты экспериментов 10](#_Toc132940245)

[Sin(x) 10](#_Toc132940246)

[Cos(x) 10](#_Toc132940247)

[Exp 11](#_Toc132940248)

[Ln(x+1) 12](#_Toc132940249)

[Заключение 13](#_Toc132940250)

[Приложение 14](#_Toc132940251)

# Постановка задачи

Мы изучили, каким образом подсчитываются встроенные математические функции на языке Си, нам требовалось написать программу, которая считает значение sin(x), cos(x), exp и ln(1+x) с помощью разложения этих функция в ряд Тейлора. Причем искать значение необходимо двумя способами. С помощью прямого и обратного суммирования. После этого мы должны были сравнить результат наших функций со встроенными, а также между собой.

**Цель работы**: создание программы, считающей значение математических функция

**Задачи**:

* Создание программы
* Сравнение результатов работы различных функций
* Написание отчета

# Метод решения

Для данной лабораторной работы необходимо было реализовать четыре математические функции. Чтобы посчитать значение каждой из них, мы использовали ряды Тейлора. В общую функцию суммирования (прямую и обратную) поступали различные слагаемые, каждое из них подсчитывалось определенным образом.

### Sin(x)

Слагаемое для прямого суммирования высчитывается по формуле

Слагаемое для обратного суммирования высчитывается по формуле

### Cos(x)

Слагаемое для прямого суммирования высчитывается по формуле

Слагаемое для обратного суммирования высчитывается по формуле

### Exp

Слагаемое для прямого суммирования высчитывается по формуле

Слагаемое для обратного суммирования высчитывается по формуле

### Ln(x+1)

Слагаемое для прямого суммирования высчитывается по формуле

Слагаемое для обратного суммирования высчитывается по формуле

# Руководство пользователя

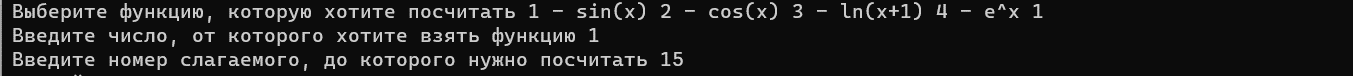
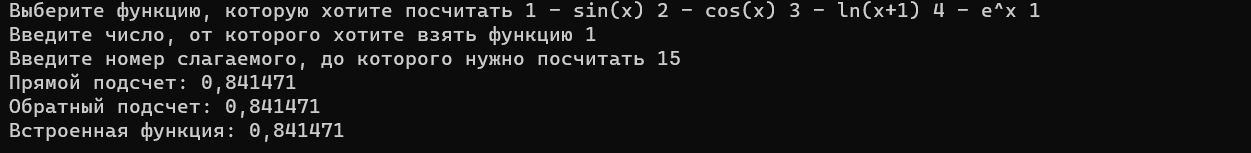
 После запуска программы перед пользователем появляется окно, куда он должен ввести необходимые данные (какую именно математическую функцию, значение которой он хочет найти, число, результат подстановки которого нужно посчитать, и слагаемое, до которого предположительно должна быть просуммирована функция). Затем пользователю будут предоставлены результаты подсчета результата тремя способами. С помощью прямого суммирования, с помощью обратного суммирования и с помощью встроенных функций

Рис. 2

Рис. 1

# Описание программной реализации

Программа состоит из одного файла (math functions.cpp), где были реализованы все необходимые функции.

float sinslg(float a, int o, float x) – функция, которая используется для подсчета слагаемых для прямого суммирования функции sin(x). Принимает значение предыдущего слагаемого, номер слагаемого, которое нужно посчитать, и значение переменной, от которой мы ищем значение функции. Возвращает значение искомого слагаемого.

float sinslgrev(float an, int pos, float x)– функция, которая используется для подсчета слагаемых для обратного суммирования функции sin(x). Принимает значение следующего слагаемого, номер следующего слагаемого, которое нужно посчитать, и значение переменной, от которой мы ищем значение функции. Возвращает значение искомого слагаемого.

float cosslg(float a, int o, float x) – функция, которая используется для подсчета слагаемых для прямого суммирования функции cos(x). Принимает значение предыдущего слагаемого, номер слагаемого, которое нужно посчитать, и значение переменной, от которой мы ищем значение функции. Возвращает значение искомого слагаемого.

float cosslgrev(float an, int pos, float x)– функция, которая используется для подсчета слагаемых для обратного суммирования функции cos(x). Принимает значение следующего слагаемого, номер следующего слагаемого, которое нужно посчитать, и значение переменной, от которой мы ищем значение функции. Возвращает значение искомого слагаемого.

float expslg(float a, int o, float x) – функция, которая используется для подсчета слагаемых для прямого суммирования функции exp. Принимает значение предыдущего слагаемого, номер слагаемого, которое нужно посчитать, и значение переменной, от которой мы ищем значение функции. Возвращает значение искомого слагаемого.

float expslgrev(float an, int pos, float x)– функция, которая используется для подсчета слагаемых для обратного суммирования функции exp. Принимает значение следующего слагаемого, номер следующего слагаемого, которое нужно посчитать, и значение переменной, от которой мы ищем значение функции. Возвращает значение искомого слагаемого.

float lnslg(float a, int o, float x) – функция, которая используется для подсчета слагаемых для прямого суммирования функции ln(x+1). Принимает значение предыдущего слагаемого, номер слагаемого, которое нужно посчитать, и значение переменной, от которой мы ищем значение функции. Возвращает значение искомого слагаемого.

float lnslgrev(float an, int pos, float x)– функция, которая используется для подсчета слагаемых для обратного суммирования функции ln(x+1). Принимает значение следующего слагаемого, номер следующего слагаемого, которое нужно посчитать, и значение переменной, от которой мы ищем значение функции. Возвращает значение искомого слагаемого.

float smfn(float x, int posl, float ak, float(\*sl)(float, int, float))– функция прямого суммирования. Принимает значение переменной, значении функции в которой нужно посчитать, номер слагаемого, до которого нужно посчитать, первый элемент суммирования, указатель на функцию подсчета слагаемого нужной функции. В ней с помощью цикла подсчитывается значение функции и возвращает его.

float poslslog(float x, int posl, float ak, float(\*sl)(float, int, float))– функция, которая подсчитывает первое значение для прямого суммирования. Принимает значение переменной, значении функции в которой нужно посчитать, номер слагаемого, до которого нужно посчитать, первый элемент суммирования, указатель на функцию подсчета слагаемого нужной функции. Внутри ищется значение последнего ненулевого слагаемого при прямом суммировании и возвращается его значение.

int poslnom(float x, int posl, float ak, float(\*sl)(float, int, float))– функция, которая подсчитывает первое значение для прямого суммирования. Принимает значение переменной, значении функции в которой нужно посчитать, номер слагаемого, до которого нужно посчитать, первый элемент суммирования, указатель на функцию подсчета слагаемого нужной функции. Внутри ищется значение последнего ненулевого слагаемого при прямом суммировании и возвращается его номер.

loat smfnrvs(float x, int posl, float an, float(\*slrvs)(float, int, int, float))– функция обратного суммирования. Принимает значение переменной, значении функции в которой нужно посчитать, номер слагаемого, с которого нужно посчитать, последний элемент прямого суммирования (первый элемент обратного), указатель на функцию подсчета слагаемого нужной функции. В ней с помощью цикла подсчитывается значение функции и возвращает его.

int main() – основная функция. В ней происходит вызов функций суммирования. Выводит на экран результат работы функций и значение встроенных математических функций.

# Результаты экспериментов

Целью проведения эксперимента было выявление точности работы математических функций. Следовало установить, при каком виде суммирования получатся более точные результаты. Изучив результаты экспериментов, мы можем заметить, что в большинстве случаев, когда у нас присутствует погрешность, разница между прямым суммированием и встроенной функцией меньше. Такой результат мог получиться, так как подсчёт значения функции обратным суммированием напрямую зависит от прямого суммирования. Первым элементом обратного суммирования становится последний элемент прямого.

### Sin(x)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Значение переменной/количество слагаемых | Прямая сумма | Обратная сумма | Встроенная функция | Разница между встроенной и прямой | Разница между встроенной и обратной | Более точная функция |
| 1/10 | 0,841471 | 0,841471 | 0,841471 | 0 | 0 | - |
| 1/100 | 0,841471 | 0,841471 | 0,841471 | 0 | 0 | - |
| 2/10 | 0,909297 | 0,909297 | 0,909297 | 0 | 0 | - |
| 7/15 | 0,656983 | 0,656983 | 0,656987 | 0,000004 | 0,000004 | - |
| 10/15 | -0,544226 | -0,542917 | -0,544021 | 0,000205 | 0,000896 | прямая |
| 10/100 | -0,544120 | -0,544138 | -0,544021 | 0,000099 | 0,000117 | прямая |
| 15/100 | 0,672943 | 0,641971 | 0,650288 | 0,022655 | 0,008317 | обратная |
| 0,1/10 | 0,099833 | 0,099833 | 0,099833 | 0 | 0 | - |
| 0,1/2 | 0,099833 | 0,099833 | 0,099833 | 0 | 0 | - |
| 0,5/5 | 0,479426 | 0,479426 | 0,479426 | 0 | 0 | - |
| 0,5/2 | 0,479427 | 0,479427 | 0,479426 | 0,000001 | 0,0000001 | - |

### Cos(x)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Значение переменной/количество слагаемых | Прямая сумма | Обратная сумма | Встроенная функция | Разница между встроенной и прямой | Разница между встроенной и обратной | Более точная функция |
| 1/10 | 0,540302 | 0,540303 | 0,540302 | 0 | 0,000001 | прямая |
| 1/100 | 0,540302 | 0,540303 | 0,540302 | 0 | 0,000001 | прямая |
| 2/10 | -0,416147 | -0,416147 | -0,416147 | 0 | 0 | - |
| 7/15 | 0,753902 | 0,753908 | 0,753902 | 0 | 0,000006 | прямая |
| 10/15 | -0,839517 | -0,835812 | -0,839072 | 0,000445 | 0,00326 | прямая |
| 10/100 | -0,839168 | -0,839230 | -0,839072 | 0,000096 | 0,000158 | прямая |
| 15/100 | -0,755031 | -0,761719 | -0,759688 | 0,004657 | 0,002031 | обратная |
| 0,1/10 | 0,995004 | 0,995004 | 0,995004 | 0 | 0 | - |
| 0,1/2 | 0,995004 | 0,995000 | 0,995004 | 0 | 0 | - |
| 0,5/5 | 0,877583 | 0,877583 | 0,877583 | 0 | 0,000001 | прямая |
| 0,5/2 | 0,877604 | 0,875000 | 0,877583 | 0,000021 | 0,002583 | прямая |

### Exp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Значение переменной/количество слагаемых | Прямая сумма | Обратная сумма | Встроенная функция | Разница между встроенной и прямой | Разница между встроенной и обратной | Более точная функция |
| 1/10 | 2,718282 | 2,718282 | 2,718282 | 0 | 0 | - |
| 1/100 | 2,718282 | 2,718282 | 2,718282 | 0 | 0 | - |
| 2/10 | 7,388995 | 7,388712 | 7,389056 | 0,000061 | 0,000344 | прямая |
| 7/15 | 1093,994019 | 1090,363403 | 1096,633158 | 2,639139 | 6,269755 | прямая |
| 10/15 | 20952,888672 | 20188,169922 | 22026,465795 | 1073,577123 | 22026,465795 | прямая |
| 10/100 | 22026,468750 | 22026,462891 | 22026,465795 | 0,002955 | 0,002904 | обратная |
| 15/100 | 3269017,500000 | 3269016,000000 | 3269017,372472 | 0,127528 | 1,372472 | обратная |
| 0,1/10 | 1,105171 | 1,105171 | 1,105171 | 0 | 0 | - |
| 0,1/2 | 1,105000 | 1,100000 | 1,105171 | 0,000171 | 0,005171 | прямая |
| 0,5/5 | 1,648698 | 1,648438 | 1,648721 | 0,000023 | 0,000284 | прямая |
| 0,5/2 | 1,625000 | 1,500000 | 1,648721 | 0,023721 | 0,148721 | прямая |

### Ln(x+1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Значение переменной/количество слагаемых | Прямая сумма | Обратная сумма | Встроенная функция | Разница между встроенной и прямой | Разница между встроенной и обратной | Более точная функция |
| 0,1/10 | 0,095310 | 0,095310 | 0,095310 | 0 | 0 | - |
| 0,1/2 | 0,095333 | 0,095000 | 0,095310 | 0,000023 | 0,000310 | прямая |
| 0,5/5 | 0,404687 | 0,407292 | 0,405465 | 0,000778 | 0,001827 | прямая |
| 0,5/10 | 0,405479 | 0,405435 | 0,405465 | 0,000014 | 0,000030 | прямая |
| 0,5/100 | 0,405465 | 0,405465 | 0,405465 | 0 | 0 | - |

# Заключение

Написав программу, содержащую функции подсчета математических функций, сравнив результаты, мы смогли выявить наиболее удачный вариант подсчета значений математических функций.

# Приложение

#include "stdio.h"

#include "math.h"

#include "stdlib.h"

#include "locale.h"

float sinslg(float a, int o, float x)

{

float an;

an = (-1) \* a \* x \* x / ((2\*o+1) \* (2\*o));

return an;

}

float sinslgrev(float an, int o, int pos, float x)

{

float a;

a =( an \* (-1) \* ((2\*pos-2)\*(2\*pos-1)) )/ (x \* x);

return a;

}

float cosslg(float a, int o, float x)

{

float an;

an = ((-1) \* (x \* x) \* a) / (2 \* o \* (2 \* o - 1));

return an;

}

float cosslgrev(float an, int o, int pos, float x)

{

float a;

a = ((-1) \* an \*(2\*pos-3)\* (2\*pos-2) / (x \* x));

return a;

}

float expslg(float a, int o, float x)

{

float an;

an = a \* x / o;

return an;

}

float expslgrev(float an, int o, int pos, float x)

{

float a;

a = (an \* (pos-1) / x);

return a;

}

float lnslg(float a, int o, float x)

{

float an;

an = ((-1) \* x \* a \* o) / (o + 1);

return an;

}

float lnslgrev(float an, int o, int pos, float x)

{

float a;

if (pos == 1)

a = 0;

else

a = ((-1) \* an \* (pos) / ((pos-1) \* x));

return a;

}

float smfn(float x, int posl, float ak, float(\*sl)(float, int, float))

{

float sm = ak;

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

{

ak = sl(ak, i, x);

sm += ak;

}

return sm;

}

float poslslog(float x, int posl, float ak, float(\*sl)(float, int, float))

{

float ai;

int i = 1;

while ((ak != 0) && (i <= posl))

{

if (ak >= 0.0000001 || ak <= -0.000001)

ai = ak;

ak = sl(ak, i, x);

i++;

}

return ai;

}

int poslnom(float x, int posl, float ak, float(\*sl)(float, int, float))

{

int nomer = 0;

int i = 1;

while ((ak != 0) && (i <= posl))

{

if (ak >= 0.0000001 || ak <= -0.000001)

nomer = i;

ak = sl(ak, i, x);

i++;

}

return nomer;

}

float smfnrvs(float x, int posl, float an, float(\*slrvs)(float, int, int, float))

{

float sm = an;

for (int i = posl; i >= 1; i--)

{

an = slrvs(an, posl, i, x);

sm += an;

}

return sm;

}

int main()

{

int n, o;

float x, ak;

setlocale(LC\_ALL, "Rus");

printf("Выберите функцию, которую хотите посчитать ");

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

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

printf("3 - ln(x+1) ");

printf("4 - e^x ");

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

printf("Введите число, от которого хотите взять функцию ");

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

printf("Ввeдите номер слагаемого, до которого нужно посчитать ");

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

if (n == 1)

{

ak = x;

printf("Прямой подсчет: %f \n", smfn(x, o, ak, sinslg));

printf("Обратный подсчет: %f \n", smfnrvs(x, poslnom(x, o, ak, sinslg), poslslog(x, o, ak, sinslg), sinslgrev));

printf("Встроенная функция: %f\n", sin(x));

}

if (n == 2)

{

ak = 1;

printf("Прямой подсчет: %f \n", smfn(x, o, ak, cosslg));

printf("Обратный подсчет: %f \n", smfnrvs(x, poslnom(x, o, ak, cosslg), poslslog(x, o, ak, cosslg), cosslgrev));

printf("Встроенная функция: %f\n", cos(x));

}

if (n == 3)

{

ak = x;

printf("Прямой подсчет: %f \n", smfn(x, o, ak, lnslg));

printf("Обратный подсчет: %f \n", smfnrvs(x, poslnom(x, o, ak, lnslg), poslslog(x, o, ak, lnslg), lnslgrev));

printf("Встроенная функция: %f\n", log(x + 1));

}

if (n == 4)

{

ak = 1;

printf("Прямой подсчет: %f \n", smfn(x, o, ak, expslg));

printf("Обратный подсчет: %f \n", smfnrvs(x, poslnom(x, o, ak, expslg), poslslog(x, o, ak, expslg), expslgrev));

printf("Встроенная функция: %f\n", exp(x));

}

return 0;

}