МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский Авиационный Институт»
(Национальный Исследовательский Университет)

Институт: №8 «Информационные технологии и прикладная
математика»
Кафедра: 806 «Вычислительная математика и программирование»

Курсовая работа
по курсу «Фундаментальная информатика»
I семестр
Задание 3
«Вещественный тип. Приближенные вычисления. Табулирование
функций»

Группа	М8О-109Б-22
Студент	Шамбилов Р.Т.
Преподаватель	Сысоев М.А.
Оценка	
Дата	

Москва, 2022

Постановка задачи

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 $\varepsilon * 10^k$, 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.

Вариант 5:

Ряд Тэйлора:

$$-\frac{4x^2}{2}+\frac{16x^4}{24}+\ldots+(-1)^n\frac{(2x)^{2n}}{(2n)!}$$

Функция:

$$2(\cos^2 x-1)$$

Значения a и b: 0.0 и 0.5

Теоретическая часть

Формула Тейлора — формула разложения функции в бесконечную сумму степенных функций. Формула широко используется в приближённых вычислениях, так как позволяет приводить трансцендентных функций к более простым. Сама она является следствием теоремы Лагранжа о среднем значении дифференцируемой функции. В случае a=0 формула называется рядом Маклорена.

$$\sum_{n=0}^{k} \frac{f^{(n)}(a)}{n!}(x-a)^n = f(a)+f^{(1)}(a)(x-a)+\frac{f^{(2)}(a)}{2!}(x-a)^2+\ldots+\frac{f^{(k)}(a)}{k!}(x-a)^k$$

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 число, удовлетворяющее равенству $1 + \varepsilon = 1$. Фактически, два отличных от нуля числа являются равными с точки зрения машинной арифметики, если их модуль разности меньше или не превосходит машинное эпсилон.

В языке Си машинные эпсилон определено для следующих типов: float – $1.19 * 10^{-7}$, double – $2.20 * 10^{-16}$, long double – $1.08 * 10^{-19}$.

Описание алгоритма

Рассмотрим алгоритм решения. Сперва нужно найти машинное эпсилон, на котором будет основываться точность вычисления. Это можно сделать просто деля 1 на 2.

Для каждой N+1 строки нужно просуммировать i членов формулы Тейлора, пока $|A_1-A_2|> \varepsilon$. Для этого просто ищем каждый новый член из формулы Тэйлора и суммируем с результатом

Использованные в программе переменные

Название переменной	Тип переменной	Смысл переменной
n	int64_t	То самое число N, на которое нужно разбить отрезок
k	int	То самое число K, используемое для вычисления точности.
FLT_EPSILON	float	То самое машинное эпсилон. 1.192092896e-07F
step	long double	Формально разница между предыдущим значением из отрезка и следующим, если отрезок разбит на n равных частей.
x	long double	Переменная, для которой будем производить вычисления
Taylor(i, x)	long double	То самое значение A_1, вычисленное с помощью формулы Тейлора
func	long double	То самое значение A_2, вычисленное с помощью встроенных функций языка
i	int	Счётчик члена формулы Тейлора + кол-во итераций

Исходный код программы:

```
#include <stdio.h>
#include <float.h>
#include <stdint.h>
#include <math.h>

uint64_t factorial(uint64_t n) {
    uint64_t res = 1;
    for (uint64_t i = 1; i <= n; ++i)
        res *= i;

    return res;
}

long double Taylor(uint64_t n, long double x) {
    long double res = 0;
    for (int i = 1; i <= n; ++i) {
        res += (powl(-1, i) * (powl(2 * x, 2 * i))) / factorial(2 *
i);
    }
    return res;
}

long double function(long double x) {
    return (2 * (cos(x) * cos(x) - 1));
}

int main() {
    long double a = 0.0;
    long double b = 0.5;

    uint64_t n;
```

```
    scanf("%ld", &n);
    printf("n = %ld\n", n);
    printf("Machine epsilon is equal to: %Lg\n\n",
LDBL_EPSILON);

    printf("        Table of values of Taylor series and standard
function\n");

printf("_____
_____\n");
    printf("|  x  | sum of Taylor series | f(x) function value |
number of iterations |\n");

printf("_____
_____\n");

    long double x = 0;
    long double step = (b - a) / n;
    long double func = 1;
    int i = 0;
    while (fabsl(func) > LDBL_EPSILON && (i < 100) &&
(i < n)) {
        i += 1;
        x += step;
        func = function(x);

        printf("|%.3llf|%.16llf|%.16llf|          %d          |\n", x,
Taylor(i, x), func, i);
    }
```

```
    printf("_____
    _____\n");

    return 0;
}
```

Входные данные

Единственная строка содержит одно целое число N (0≤N≤100) – число разбиений отрезка на равные части

Выходные данные

Программа должна вывести значение машинного эпсилон, а затем N+1 строку.

В каждой строке должно быть значение x, для которого вычисляется функция, число A_1 — значение, вычисленное с помощью формулы Тейлора, A_2 – значение, вычисленное с помощью встроенных функций языка, i – количество итерация, требуемых для вычисления, и Δ – разница значений A_1 и A_2 по модулю. A_1, A_2 и Δ должны быть выведены с точностью 16 знаков после запятой.

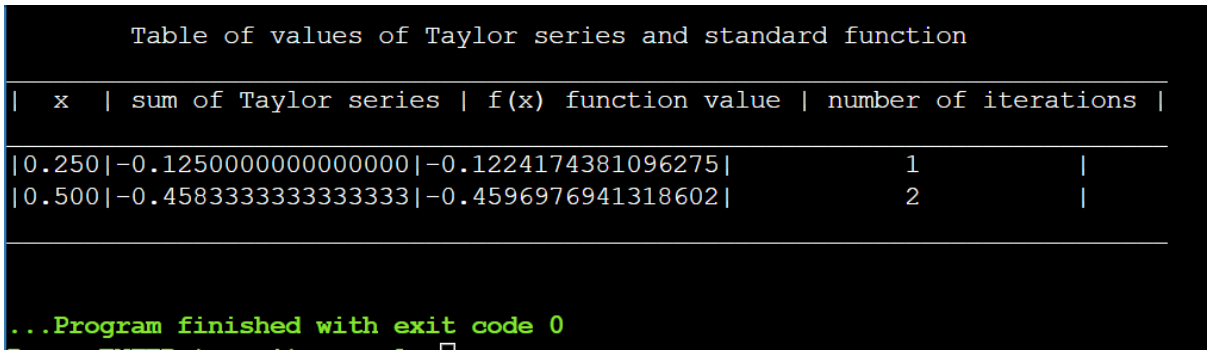
Протокол исполнения и тесты

Тест №1

Ввод:

2

Вывод:



Тест №2

Ввод:
100

Вывод:

```
100
n = 100
Machine epsilon is equal to: 1.0842e-19

        Table of values of Taylor series and standard function
_____________________________________________________________________________
|  x   | sum of Taylor series | f(x) function value | number of iterations |
_____________________________________________________________________________
|0.005|-0.0000500000000000|-0.0000499995833347|           1           |
|0.010|-0.0001999933333333|-0.0001999933334222|           2           |
|0.015|-0.0004499662510125|-0.0004499662510125|           3           |
|0.020|-0.0007998933390221|-0.0007998933390221|           4           |
|0.025|-0.0012497396050338|-0.0012497396050335|           5           |
|0.030|-0.0017994600647958|-0.0017994600647957|           6           |
|0.035|-0.0024489997467204|-0.0024489997467203|           7           |
|0.040|-0.0031982936973806|-0.0031982936973807|           8           |
|0.045|-0.0040472669880057|-0.0040472669880056|           9           |
|0.050|-0.0049958347219742|-0.0049958347219741|           10           |
|0.055|-0.0060439020433031|-0.0060439020433030|           11           |
```

Тест №3

Ввод:

323232

Вывод:

```
323232
n = 323232
Machine epsilon is equal to: 1.0842e-19

        Table of values of Taylor series and standard function
_____________________________________________________________________________
|  x   | sum of Taylor series | f(x) function value | number of iterations |
_____________________________________________________________________________
|0.000|-0.0000000000047857|-0.0000000000047855|           1           |
|0.000|-0.0000000000191426|-0.0000000000191425|           2           |
|0.000|-0.0000000000430709|-0.0000000000430709|           3           |
|0.000|-0.0000000000765705|-0.0000000000765703|           4           |
|0.000|-0.0000000001196414|-0.0000000001196412|           5           |
|0.000|-0.0000000001722835|-0.0000000001722835|           6           |
|0.000|-0.0000000002344971|-0.0000000002344969|           7           |
|0.000|-0.0000000003062819|-0.0000000003062817|           8           |
|0.000|-0.0000000003876380|-0.0000000003876379|           9           |
```

Вывод

В работе описано определение машинного эпсилон, приведены его значения для разных переменных языка Си, описана формула Тейлора и составлен алгоритм реализации вычисления значения функции с заданной точностью для заданного числа точек на отрезке. На основе алгоритма составлена программа на языке Си, проведено её тестирование на различных тестах, составлен протокол исполнения программы. В целом, работа понравилась. Приятно применять знания из других областей для решения какой-либо задачи по программированию.

Список литературы

1. Машинный ноль – URL: https://ru.wikipedia.org/wiki/Машинный_ноль
2. Ряд Тейлора – URL: https://ru.wikipedia.org/wiki/Ряд_Тейлора