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 «Фундаментаментальная информатика» I семестр
Задание №3 «Вещественный тип. Приближенные вычисления.Табулирование функций »

Группа	М8О-109Б-22
Студент	Потехин Ф.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 задания 2:

Исходная функция: Ряд Тейлора:

$$\ln\frac{1+x}{1-x}$$

$$2\left(\frac{x}{1}+\frac{x^3}{3}+\ldots+\frac{x^{2n+1}}{2n+1}\right)$$

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, причём почти без погрешности. 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 нашем же Машинное эпсилон понадобится как критерий точности значения ряда Тейлора и исходной функции. То есть мы будем осуществлять новую итерацию в ряде Тейлора до тех пор, пока разница значения исходной функции и ряда будем меньше, чем машинное эпсилон (или пока число итераций не дойдёт до 100, что следует из
условия). В языке Си машинные эпсилон определено для следующих типов: float – $1.19 * 10^{-7}$, double – $2.20 * 10^{-16}$, long double – $1.08 * 10^{-19}$. Мы будем вычислять машинное эпсилон самостоятельно.

Описание алгоритма

Рассмотрим алгоритм решения. Сперва нужно найти машинное эпсилон, на котором будет основываться точность вычисления. Это можно сделать путём деления 1 на 2 до тех пор, пока не будет выполнено 1 + ε = 1. Все переменные должны быть одного типа с машинным эпсилон. Для каждой N+1 строки нужно просуммировать i членов формулы Тейлора, пока $|A_1-A_2|> ε$ или пока i <= 100. 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	Количество разбиений отрезка
count	int	Количество итераций при сложении ряда Тейлора
Taylor	double	Сумма ряда Тейлора
fuction	double	Значение функции
eps	double	Значение машинного эпсилон
left	double	Левая граница отрезка
right	double	Правая граница отрезка
x	double	Текущий икс

Исходный код программы:

```c
int main () {
    int n, count = 1;
    long double Taylor, fuction, left = 0.0, right = 0.5, x = left, eps = 1.0;
    while (1.0L + eps > 1.0L) {
        eps /= 2.0L;
    }
    printf("Machine epsilon for long double is %.16Le\n", eps);
    printf("Write n: ");
    scanf("%d", &n);
    assert(n >= 0);
    printf("Your n is %d\n", n);
    printf("Table of values for Taylor's Formula and f(x) = ln((1 + x) / (1 - x))\n");
    printf("|-------------------------------------------------------------------|\n");
    printf("|    x   |        Taylor        |          f(x)        | iterations |\n");
    printf("|-------------------------------------------------------------------|\n");
    for (int i = 1; i <= n + 1; i++) {
        Taylor = 2 * x;
        fuction = logl((1 + x) / (1 - x));
        count = 1;
        while (fabsl(Taylor - fuction) > eps && count <= 100) {
            Taylor += 2 * ((powl(x, (count * 2) + 1)) / ((count * 2) + 1));
            count++;
        }
        count--;
        if (count >= 0 && count <=9) {
            printf("| %.4Lf | %.18Lf | %.18Lf |     %d      |\n", x, Taylor, fuction, count);
        }
        if (count >= 10 && count <= 99) {
            printf("| %.4Lf | %.18Lf | %.18Lf |     %d     |\n", x, Taylor, fuction, count);
        }
        if (count == 100) {
            printf("| %.4Lf | %.18Lf | %.18Lf |     %d    |\n", x, Taylor, fuction, count);
        }
        printf("|-------------------------------------------------------------------|\n");
        x += (right - left) / n;
    }

    return 0;
}
```

Входные данные

Единственное входное число - количество разбиений отрезка.

Выходные данные:

Программа должна вывести значение машинного эпсилон, введёное ранее количество разбиений, а затем N+1 строку.

В каждой строке должно быть значение x, для которого вычисляется функция, значение по формуле Тейлора, значение функции, вычисленное с помощью встроенных функций языка, количество итерация, требуемых для вычисления.

Протокол исполнения и тесты

Тест №1

Ввод:

10

Вывод:

```
Machine epsilon for long double is 5.4210108624275222e-20
Write n: 10
Your n is 10
Table of values for Taylor's Formula and f(x) = ln((1 + x) / (1 - x))
|-----------------------------------------------------------------------|
|    x   |         Taylor          |           f(x)          | iterations |
|-----------------------------------------------------------------------|
| 0.0000 | 0.000000000000000000 | 0.000000000000000000 |      0      |
|-----------------------------------------------------------------------|
| 0.0500 | 0.100083458556982536 | 0.100083458556982537 |      6      |
|-----------------------------------------------------------------------|
| 0.1000 | 0.200670695462151161 | 0.200670695462151161 |      8      |
|-----------------------------------------------------------------------|
| 0.1500 | 0.302280871872933611 | 0.302280871872933611 |      10     |
|-----------------------------------------------------------------------|
| 0.2000 | 0.405465108108164382 | 0.405465108108164382 |      12     |
|-----------------------------------------------------------------------|
| 0.2500 | 0.510825623765990683 | 0.510825623765990683 |      13     |
|-----------------------------------------------------------------------|
| 0.3000 | 0.619039208406223431 | 0.619039208406223431 |      16     |
|-----------------------------------------------------------------------|
| 0.3500 | 0.730887508542792338 | 0.730887508542792338 |      100    |
|-----------------------------------------------------------------------|
| 0.4000 | 0.847297860387203614 | 0.847297860387203614 |      21     |
|-----------------------------------------------------------------------|
| 0.4500 | 0.969400557188103483 | 0.969400557188103483 |      25     |
|-----------------------------------------------------------------------|
| 0.5000 | 1.098612288668109692 | 1.098612288668109692 |      28     |
|-----------------------------------------------------------------------|
```

Тест №2

Ввод: 5

Вывод:

```
Machine epsilon for long double is 5.4210108624275222e-20
Write n: 5
Your n is 5
Table of values for Taylor's Formula and f(x) = ln((1 + x) / (1 - x))
|-----------------------------------------------------------------------|
|    x   |         Taylor          |           f(x)          | iterations |
|-----------------------------------------------------------------------|
| 0.0000 | 0.000000000000000000 | 0.000000000000000000 |      0      |
|-----------------------------------------------------------------------|
| 0.1000 | 0.200670695462151161 | 0.200670695462151161 |      8      |
|-----------------------------------------------------------------------|
| 0.2000 | 0.405465108108164382 | 0.405465108108164382 |      12     |
|-----------------------------------------------------------------------|
| 0.3000 | 0.619039208406223431 | 0.619039208406223431 |      16     |
|-----------------------------------------------------------------------|
| 0.4000 | 0.847297860387203614 | 0.847297860387203614 |      21     |
|-----------------------------------------------------------------------|
| 0.5000 | 1.098612288668109692 | 1.098612288668109691 |      100    |
|-----------------------------------------------------------------------|
```

Тест №3

Ввод: 20

Вывод:

```
Machine epsilon for long double is 5.4210108624275222e-20
Write n: 20
Your n is 20
Table of values for Taylor's Formula and f(x) = ln((1 + x) / (1 - x))
```

x	Taylor	f(x)	iterations
0.0000	0.000000000000000000	0.000000000000000000	0
0.0250	0.050010420574661376	0.050010420574661376	5
0.0500	0.100083458556982536	0.100083458556982537	6
0.0750	0.150282203049337979	0.150282203049337979	7
0.1000	0.200670695462151161	0.200670695462151161	8
0.1250	0.251314428280906078	0.251314428280906078	9
0.1500	0.302280871872933611	0.302280871872933611	10
0.1750	0.353640040243578351	0.353640040243578351	11
0.2000	0.405465108108164382	0.405465108108164382	12
0.2250	0.457833093625480364	0.457833093625480364	13
0.2500	0.510825623765990683	0.510825623765990683	13
0.2750	0.564529802737851744	0.564529802737851745	100
0.3000	0.619039208406223431	0.619039208406223431	16
0.3250	0.674455047547792760	0.674455047547792760	17
0.3500	0.730887508542792338	0.730887508542792338	18
0.3750	0.788457360364270169	0.788457360364270170	100
0.4000	0.847297860387203614	0.847297860387203614	21
0.4250	0.907557051905400461	0.907557051905400461	22
0.4500	0.969400557188103483	0.969400557188103483	24
0.4750	1.033015006182296454	1.033015006182296454	100
0.5000	1.098612288668109692	1.098612288668109692	28

Вывод

В ходе выполнения КП была проделана теоретическая работа, а также был реализован алгоритм подсчёта функции с помощью Тейлора с предоставленным листингом и тестами. Данный опыт, несомненно, поможет мне в дальнейшей работе по реализации других математических алгоритмов и вычислений. В целом работа мне понравилась.

Список литературы

1. Машинный ноль – URL: https://ru.wikipedia.org/wiki/Машинный_ноль
2. Ряд Тейлора – URL: https://ru.wikipedia.org/wiki/Ряд_Тейлора