МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский Авиационный Институт»

(Национальный Исследовательский Университет)

Институт: №8 «Компьютерные науки 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	Недосекин А.А.
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 8:**

Ряд Тэйлора:

$$-\frac{1}{5}-\frac{2x}{5^2}-\frac{4x^2}{5^3}-\ldots-\frac{2^{n-1}x^{n-1}}{5^n}$$

Функция:

$$\frac{1}{2x-5}$$

Значения a и b: 0.0 и 2.0

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	То самое число N, на которое нужно разбить отрезок
i	int	Счетчик числа итераций
LDBL_EPSILON	Long double	машинное эпсилон 1.0842e-19
step	Long double	Разница между текущим и предыдущем значениями переменной
x	Long double	Переменная, для которой производятся вычисления
taylor(int n, long double x)	Long double	Значение ряда Тейлора для функции
function(long double x)	Long double	Значение функции

Исходный код программы:

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 K знаков после запятой.

Протокол исполнения

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

long double function(long double x){
return (1 / (2*x - 5));
}

int main(){     long double a =
0.0;       long double b = 2.0;
int n;     printf("Input n please
: \n");     scanf("%d", &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 taylor, sum;     int i = 0;    for (long double x
```

```
= a + step ; x < b + step ; x += step){         for (int n = 1; n
< 100; n++) {             taylor = powl(2, (n - 1)) * powl(x, (n
- 1)) / powl(5, n);
            sum += taylor;             if (fabsl(sum - function(x)) <
LDBL_EPSILON || i > 100) {                 break;
            }         }         i += 1;
printf("|%.3Lf|%.20Lf|%.19Lf|           %d          |\n", x,
sum, function(x), i - 1);
        sum = 0;
    }
printf("____________________________________________________
_________________________________________\n");

    return 0;
}
```

RESULT

If n = 7 :

```
Input n please :
```

```
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|

______________________________________________________________________________

|0.286|0.22580645161290322579|-0.2258064516129032258|          0          |
|0.571|0.25925925925925925918|-0.2592592592592592593|          1          |
|0.857|0.30434782608695652175|-0.3043478260869565217|          2          |
|1.143|0.36842105263157894730|-0.3684210526315789474|          3          |
|1.429|0.46666666666666666645|-0.4666666666666666666|          4          |
|1.714|0.63636363636363632514|-0.6363636363636363636|          5          |
|2.000|0.99999999974537050266|-0.9999999999999999998|          6          |




______________________________________________________________________________
```

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