МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Факультет компьютерных наук и прикладной математики

Кафедра вычислительной математики и программирования

**КУРСОВОЙ ПРОЕКТ**

**по курсу**

**«Вычислительные системы»**

Задание 3

«Вещественный тип. Приближённые вычисления. Табулирование функций»

**Выполнил:**

студент группы М8О-108Б-22

Формалёв Александр Сергеевич

**Преподаватель:**

Сахарин Никита Александрович

Дата:

Подпись:

Москва, 2022

**Цель работы**

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 \* 10k, 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.

**Задание**

**Функция:**

**Ряд:**

**Отрезок:** [-1;1]

**Теоретическая часть.**

**Машинное эпсилон** — числовое значение, меньше которого невозможно задавать относительную точность для любого алгоритма, возвращающего вещественные числа. Абсолютное значение для машинного эпсилон зависит от разрядности сетки применяемой ЭВМ и от разрядности используемых при расчёте чисел. Формально это машинное эпсилон определяют как минимальное число, удовлетворяющее неравенству 1 + ε > 1. **Машинный ноль** — числовое значение с таким отрицательным порядком, которое воспринимается машиной как ноль.

**Формула Тейлора:** Пусть функция определена в окрестности точки и имеет в этой окрестности производные до порядка включительно, и пусть существует . Тогда

.

Многочлен – многочлен Тейлора.

При – формула Маклорена.

**Код программы**

#include <stdio.h>

#include <float.h>

#include <math.h>

#define MAX\_ITER 100

#define K 0.1

**typedef** **struct** {

**double** sum;

**int** iters\_count;

} Taylor;

**double** epsilon(**void**) {

**double** eps = 1.0;

**while** (eps + 1.0 > 1.0) {

eps /= 2.0;

}

**return** eps;

}

Taylor taylor\_log(**double** func\_arg) {

Taylor res;

res.sum = 0.0;

res.iters\_count = 1;

**double** term = log(2), x = func\_arg;

**if** (res.sum + term == res.sum) {

**return** res;

}

res.sum += term;

**for** (**int** n = 1; (n < MAX\_ITER) && (term > (epsilon() \* K)); n++) {

res.iters\_count++;

term = pow(-1, n-1)\*(pow(x, n)/(n\*pow(2, n)));

**if** (res.sum + term == res.sum) **break**;

res.sum += term;

}

**return** res;

}

**void** output(**double** a, **double** b, **int** n) {

**double** step = (b - a) / n, x;

printf(" x log taylor\_log i\n");

**for** (**int** i = 0; i < n; i++) {

x = a + i \* step;

printf("%.5lf %.10lf %.10lf %d \n", x, log(2+x), taylor\_log(x).sum, taylor\_log(x).iters\_count);

}

}

**int** main(**void**) {

**const** **double** a = -1.0, b = 1.0;

**const** **int** n = 30;

output(a, b, n);

**return** 0;

}

**Входные данные**

На вход поступает число: количество точек, значения функции в которых должна вычислить программа двумя способами.

**Выходные данные**

Программа должна вывести значение машинного эпсилон, а затем таблицу значений функции с их аргументами. В каждой строке находится значение аргумента, для которого вычисляется функция, затем значение, вычисленное с помощью формулы Тейлора, и, наконец, значение, вычисленное с помощью встроенных функций языка.

**Протокол исполнения**

x log taylor\_log i

-1.00000 0.0000000000 0.1931471806 2

-0.93333 0.0645385211 0.2264805139 2

-0.86667 0.1251631430 0.2598138472 2

-0.80000 0.1823215568 0.2931471806 2

-0.73333 0.2363887781 0.3264805139 2

-0.66667 0.2876820725 0.3598138472 2

-0.60000 0.3364722366 0.3931471806 2

-0.53333 0.3829922523 0.4264805139 2

-0.46667 0.4274440148 0.4598138472 2

-0.40000 0.4700036292 0.4931471806 2

-0.33333 0.5108256238 0.5264805139 2

-0.26667 0.5500463369 0.5598138472 2

-0.20000 0.5877866649 0.5931471806 2

-0.13333 0.6241543091 0.6264805139 2

-0.06667 0.6592456289 0.6598138472 2

0.00000 0.6931471806 0.6931471806 2

0.06667 0.7259370034 0.7259249583 3

0.13333 0.7576857017 0.7575916250 3

0.20000 0.7884573604 0.7881471806 3

0.26667 0.8183103235 0.8175916250 3

0.33333 0.8472978604 0.8459249583 3

0.40000 0.8754687374 0.8731471806 3

0.46667 0.9028677115 0.8992582917 3

0.53333 0.9295359586 0.9242582917 3

0.60000 0.9555114450 0.9481471806 3

0.66667 0.9808292530 0.9709249583 3

0.73333 1.0055218656 0.9925916250 3

0.80000 1.0296194172 1.0131471806 3

0.86667 1.0531499146 1.0325916250 3

0.93333 1.0761394328 1.0509249583 3

Program ended with exit code: 0Вывод

В ходе выполнения курсового проекта были получены знания о реализации вещественных типов данных в языке Си. Также были изучены некоторые математические аспекты, такие как формула Тейлора и ряд Маклорена.