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	Юсуфов Р.Г.
Преподаватель	Сысоев М. А.
Оценка	
Дата	29 декабря 2022 г.

Москва, 2022

Задание

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 ε * 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 обеспечивать корректные размеры генерируемой таблицы.

Вариант 16:

16	$1+3x^2+\ldots+\frac{2n+1}{n!}x^{2n}$	0.0	1.0	$(1+2x^2)e^{x^2}$

Введение:

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 + ε = 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

Алгоритм программы и план работы

Пока новые члены ряда Тейлора больше или равны ε*k, суммируем члены ряда Тейлора для каждой строки таблицы, прибавляя к переменной общей суммы новые члены ряда Тейлора.

Для написания программы используется библиотека math.h. Для корректной компиляции кода, в компиляторе gcc, после стандартной команды для компиляции, необходимо написать -lm.

Также, помимо стандартной библиотеки stdio.h, используются библиотеки assert.h и float.h

Программа запрашивает два числа, n количество частей отрезка, k коэффициент приемлемой сходимости. Учтено требование к знаку n, так как n > 0.

На выход программа выдаёт требуемую таблицу значений.

В программе использовал переменные, представленные в таблице:

Название переменной	**Тип переменной**	**Смысл переменной**
a	long double	Начало отрезка
b	long double	Конец отрезка
n	int	Количество частей отрезка

x	long double	Значения в промежутке [a;b], для которого вычисляются значения
pl	long double	Значение, прибавляемое к x на каждом шаге
k	long double	Коэффициент, обеспечивающий приемлемую сходимость
t	long double	Текущий член ряда Тейлора
summ	long double	Переменная суммы членов ряда Тейлора
c	int	Количество итераций вычисления
LDBL_EPSILON	long double	Машинный эпсилон. Для long double ε = $1.08 * 10^{-19}$

Распечатка кода программы:

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

long double f(long double x) {
    return (1 + 2*(powl(x,2)))*(expl(powl(x, 2)));
}

long double fact(long double n) {
    long double res = 1;
    for (long double i = 1; i <= n; i++) res *= i;
    return res;
}

void tablica() {
    printf("Таблица значений ряда Тейлора для функции f(x) = (1 +
2*(x^2))*e^(x^2)\n");
    printf("
-----------------------------------------------------------------------------------------\n");
    printf("| x \t| Ряд Тейлора\t\t | Функция\t\t | Итерации | Разница\t\t |\n");
```

```
    printf("
-------+-----------------------+-----------------------+-------+------------------------\n");
}

void str(long double x, long double summ, int n) {
    if (summ < 0) {
    printf("| %.2Lf\t| %.19Lf | %.19Lf | %d\t  | %.19Lf |\n", x, summ, f(x), n, fabsl(f(x) -
summ));
    } else {
    printf("| %.2Lf\t| %.19Lf  | %.19Lf  | %d\t  | %.19Lf |\n", x, summ, f(x), n, fabsl(f(x)
- summm));
    }
}


int main() {
    const long double a = 0, b = 1;
    long double summ, t, k;
    int c = 0;
    int n;

    printf("Количество итераций(n): \n");
    scanf("%d", &n);
    assert((n > 0) && "n должно быть больше нуля!");
    printf("Коэффицент(k): ");
    scanf("%Lf", &k);
    printf("\n\n");
    tablica();

    long double pl = (b - a) / n;
    for (long double x = a; x <= b; x += pl) {
    for (int i = 0; i < 99; ++i) {
            t = ((2*i + 1)/(fact(i)))*powl(x, 2*i);
            summ += t;
            ++c;
            if (fabsl(summ - f(x)) < LDBL_EPSILON * k) {
            break;
            }
    }
    str(x, summ, c);
    summ = 0;
    c = 0;
```

```
    }
}
```

Тестирование программы

Тест №1

Входные данные:

6

1

Таблица значений ряда Тейлора для функции f(x) = (1 + 2*(x^2))*e^(x^2)

x	Ряд Тейлора	Функция	Итерации	Разница
0.00	1.0000000000000000000	1.0000000000000000000	1	0.0000000000000000000
0.17	1.0852875761842540760	1.0852875761842540760	10	0.0000000000000000000
0.33	1.3658566395733889039	1.3658566395733889038	99	0.0000000000000000001
0.50	1.9260381250316122260	1.9260381250316122260	15	0.0000000000000000000
0.67	2.9459554954794746854	2.9459554954794746852	99	0.0000000000000000002
0.83	4.7839798383218441035	4.7839798383218441022	99	0.0000000000000000013
1.00	8.1548454853771357043	8.1548454853771357061	99	0.0000000000000000017

Выходные данные:

Тест №2

Входные данные:

8

2

Таблица значений ряда Тейлора для функции f(x) = (1 + 2*(x^2))*e^(x^2)

x	Ряд Тейлора	Функция	Итерации	Разница
0.00	1.0000000000000000000	1.0000000000000000000	1	0.0000000000000000000
0.12	1.0474898244800196771	1.0474898244800196770	9	0.0000000000000000001
0.25	1.1975562662825918582	1.1975562662825918582	11	0.0000000000000000000
0.38	1.4747097103855698256	1.4747097103855698257	13	0.0000000000000000001
0.50	1.9260381250316122260	1.9260381250316122260	15	0.0000000000000000000
0.62	2.6325168480771590745	2.6325168480771590745	17	0.0000000000000000000
0.75	3.7294911460406344342	3.7294911460406344342	18	0.0000000000000000000
0.88	5.4430428497542592626	5.4430428497542592621	99	0.0000000000000000004
1.00	8.1548454853771357043	8.1548454853771357061	99	0.0000000000000000017

Выходные данные:

Тест №3

Входные данные:

10

3

Выходные данные:

Таблица значений ряда Тейлора для функции f(x) = (1 + 2*(x^2))*e^(x^2)

x	Ряд Тейлора	Функция	Итерации	Разница
0.00	1.0000000000000000000	1.0000000000000000000	1	0.0000000000000000000
0.10	1.0302511704258514187	1.0302511704258514187	8	0.0000000000000000000
0.20	1.1240756361277792850	1.1240756361277792849	10	0.0000000000000000001
0.30	1.2911256547721482223	1.2911256547721482224	12	0.0000000000000000001
0.40	1.5490343497091895101	1.5490343497091895104	14	0.0000000000000000002
0.50	1.9260381250316122260	1.9260381250316122260	15	0.0000000000000000000
0.60	2.4653265930437852432	2.4653265930437852434	17	0.0000000000000000002
0.70	3.2319861155116503613	3.2319861155116503613	18	0.0000000000000000000
0.80	4.3239764048152890858	4.3239764048152890862	99	0.0000000000000000004
0.90	5.8895189250923551199	5.8895189250923551199	20	0.0000000000000000000

Заключение

Для выполнения задания, была теория по машинному эпсилону, формуле Тейлора, была составлена таблица значений и сравнений вычислений по формуле Тейлора и при помощи встроенных функций

Вычисление значений функции по формуле Тейлора довольно ресурсоемкое и имеет погрешность.