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 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 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 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
f	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>

long double Taylor(uint64_t n, long double x) {
   long double res = 0;
   for (int i = 0; i < n; ++i) {
      res += (pow(x, 4 * i)/(pow(4, i + 1)));
   }
   return res;
}

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

int main() {

   long double a = 0.0;
   long double b = 1.0;

   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 f = 1;
  int i = 0;

  while (fabsl(f) > LDBL_EPSILON && (i < 100) && (i < n)) {
    i += 1;
    x += step;
    f = function(x);

    printf("|%.5llf|%.20llf|%.20llf|        %d        |\n", x, Taylor(i, x), f, i);
  }


printf("_________________________________________________________________________________\n");

  return 0;
}
```

Входные данные

Единственная строка содержит одно целое число N ($0 \leq N \leq 100$) – 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:

Table of values of Taylor series and standard function

x	sum of Taylor series	f(x) function value	number of iterations
0.33333	0.25000000000000000000	0.25077399380804954454	1
0.66667	0.26234567901234567659	0.26298701298701299134	2
1.00000	0.32812500000000000000	0.33333333333333331483	3

...Program finished with exit code 0

Тест №2

Ввод:
100

Вывод:

Table of values of Taylor series and standard function

x	sum of Taylor series	f(x) function value	number of iterations
0.01000	0.25000000000000000000	0.25000000062499999620	1
0.02000	0.25000000999999999999	0.25000001000000038331	2
0.03000	0.25000005062501025156	0.25000005062501023945	3
0.04000	0.25000016000010240005	0.25000016000010238937	4
0.05000	0.25000039062561035252	0.25000039062561035808	5
0.06000	0.25000081000262440849	0.25000081000262441844	6
0.07000	0.25000150063400755564	0.25000150063400755629	7
0.08000	0.25000256002621466846	0.25000256002621468188	8
0.09000	0.25000410069226160480	0.25000410069226158827	9
0.10000	0.25000625015625390634	0.25000625015625393965	10
0.11000	0.25000915095994801149	0.25000915095994802329	11

Тест №3

Ввод:

553793

Вывод:

Table of values of Taylor series and standard function

x	sum of Taylor series	f(x) function value	number of iterations
0.00000	0.25000000000000000000	0.25000000000000000000	1
0.00000	0.25000000000000000000	0.25000000000000000000	2
0.00001	0.25000000000000000000	0.25000000000000000000	3
0.00001	0.25000000000000000000	0.25000000000000000000	4
0.00001	0.25000000000000000000	0.25000000000000000000	5
0.00001	0.25000000000000000000	0.25000000000000000000	6
0.00001	0.25000000000000000000	0.25000000000000000000	7
0.00001	0.25000000000000000000	0.25000000000000000000	8
0.00002	0.25000000000000000000	0.25000000000000000000	9
0.00002	0.25000000000000000000	0.25000000000000000000	10
0.00002	0.25000000000000000000	0.25000000000000000000	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