

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 $\varepsilon * k$, 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	$1 + 3x^2 + \dots + \frac{2n+1}{n!} x^{2n}$	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.

$$\sum_{n=0}^k \frac{f^{(n)}(a)}{n!} (x-a)^n = f(a) + f^{(1)}(a)(x-a) + \frac{f^{(2)}(a)}{2!} (x-a)^2 + \dots + \frac{f^{(k)}(a)}{k!} (x-a)^k$$

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 – $1.19 \cdot 10^{-7}$, double – $2.20 \cdot 10^{-16}$, long double – $1.08 \cdot 10^{-19}$.

Алгоритм программы и план работы

Пока новые члены ряда Тейлора больше или равны $\varepsilon \cdot k$, 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 $\varepsilon = 1.08 * 10^{-19}$

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)));
}
```

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

```

```

int main() {
    const long double a = -1, 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("Коэффициент(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 n = 0; n < 99; ++n) {
            t = powl(-1, n) * powl(x, 2 * n + 1) / powl(9, n + 1);
            summ += t;
            ++c;
            if (fabsl(summ - f(x)) < LDBL_EPSILON * k) {
                break;
            }
        }
        str(x, summ, c);
        summ = 0;
        c = 0;
    }

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

```

Тестирование программы

Тест №1

Входные данные:

6

1

Выходные данные:

Таблица значений ряда Тейлора для функции $f(x) = (1 + 2*(x^2))*e^{(x^2)}$					
x	Ряд Тейлора	Функция	Итерации	Разница	
-1.00	-0.10000000000000000000	8.1548454853771357061	99	8.2548454853771357064	
-0.67	-0.0705882352941176470	2.9459554954794746845	99	3.0165437307735923315	
-0.33	-0.0365853658536585366	1.3658566395733889037	99	1.4024420054270474403	
0.00	0.00000000000000000000	1.00000000000000000000	99	1.0000000000000000000	
0.33	0.0365853658536585366	1.3658566395733889040	99	1.3292712737197303675	
0.67	0.0705882352941176471	2.9459554954794746854	99	2.8753672601853570382	

Тест №2

Входные данные:

9

3

Выходные данные:

Таблица значений ряда Тейлора для функции $f(x) = (1 + 2*(x^2))*e^{(x^2)}$					
x	Ряд Тейлора	Функция	Итерации	Разница	
-1.00	-0.10000000000000000000	8.1548454853771357061	99	8.2548454853771357064	
-0.78	-0.0809768637532133676	4.0465915038092286110	99	4.1275683675624419786	
-0.56	-0.0596816976127320955	2.2020530810319509224	99	2.2617347786446830179	
-0.33	-0.0365853658536585366	1.3658566395733889037	99	1.4024420054270474403	
-0.11	-0.0123287671232876712	1.0374202805397057955	99	1.0497490476629934667	
0.11	0.0123287671232876712	1.0374202805397057957	99	1.0250915134164181245	
0.33	0.0365853658536585366	1.3658566395733889038	99	1.3292712737197303673	
0.56	0.0596816976127320955	2.2020530810319509233	99	2.1423713834192188278	
0.78	0.0809768637532133676	4.0465915038092286123	99	3.9656146400560152447	

Тест №3

Входные данные:

20

3

Выходные данные:

Таблица значений ряда Тейлора для функции $f(x) = (1 + 2*(x^2))*e^{(x^2)}$					
x	Ряд Тейлора	Функция	Итерации	Разница	
-1.00	-0.10000000000000000000	8.1548454853771357061	99	8.2548454853771357064	
-0.90	-0.0917431192660550459	5.8895189250923551177	99	5.9812620443584101635	
-0.80	-0.0829875518672199170	4.3239764048152890849	99	4.4069639566825090020	
-0.70	-0.0737618545837723920	3.2319861155116503602	99	3.3057479700954227522	
-0.60	-0.0641025641025641026	2.4653265930437852430	99	2.5294291571463493455	
-0.50	-0.0540540540540540540	1.9260381250316122256	99	1.9800921790856662796	
-0.40	-0.0436681222707423581	1.5490343497091895100	99	1.5927024719799318681	
-0.30	-0.0330033003300330033	1.2911256547721482222	99	1.3241289551021812255	
-0.20	-0.0221238938053097345	1.1240756361277792848	99	1.1461995299330890193	
-0.10	-0.0110987791342952275	1.0302511704258514187	99	1.0413499495601466462	
0.00	0.0000000000000000000	1.0000000000000000000	99	1.0000000000000000000	
0.10	0.0110987791342952275	1.0302511704258514187	99	1.0191523912915561911	
0.20	0.0221238938053097345	1.1240756361277792850	99	1.1019517423224695504	
0.30	0.0330033003300330033	1.2911256547721482224	99	1.2581223544421152191	
0.40	0.0436681222707423581	1.5490343497091895105	99	1.5053662274384471524	
0.50	0.0540540540540540541	1.9260381250316122263	99	1.8719840709775581723	
0.60	0.0641025641025641026	2.4653265930437852440	99	2.4012240289412211415	
0.70	0.0737618545837723920	3.2319861155116503617	99	3.1582242609278779698	
0.80	0.0829875518672199170	4.3239764048152890871	99	4.2409888529480691700	
0.90	0.0917431192660550459	5.8895189250923551212	99	5.7977758058263000754	

Заклучение

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 большую погрешность.