МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

“Московский Авиационный Институт

(Национальный Исследовательский Университет)”

Факультет информационных технологий и прикладной математики

Кафедра 806 “Вычислительная математика и программирование”

Курсовая работа

по курсу “Вычислительные системы”

1 семестр

Задание 3. Вещественный тип. Приближенные вычисления. Табулирование функций

Студент: Шелаев С.И

Группа: М8О-108Б-22,

№ по списку 24

Руководитель: Сахарин Н.А.

Дата: 28.12.12

Оценка:

Москва, 2022

**Содержание**

ЗАДАЧА …………………………………………………………………...…. 3

ВАРИАНТ …………………………………………………………………..... 3

ОБЩИЙ МЕТОД РЕШЕНИЯ ……………………………………………… 3

ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ ……………………………………. 3

ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ……………………………………. 4

ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ ……………………………… 4

ОПИСАНИЕ ПЕРЕМЕННЫХ, КОНСТАНТ И ПОДПРОГРАММ ………. 4

ПРОТОКОЛ ………………………………………………………………….. 6

ВХОДНЫЙ ДАННЫЕ ……………………………………………………..... 7

ВЫХОДНЫЕ ДАННЫЕ ………………………………………………...…... 7

ВЫВОДЫ …………………………………………………………………….. 9

**1. Задача**

Составить программу на Си, которая печатает таблицу значений элементарной функции, вычисленной двумя способами: по формуле Тейлора и с помощью встроенных функций языка программирования. В качестве аргументов таблицы взять точки разбиения отрезка [a, b] на n равных частей (n + 1 точка включая концы отрезка), находящихся в рекомендованной области хорошей точности формулы Тейлора. Вычисления по формуле Тейлора проводить по экономной в сложностном смысле схеме с точностью ε\*k, где ε - машинное эпсилон аппаратно реализованного вещественного типа для данной ЭВМ, а k - экспериментально подбираемый коэффициент, обеспечивающий приемлемую сходимость. Число итераций должно ограничиваться сверху числом порядка 100. Программа должна сама определять машинное ε и обеспечивать корректные размеры генерируемой таблицы.

**2. Вариант**

| № | Ряд | a | b | Функция |
| --- | --- | --- | --- | --- |
| 24 |  | 0,0 | 1,0 |  |

**3. Общий метод решения**

Вычисление значения функции в некоторой точке на отрезке от 0,0 до 1,0 двумя способами.

1 способ - использование программных средств, встроенных в стандартную математическую библиотеку языка Си “math.c”.

2 способ - при помощи ряда Тейлора.

**4. Общие сведения о программе**

Аппаратное обеспечение: домашний ноутбук

Операционная система: Linux Ubuntu, версия 22.04.1 LTS

Язык и система программирования: С, GNU

**5. Функциональное назначение**

Программа предназначена для высокоточного вычисления вещественного значения трансцедентных функций в алгебраической форме с использованием ряда Тейлора и при помощи встроенных программных функций библиотеки языка Си.

**6. Описание логической структуры**

Программа вычисляет значение функции в данной точке при помощи разложения по ряду Тейлора и с использованием программных средств языка программирования СИ. Ряд Тейлора мы преобразуем в функцию, которая вычисляет слагаемые ряда. Далее мы складываем полученные слагаемые, пока их количество не превысит 100 или значение одного из них не станет совсем мало (меньше ε\*k по модулю). В конце мы выводим таблицу с значением аргумента, значением функции, вычисленным с помощью ряда Тейлора и с использованием программной библиотеки, и номером итерации.

**7. Описание переменных, констант и подпрограмм**

Таблица 1. Описание функций программы

| Функция | Входные аргументы | Описание |
| --- | --- | --- |
| machine\_eps | - | Функция для подсчета машинного ε. Сравниваем 1+ε с 1. Последнее число, при стремлении ε к нулю, при котором 1+ε > 1 и будет машинным ε |
| func | long double x | Вычисляет значение входной функции при помощи встроенной библиотеки “math.c” |
| form\_teilor | long double x, int n, | Используя схему Горнера, считает сумму ряда по формуле Тейлора |

Таблица 2. Описание переменных

| Переменная | Значение |
| --- | --- |
| long double eps | Машинный эпсилон |
| const long double k | Эмпирический коэффициент для эпсилон |
| int MAX\_ITER | Максимальное число итераций |
| long double b, а | Границы отрезка |
| long double n | Количество частей, на которые разбивается отрезок [a, b] |
| long double result, sum | Сумма ряда |
| long double x | Значение аргумента функции |
| int n | Текущая итерация |

**8. Протокол**

Код программы:

#include <stdio.h>

#include <math.h>

const long double k = 10e2;

const int max\_iters = 100;

long double machine\_eps() {

long double eps = 1.0;

while ((1.0 + eps) > 1.0) {

eps/=2;

}

return eps;

}

long double func(long double x) {

return exp( pow(x,2));

}

long double factorial(int n) {

long double ans = 1;

for (long double i = 2; i <= n; i ++) {

ans = ans \* i;

}

return ans;

}

int main () {

long double ans, f;

int n, cnt;

printf("Машинное эпсилон для типа long double = %.20Lf\n", machine\_eps());

printf("Введите число n\n");

scanf("%d", &n);

printf("n = %d, \n", n);

long double a = 0.0;

long double b = 1.0;

printf("Таблица значений ряда Тейлора и стандартной функции для f(x) = e^(x^2)\n");

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("| x | sum | f(x) |число итераций|\n");

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

long double x = 0;

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

long double d = 1;

x = x+((a - b)/n);

ans = 0;

cnt = 1;

f = func (x);

while (fabs(d) > machine\_eps() \* k && cnt < max\_iters ) {

d = (pow(x,2\*cnt)) / (factorial(cnt));

ans = ans + d;

cnt++;

}

printf("| %.3Lf | %.20Lf | %.20Lf | %d |\n", x, ans, f, cnt);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

}

return 0;

}

**9. Входные данные**

**10. Выходные данные**