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	Филиппов А. 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 11:

Ряд Тэйлора:

$$1-\frac{3}{2}x^2+\ldots+(-1)^n\frac{2n^2+1}{(2n)!}x^{2n}$$

Функция:

$$\left(1-\frac{x^2}{2}\right)\cos x-\frac{x}{2}\sin x$$

Значения a и b: 0.1 и 0.6

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.
currentX	long double	Переменная, для которой будем производить вычисления
getTaylorSeries (currentX, i)	double	То самое значение A_1, вычисленное с помощью формулы Тейлора
func(currentX)	double	То самое значение A_2, вычисленное с помощью встроенных функций языка
i	double	Счётчик члена формулы Тейлора + кол-во итераций

Исходный код программы:

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

int64_t factorial(int64_t n) {
    int64_t res = 1;

    for (int64_t i = 1; i <= n; ++i)
        res *= i;

    return res;
}

long double func(long double x) {
    return (1 - x*x/2) * cosl(x) - x / 2 * sinl(x);
}

long double getTaylorSeries(long double x, int64_t n) {
    long double res = 0;
    for (int64_t i = 0; i <= n; ++i)
        res += pow(-1, i) * (((long double) (2 * i * i + 1)) / ((long
double) factorial(2 * i))) * pow(x, (long double) 2 *i);

    return res;
}

int main() {
    long double sum = 0.0;
    long double a = 0.1;
    long double b = 0.6;

    int64_t n;
    scanf_s("%lld", &n);
    printf("N = %lld\n", n);
    printf("Machine epsilon is equal to: %g\n", LDBL_EPSILON);

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

printf("____________________________________________________________________
_____\n");
    printf("|   x   |           sum         |         f(x)         |number of
iterations |\n");

printf("____________________________________________________________________
_____\n");

    long double currentX;
    long double step = (b - a) / (long double) n;
    for (int64_t i = 1; i <= n; ++i) {
        currentX = a + step * (long double) i;
        sum = getTaylorSeries(currentX, i);

        printf("| %.3Lf | %.16Lf | %.16Lf |       %lld        |\n", currentX,
sum, func(currentX), i);

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

```
        if (fabsl(func(currentX) - sum) < LDBL_EPSILON) break;
    }

    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:

```
2
N = 2
Machine epsilon is equal to: 3.50172e-312
Table for values of Taylor series and of base function
-----------------------------------------------------------------------
|   x   |          sum        |        f(x)        |number of iterations |
-----------------------------------------------------------------------
| 0.350 | 0.8162500000000000 | 0.8218290178807730 |      1       |
-----------------------------------------------------------------------
| 0.600 | 0.5086000000000000 | 0.5073824622074256 |      2       |
-----------------------------------------------------------------------

Process finished with exit code 0
```

Тест №2

Ввод:
200

Вывод:

```
200
N = 200
Machine epsilon is equal to: 4.2329e-312
Table for values of Taylor series and of base function
-------------------------------------------------------------------------
|   x   |          sum          |          f(x)         |number of iterations |
-------------------------------------------------------------------------
| 0.103 | 0.9842406250000000 | 0.9842819873903479 |      1        |
-------------------------------------------------------------------------
| 0.105 | 0.9835080814843750 | 0.9835080461328300 |      2        |
-------------------------------------------------------------------------
| 0.108 | 0.9827156643667610 | 0.9827156643813550 |      3        |
-------------------------------------------------------------------------
| 0.110 | 0.9819048570180179 | 0.9819048570180142 |      4        |
-------------------------------------------------------------------------
| 0.113 | 0.9810756392721796 | 0.9810756392721796 |      5        |
-------------------------------------------------------------------------
| 0.115 | 0.9802280267203024 | 0.9802280267203024 |      6        |
-------------------------------------------------------------------------

Process finished with exit code 0
```

Тест №3

Ввод:

100000

Вывод:

```
100000
N = 100000
Machine epsilon is equal to: 1.40416e-312
Table for values of Taylor series and of base function
------------------------------------------------------------------------
|   x   |          sum        |         f(x)        |number of iterations |
------------------------------------------------------------------------
| 0.100 | 0.9849984999625000 | 0.9850359810744424 |       1       |
------------------------------------------------------------------------
| 0.100 | 0.9850345148522501 | 0.9850344884557136 |       2       |
------------------------------------------------------------------------
| 0.100 | 0.9850329957549152 | 0.9850329957631082 |       3       |
------------------------------------------------------------------------
| 0.100 | 0.9850315029966274 | 0.9850315029966260 |       4       |
------------------------------------------------------------------------
| 0.100 | 0.9850300101562674 | 0.9850300101562674 |       5       |
------------------------------------------------------------------------
| 0.100 | 0.9850285172420323 | 0.9850285172420323 |       6       |
------------------------------------------------------------------------
| 0.100 | 0.9850270242539209 | 0.9850270242539209 |       7       |
------------------------------------------------------------------------

Process finished with exit code 0
```

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