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 «Вычислительные системы»
I семестр
Задание 3
«Вещественный тип. Приближенные вычисления. Табулирование функций»

Группа	М8О-109Б-22
Студент	Дударь Ю.М.
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 9:

Ряд Тэйлора:

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

Функция:

$$\cos x$$

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 программе переменные

Название переменной	Тип переменной	Смысл переменной
n	int	Кол-во разбиений отрезка
iterat	int	Глубина ряда Тейлора
tal_ans	double	Сумма ряда Тейлора
tal_ch	double	Член ряда Тейлора
func	double	Значение функции
a	double	Левая граница отрезка
b	double	Правая граница отрезка
x	double	Шаг
eps	long double	Вычисление машинного епсилон

Исходный код программы:

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

//функция факториала
long double fac_num (long double n)
{
  long double fa;
  for (fa = 1; n > 1; fa *= (n--));
  return fa;
}

int main()
{
    int n, iterat;
    double tal_ans, func, tal_ch, a = 0.0, b = 1.0, x = 0.0;
    long double eps = 1.0l;
    //Вычисляем машинный епсилон
    while (2.0l + eps / 2.0l > 2.0l) {
        eps /= 2.0l;
    }
    printf("Machine eps double = %.16Le\n", eps);
    printf("Write n: \n");
    scanf("%d", &n);
    printf("n = %d, \n", n);
    printf("Table of Teylor values and stand f(x) = cos(x)\n");

printf("__________________________________________________________________________
___\n");
    printf("|   x   |         sum          |        f(x)          |count iter |\n");

printf("__________________________________________________________________________
___\n");
    for (int i = 1; i <= n; i++) {
        iterat = 1;
        tal_ch = 1;
        func = cos(x);
        tal_ans = 1;
        while (fabs(tal_ch) > eps && iterat < 100) {
            tal_ch = pow(-1, iterat)*(pow(x, 2*iterat))/fac_num(2*iterat);
            tal_ans += tal_ch;
            iterat++;
        }
        printf("| %.3f | %.18lf | %.18lf |      %d       |\n", x, tal_ans, func, iterat);
```

```
printf("______________________________________________________________________
___\n");
        x += fabs(a - b) / n;
    }
    return 0;
}
```

Входные данные

Единственная строка содержит два целых числа N (0≤N≤100) – число разбиений отрезка на равные части, K (0≤K≤16) — коэффициент для вычисления точности формулы Тейлора.

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 K знаков после запятой.

Протокол исполнения и тесты

Тест №1

Ввод:

5

Вывод:

```
Machine eps double = 2.16840434497100889e-19
Write n:
5
n = 5,
Table of Teylor values and stand f(x) = cos(x)

____________________________________________________________________________
|   x   |            sum            |          f(x)             |count iter |
____________________________________________________________________________
| 0.000 | 1.000000000000000000 | 1.000000000000000000 |       2       |
____________________________________________________________________________
| 0.200 | 0.980066577841241626 | 0.980066577841241626 |       8       |
____________________________________________________________________________
| 0.400 | 0.921060994002884992 | 0.921060994002885103 |       9       |
____________________________________________________________________________
| 0.600 | 0.825335614909678217 | 0.825335614909678217 |       10       |
____________________________________________________________________________
| 0.800 | 0.696706709347165387 | 0.696706709347165387 |       11       |
____________________________________________________________________________
```

Тест №2

Ввод:

10

Вывод:

```
Machine eps double = 2.1684043449710089e-19
Write n:
10
n = 10,
Table of Teylor values and stand f(x) = cos(x)

|    x    |          sum          |         f(x)          |count iter |

| 0.000 | 1.0000000000000000000 | 1.0000000000000000000 |     2      |

| 0.100 | 0.9950041652780258821 | 0.9950041652780258821 |     7      |

| 0.200 | 0.9800665778412416226 | 0.9800665778412416226 |     8      |

| 0.300 | 0.9553364891256059981 | 0.9553364891256059981 |     9      |

| 0.400 | 0.9210609940028849992 | 0.9210609940028851003 |     9      |

| 0.500 | 0.8775825618903727599 | 0.8775825618903727599 |     10      |

| 0.600 | 0.8253356149096783288 | 0.8253356149096783288 |     10      |

| 0.700 | 0.7648421872844883844 | 0.7648421872844884995 |     11      |

| 0.800 | 0.6967067093471654988 | 0.6967067093471654988 |     11      |

| 0.900 | 0.6216099682706646115 | 0.6216099682706645004 |     11      |
```

Тест №3

Ввод:

15

Вывод:

```
Machine eps double = 2.1684043449710089e-19
Write n:
15
n = 15,
Table of Teylor values and stand f(x) = cos(x)

|    x    |             sum             |           f(x)             |count iter |

| 0.000 | 1.0000000000000000000 | 1.0000000000000000000 |      2      |

| 0.067 | 0.9977786007011222231 | 0.9977786007011222342 |      7      |

| 0.133 | 0.9911242720341794410 | 0.9911242720341794410 |      7      |

| 0.200 | 0.9800665778412416 26 | 0.9800665778412416 26 |      8      |

| 0.267 | 0.9646546452305638 79 | 0.9646546452305638 79 |      8      |

| 0.333 | 0.9449569463147377 00 | 0.9449569463147377 00 |      9      |

| 0.400 | 0.9210609940028851 03 | 0.9210609940028851 03 |      9      |

| 0.467 | 0.8930729531984293 87 | 0.8930729531984293 87 |      10      |

| 0.533 | 0.8611171691298101 78 | 0.8611171691298102 89 |      10      |

| 0.600 | 0.8253356149096783 28 | 0.8253356149096783 28 |      10      |

| 0.667 | 0.7858872607769480 38 | 0.7858872607769480 38 |      10      |

| 0.733 | 0.7429473678240441 36 | 0.7429473678240441 36 |      11      |

| 0.800 | 0.6967067093471654 98 | 0.6967067093471654 98 |      11      |

| 0.867 | 0.6473707232789526 20 | 0.6473707232789525 09 |      11      |

| 0.933 | 0.5951585994691279 69 | 0.5951585994691278 58 |      11      |
```

Тест №4

Ввод:

3

Вывод:

```
Machine eps double = 2.1684043449710089e-19
Write n:
3
n = 3,
Table of Teylor values and stand f(x) = cos(x)
_______________________________________________________________________________
|   x   |           sum            |          f(x)          |count iter |
_______________________________________________________________________________
| 0.000 | 1.000000000000000000 | 1.000000000000000000 |      2        |
_______________________________________________________________________________
| 0.333 | 0.944956946314737700 | 0.944956946314737700 |      9        |
_______________________________________________________________________________
| 0.667 | 0.785887260776948038 | 0.785887260776948038 |      10        |
_______________________________________________________________________________
```

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