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 | Мозговой Н.Е. |
| 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 $\varepsilon$ - 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 $\varepsilon$ и обеспечивать корректные размеры генерируемой таблицы.

## Вариант 25:

Ряд Тэйлора:

$$\frac{1}{4}+\frac{x^4}{4^2}+\ldots+\frac{x^{4n}}{4^{n+1}}$$

Функция:

$$\frac{1}{4-x^4}$$

Значения a и b: 0.0 и 1.0

## 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 програме 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.<br>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 |
| f | long double | То самое значение $A_2$, вычисленное с помощью встроенных функций языка |
| n | int | Счётчик члена формулы Тейлора + кол-во итераций |

## Исходный код программы:

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

long double function(long double x){
    return 1/(4 - pow(x, 4));
}

int main(){
    const long double a = 0.0;
    const long double b = 1.0;

    int N;

    printf("Input N:");
    scanf("%d", &N);
    printf("N = %d\n", N);
    printf("Machine epsilon is equals 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 step = (b - a) / (long double) N;
    long double taylor, sum;

    int iter = 0;

    for (long double x = a + step; x < b + step; x += step){
        for (int n = 0; n < 100; ++n) {
            taylor = (pow(x, 4 * n)/(pow(4, n + 1)));
            sum += taylor;
            if (fabsl(sum - function(x)) < LDBL_EPSILON || iter > 100) {
```

```
                break;
            }
        }
        iter += 1;
        printf("|%.3Lf|%.20Lf|%.19Lf|          %d          |\n", x, sum, function(x), iter);
        sum = 0;
    }


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:

3

Вывод:

```
Input N:3
N = 3
Machine epsilon is equals 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 |
__________________________________________________________________________
|0.333|0.25077399380804953546|0.2507739938080495445|          1           |
|0.667|0.26298701298701298426|0.2629870129870129913|          2           |
|1.000|0.33333333333333331483|0.3333333333333333148|          3           |
__________________________________________________________________________
```

# Тест №2

Ввод:

100

Вывод:

```
Input N:100
N = 100
Machine epsilon is equals 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 |
|---|---|---|---|
| 0.010 | 0.2500000062500000157 | 0.2500000006249999962 | 1 |
| 0.020 | 0.2500000100000039998 | 0.2500000100000003833 | 2 |
| 0.030 | 0.2500000506250102515 6 | 0.2500000506250102394 | 3 |
| 0.040 | 0.2500001600001024000 5 | 0.2500001600001023894 | 4 |
| 0.050 | 0.2500003906256103525 2 | 0.2500003906256103581 | 5 |
| 0.060 | 0.2500008100026244084 9 | 0.2500008100026244184 | 6 |
| 0.070 | 0.2500015006340075556 4 | 0.2500015006340075563 | 7 |
| 0.080 | 0.2500025600262146684 6 | 0.2500025600262146819 | 8 |
| 0.090 | 0.2500041006922616048 0 | 0.2500041006922615883 | 9 |
| 0.100 | 0.2500062501562539063 4 | 0.2500062501562539397 | 10 |
| 0.110 | 0.2500091509599480114 9 | 0.2500091509599480233 | 11 |

## Тест №3

Ввод:

553

Вывод:

```
Input N:553
N = 553
Machine epsilon is equals 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 |
|---|---|---|---|
| 0.002 | 0.2500000000000066831236 | 0.2500000000000006682987 | 1 |
| 0.004 | 0.2500000000010692997 73 | 0.2500000000106930020 | 2 |
| 0.005 | 0.2500000000541333010 3 | 0.2500000000541333089 | 3 |
| 0.007 | 0.2500000001710879638 4 | 0.2500000001710879771 | 4 |
| 0.009 | 0.2500000004176952246 8 | 0.2500000004176952118 | 5 |
| 0.011 | 0.2500000008661328194 4 | 0.2500000008661328321 | 6 |
| 0.013 | 0.2500000016046179827 3 | 0.2500000016046179940 | 7 |
| 0.014 | 0.2500000027374074498 3 | 0.2500000027374074674 | 8 |
| 0.016 | 0.2500000043847974601 0 | 0.2500000043847974696 | 9 |
| 0.018 | 0.2500000066831237622 7 | 0.2500000066831237766 | 10 |
| 0.020 | 0.2500000097847616217 4 | 0.2500000097847616121 | 11 |

## 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