Лабораторная работа №1

Вычисление элементарной функции с помощью разложения в ряд Тейлора

Балахонский Дмитрий Михайлович 22928/3

Вариант 7

# Задание

В первой части лабораторной работы вычисления элементарной функции с помощью разложения в ряд Тейлора проводятся при фиксированном числе слагаемых n.

Для каждого n выводятся в столбик заданные значения аргумента, во второй столбец выводятся значения функции по Тейлору, в третьем столбце приводится погрешность вычислений (разность значения функции по Тейлору и соответствующей библиотечной функции).

Во второй части работы вычисления выполняются с учетом заданного параметра точности. Под этим понимаем абсолютное значение последнего вычисленного члена ряда e.

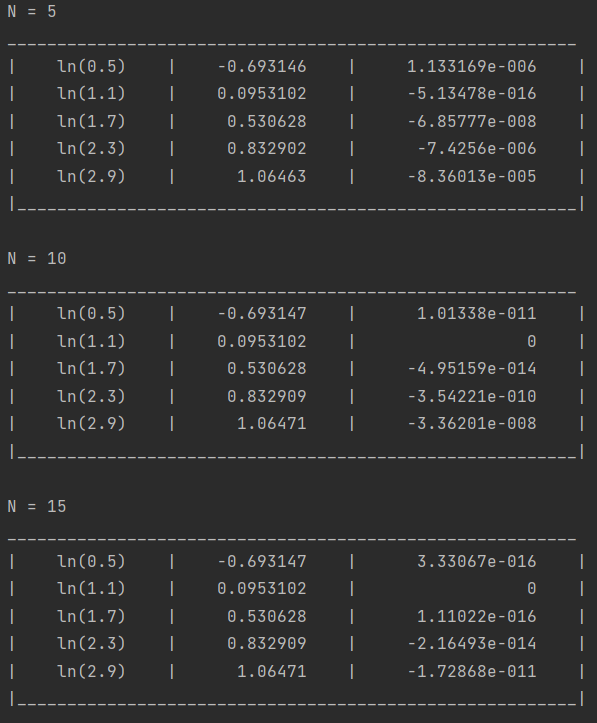
Для каждого e выводятся в столбик числа итераций цикла, далее следует столбец значений аргумента, в третий столбец выводятся значения функции по Тейлору и в четвертый - погрешность вычислений.

Ход Выполнения:

1. Для начала я написал 2 функции для вычисления ряда Тейлора
   1. tailersRow\_N
   2. tailersRow\_E
2. В основном файле я написал циклы, которые перебирают x n и z
3. Вычисления производятся через функции и выводятся в столбики))

using namespace std;  
  
string set\_length(int finalLength, string st){ // Функция для выравнивания Вывода  
 // string length calculation  
 int stringLength = st.length();  
 int spaceQ = finalLength - stringLength;  
  
 for (int i = 0; i < spaceQ; i++){  
 st += " ";  
 }  
 return st;  
}  
  
string set\_spaces(int spaceQ){ // Табуляция для красивого вывода))  
 string result = "";  
 for (int i = 0; i < spaceQ; i++){  
 result += " ";  
 }  
 return result;  
}  
  
double tailersRow\_N(double x, int n){  
 short numberSign = 1;  
 double summand = (x-1) / (x+1);  
 double result = summand;  
 for(int i = 3; i <= n \* 2; i += 2){  
 summand \*= ((x-1)\*(x-1))/((x+1)\*(x+1));  
 result += summand/i;  
 }  
 return result\*2;  
}  
  
vector <double> tailersRow\_E(double x, double e){  
 double i = 1;  
 double summand = (x-1) / (x+1);  
 double result = summand;  
 while(abs(summand/i) >= e){  
 summand \*= ((x-1)\*(x-1))/((x+1)\*(x+1));  
 result += summand / i;  
 i++;  
 }  
 return { result\*2, i };  
}

#include <iostream>  
#include <vector>  
#include <cmath>  
#include <string>  
#include <iomanip>  
#include "LabsCode/customFunctions.h"  
  
using namespace std;  
  
const vector<int> N = { 5, 10, 15 };  
const vector<double> X = { 0.5, 1.1, 1.7, 2.3, 2.9 };  
const vector<double> E = { 0.001, 0.0001, 0.00001 };  
const int tab = 4;  
  
int main(){  
  
 for(auto n : N){  
 cout << "\nN = " << n << endl;  
 cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";  
 for(auto x : X){  
 double biba = tailersRow\_N(x, n); // Вычисление ln  
 // Вывод в столбики  
 cout.fill(' ');  
 cout << "|" << set\_spaces(tab) << "ln("; cout.width(3); cout << x; cout << ")" << set\_spaces(tab); // Вывод Х  
 cout << "|" << set\_spaces(tab); cout.width(9); cout << biba; cout << set\_spaces(tab); // Вывод результата Функции  
 cout.setf(ios::scientific);  
 cout << "|" << set\_spaces(tab); cout.width(14); cout << biba-log(x); cout << set\_spaces(tab); // Вывод погрешности  
 cout.setf(ios::fixed);  
 cout << "|" << "\n";  
 }  
 cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";  
 }  
  
 /\*  
  
 ✔️ task 2 ✔️  
  
 \*/  
  
 for(auto e : E){  
 cout << "\nE = " << e << endl;  
 cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";  
 for(auto x : X){  
 vector<double> biba = tailersRow\_E(x, e); // Вычисление ln  
 cout.fill(' ');  
 cout << "|" << set\_spaces(tab) << biba[1] << set\_spaces(tab); // Вывод индекса  
 cout << "|" << set\_spaces(tab) << "ln("; cout.width(3); cout << x; cout << ")" << set\_spaces(tab); // Вывод Х  
 cout << "|" << set\_spaces(tab); cout.width(9); cout << biba[0]; cout << set\_spaces(tab); // Вывод результата Функции  
 cout.setf(ios::scientific);  
 cout << "|" << set\_spaces(tab); cout.width(14); cout << biba[0]-log(x); cout << set\_spaces(tab); // Вывод погрешности  
 cout.setf(ios::fixed);  
 cout << "|" << "\n";  
 }  
 cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n";  
 }  
}



Изображение выглядит как текст, монитор, табличка

Автоматически созданное описание