Министерство образования и науки Российской Федерации

Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

**Отчет по лабораторной работе**

**“Математические функции”**

**Выполнил**:студент группы 381606-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Тимакин Н. Е.

Подпись

**Проверил**: к.ф.-м.н.,доц.

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

Подпись

Нижний Новгород

2016

**Введение**

В современных ЭВМ напрямую не заложены алгоритмы вычисления таких сложных функций, как показательные, тригонометрические или логарифмические. Но их нетрудно представить с помощью более простых операций: сложения, вычитания, умножения и деления. Это можно реализовать при помощи формулы Тейлора или её частного случая - формулы Маклорена:

**Постановка задачи**

В программе должны быть реализованы 4 функции: ex,sin(x), cos(x), ln(x). Причём ln(x) должна работать только на полуинтервале (0;1], а остальные на всей области определения, то есть на интервале (-.

**Описание алгоритмов**

**Описание структуры программы**

Программа состоит из трёх модулей:

1. funcs.h - заголовочный файл

Он содержит включения стандартных библиотек <stdio.h> и <math.h>, а также прототипы функций:

* double mexp(double x,int n); - экспонента для маленьких значений переменной
* double fexp(double x,int n); - экспонента для больших значений переменной
* double fsin(double x,int n); - синус
* double fcos(double x,int n); - косинус
* double fln(double x,int n); натуральный логарифм для маленьких значений переменной
* void table(double x,double h,int n,int m,funcd stdf,funcdi myf); - построение таблицы результатов вычислений

1. funcs.cpp - файл, содержащий тела функций
2. main.cpp - файл, содержащий функцию:

* void main() - выполняет вычисление экспоненты, логарифма, синуса или косинуса на заданном отрезке, с заданным шагом и с заданной точностью. После вычислений выдаёт таблицу результатов, в которой отражается значение x, значение функции из стандартной библиотеки, значение функции из funcs.h, и разницу между последними двумя значениями. В конце подсчитывается максимальное значение погрешности.

**Результаты экспериментов**

На отрезке [-1;1] с шагом 0.1 и точностью 10 слагаемых максимальная погрешность составила:

* 1.489756 для экспоненты
* 0.000057 для синуса
* 0.000212 для косинуса
* 0.183837 для логарифма

На отрезке [-1;1] с шагом 0.1 и точностью 20 слагаемых максимальная погрешность составила:

* 1.489756 для экспоненты
* 0.000000 для синуса
* 0.000000 для косинуса
* 0.039235 для логарифма

На отрезке [0;20] с шагом 1 и точностью 10 слагаемых максимальная погрешность составила:

* 0.022086 для экспоненты
* 0.000083 для синуса
* 0.000301 для косинуса
* 579255062774.178590 для логарифма

На отрезке [-20;0] с шагом -1 и точностью 10 слагаемых максимальная погрешность составила:

* 1.888027 для экспоненты
* 0.000083 для синуса
* 0.000301 для косинуса

**Заключение**

Исходя их результатов экспериментов, можно сказать, что значения функций синус и косинус считаются довольно точно, и уже при 20 слагаемых они неотличимы от стандартных. Погрешность экспоненты не такая большая, но уже заметная, так что она непригодна для точных вычислений. А вот логарифм корректно считается только на полуинтервале (0;2], дальше величина погрешности становится слишком большой.

**Литература**

Б. Керниган, Д. Ритчи “Язык программирования Си”

А.О. Грудзинский, И.Б. Мееров, А.В. Сысоев “Методы программирования”

Г.М. Фихтенгольц “Основы математического анализа”

**Приложение**