Министерство науки и высшего образования Российской Федерации

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

ОТЧЕТ

по дисциплине «Алгоритмизация и программирование»

**Лабораторная работа № 4**

**Выполнил:**

Cтудент гр. 5130902/30003 Садчев Р.В.

**Проверил**

Ст. преподаватель Журавская А.М.

Санкт-Петербург

2023 г.

**Задание**

* 1. **Формулировка задания**

Есть – рядная функция, есть – функция взятая из стандартной математической библиотеки.

Необходимо:

* Вывести рекуррентную форму приращения рядной функции;
* написать программу для вычисления значения рядной и библиотечной функции;
* вычислить невязку (дельту) формула (1) значений рядной функции и стандартной библиотечной функции;
* проанализировать динамику изменения значения невязки в зависимости от количества слагаемых в ряде.

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

где – рядная функция,

– библиотечная функция.

Вычисления ряда проводить до условия минимизации значения разности двух соседних членов ряда меньше заданного формула (2). Ряд рассчитывать через рекуррентную форму расчета ряда, что есть в приложении А и Б в конце документа.

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

Результаты вычислений оформить виде таблицы!

Задание по варианту 18:

**Код программы**

#include <iostream>  
#include <cmath>  
#include <iomanip>  
#include <limits>  
#include <string>  
  
using namespace std;  
  
double binom(int n, int k) {  
 if (k == 0 || k == n) {  
 return 1.0;  
 } else {  
 return binom(n - 1, k - 1) + binom(n - 1, k);  
 }  
}  
  
double fact(int n) {  
 if (n == 0) {  
 return 1.0;  
 } else {  
 return n \* fact(n - 1);  
 }  
}  
  
double cos2\_recursive(int n, double x) {  
 return pow(-1, n + 1) \* (pow(2, 2 \* n - 1) \* pow(x, 2 \* n)) / fact(2 \* n);  
}  
  
double cos2\_series(double x, double E) {  
 double result = 1.0;  
 double term = 1.0;  
 int n = 1;  
  
 do {  
 term = cos2\_recursive(n, x);  
 result -= term;  
 n++;  
 } while (fabs(term) > E);  
  
 return result;  
}  
  
int main() {  
 double E, x\_start, x\_end, dx, x\_ideal;  
 string NewStr = "0";  
 int count = 0;  
 do {  
 cout << "Введите точность E (от 10^-7 до 10^-1): ";  
 cin >> E;  
 if (cin.fail() || E < 1e-7 || E > 0.1) {  
 cout << "Пожалуйста, введите корректное значение для точности E (от 10^-7 до 10^-1)." << endl;  
 cin.clear();  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 }  
 } while (E < 1e-7 || E > 0.1);  
  
 do {  
 cout << "Введите границы промежутка вычисления x\_start и x\_end:";  
 cin >> x\_start >> x\_end;  
 if (cin.fail() || x\_start >= x\_end) {  
 cout << "Пожалуйста, введите корректные значения границ промежутка x\_start и x\_end (x\_start должен быть меньше x\_end)." << endl;  
 cin.clear();  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 }  
 } while (x\_start >= x\_end);  
  
 do {  
 cout << "Введите шаг dx:";  
 cin >> dx;  
 if (cin.fail() || dx <= 0) {  
 cout << "Пожалуйста, введите корректное положительное число для шага dx." << endl;  
 cin.clear();  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 }  
 } while (dx <= 0);  
  
 do {  
 cout << "Введите значение x\_ideal для точного расчёта:";  
 cin >> x\_ideal;  
 if (cin.fail()) {  
 cout << "Пожалуйста, введите числовое значение для x\_ideal." << endl;  
 cin.clear();  
 cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
 }  
 } while (cin.fail());  
  
 cout << left << setw(12) << "| x" << "| f(x)" << " | F(x)" << " | delta |" << endl;  
 cout << setw(45) << setfill('-') << "" << setfill(' ') << endl;  
  
 for (double x = x\_start; x <= x\_end; x += dx) {  
 double f\_x = cos2\_series(x, E);  
 double F\_x = cos(x) \* cos(x);  
 double delta = sqrt(fabs(f\_x \* f\_x - F\_x \* F\_x)) - 1.0;  
  
 cout << left << setw(12) << "| " + to\_string(x).substr(0, 6) << "| " + to\_string(f\_x).substr(0, 10) << "| " + to\_string(F\_x).substr(0, 10) << "| " + to\_string(delta).substr(0, 10) << " |" << endl;  
 }  
  
 cout << "\nТаблица 2:\n";  
 cout << left << setw(12) << "| E" << "| f(x\_i) " << "| F(x\_i) " << "| delta |" << endl;  
 cout << setw(45) << setfill('-') << "" << setfill(' ') << endl;  
  
 for (double current\_E = 0.1; current\_E >= 1e-7; current\_E \*= 0.1) {  
 count++;  
 double f\_x\_ideal = cos2\_series(x\_ideal, current\_E);  
 double F\_x\_ideal = cos(x\_ideal) \* cos(x\_ideal);  
 double delta\_ideal = sqrt(fabs(f\_x\_ideal \* f\_x\_ideal - F\_x\_ideal \* F\_x\_ideal)) - 1.0;  
  
 cout << left << setw(12) << "| " + to\_string(current\_E).substr(0, 13) + NewStr << "| " + to\_string(f\_x\_ideal).substr(0, 10) << "| " + to\_string(F\_x\_ideal).substr(0, 10) << "| " + to\_string(delta\_ideal).substr(0, 10) << " |" << endl;  
 if (count == 6)  
 NewStr = "1";  
  
 }  
  
 return 0;  
}

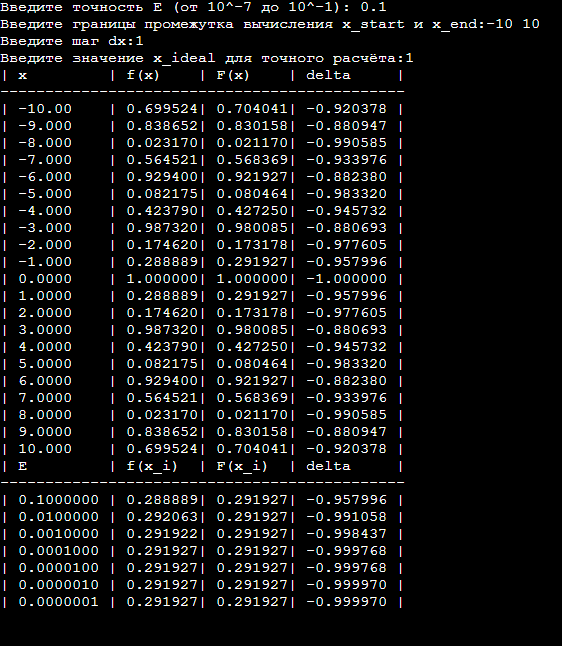


Рисунок 1 – Правильная работа алгоритма

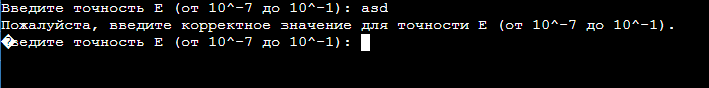


Рисунок 2 – Работа алгоритма при вводе неправильных данных

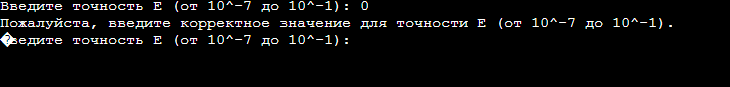


Рисунок 3 – Работа алгоритма при вводе неправильных данных

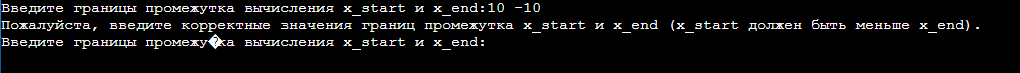


Рисунок 4 – Работа алгоритма при вводе неправильных данных

**Вывод**

В ходе выполнения лабораторной работы все цели и задачи были выполнены, были получены навыки работы с функциями, циклами, условным оператором, а также с оформлением вывода данных в терминале с помощью языка программирования C++, а также изучена математическая библиотека cmath. В результате выполнения программы были вычислены значения функции для значений аргумента X с учётом точности изменяющегося в интервале от X\_начальное до X\_конечное c шагом ∆x.