

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»**

Курсовой проект

по курсу фундаментальная информатика 1 семестра

Задание 3. Вещественный тип. Приближенные вычисления. Табулирование
функций

Студент: Калюжный М.С.

Группа: М8О-108Б-22

Преподаватель: Сахарин Н.А.

Подпись:

Оценка:

Москва 2022

СОДЕРЖАНИЕ

ЗАДАЧА.....	3
ОБЩИЙ МЕТОД РЕШЕНИЯ	4
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.....	5
ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ.....	6
ОПИСАНИЕ ПРОГРАММЫ	7
ПРОТОКОЛ.....	8
ВЫВОД.....	11
ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ.....	12

ЗАДАЧА

Составить программу на языке программирования Си, которая печатает таблицу значений элементарной функции, вычисленной двумя способами: по формуле Тейлора и с помощью функций из стандартной библиотеки языка Си. В качестве аргументов таблицы взять точки разбиения $[a, b]$ на n равных частей ($n+1$ точка включая концы отрезка), находящихся в рекомендованной области достаточной точности формулы Тейлора. Вычисления по формуле Тейлора проводить по экономной в сложностном смысле схеме с точностью $\varepsilon * k$, где ε – машинное эпсилон аппаратно реализованного 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.

11 вариант задания:

Отрезок - $[0.1, 0.6]$

Функция: $(1 - (x^2 / 2)) * \cos(x) - (x / 2) * \sin(x)$

Разложение в ряд: $1 - (3 / 2) * x^2 + \dots + (-1)^n * (2n^2 + 1) / (2n)! * x^{2n}$

За количество x -ов на отрезке $[0.0, 1.0]$ взято число 15.

ОБЩИЙ МЕТОД РЕШЕНИЯ

Общий метод решения заключается в нахождении значения функции в некоторой точке при помощи двух способов.

Первый способ заключается в использовании функций, имеющихся в стандартной библиотеке «math.h» языка Си.

$$f(x) = f(a) + \frac{f'(a)}{1!} \cdot (x - a) + \frac{f''(a)}{2!} \cdot (x - a)^2 + \dots + \frac{f^{(n)}(a)}{n!} \cdot (x - a)^n + \dots$$

Основополагающей вещью в вычислении данной функции является наличие, так называемого, машинного эпсилон, которое является критерием точности вычислений на заданной ЭВМ.

Машинное эпсилон — минимальное число, выразимое на конечной вычислительной машине.

Его можно найти путём сравнения $1 + \varepsilon$ с 1 ($1 + \varepsilon = 1$). Последнее число, при стремлении к нулю, при котором данное выражение выдаст false и будет машинным эпсилон.

Я буду вычислять на каждом шаге итерации n-ное слагаемое ряда Тейлора и, в случае если данное слагаемое будет меньше $k \cdot \varepsilon$ (где k — экспериментально подобранный коэффициент), то далее вычислять ряд Тейлора является бессмысленным, т.к. члены ряда дошли до максимальной точности компьютера.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

ОС семейства: UNIX

Наименование: Ubuntu

Версия: Ubuntu 22.04 LTS

Интерпретатор команд: bash

Версия: 5.1-6ubuntu1

Компилятор: gcc

ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Программа предназначена для выполнения вещественных вычислений значений трансцендентных функций в алгебраической форме с использованием ряда Тейлора.

Ряд Тейлора – это разложение функции в бесконечную сумму степенных функций. Если функция $f(x)$ имеет непрерывные производные до $(n + 1)$ порядка, то ее можно разложить по формуле Тейлора.

Ранее данный метод использовался для аппаратного вычисления подобных функций, так как в то время компьютеры были способны только на сложение, вычитание и умножение. Но на сегодняшний день аппаратное обеспечение позволяет вычислять трансцендентные функции другими способами, которые более эффективны во всех смыслах.

ОПИСАНИЕ ПРОГРАММЫ

Программа работы:

- Подключаем заголовки «math.h» и «stdio.h»
- Определяем функцию вычисления машинного эпсилон
- Определяем функцию для вычисления члена ряда Тейлора
- Определяем функцию для вычисления функции при помощи встроенных функций
- Вычисляем машинное эпсилон и выводим.
- Печатаем таблицу аргументов функций, значений полученных средствами языка C и ряда Тейлора, количество итераций запрошенное машиной для вычисления значения функции

ПРОТОКОЛ

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

#define MAX_ITER 20

typedef unsigned uint;
typedef double dbl;
typedef long double ldbl;

dbl epsilon();

dbl TaylorSeries(dbl x);

void tabulation(dbl a, dbl b, uint n);

int main() {
    dbl a = 0.1;
    dbl b = 0.6;
    dbl epss = epsilon();
    printf("dbl epsilon is %.20e\n", epss);
    tabulation(a, b, 20);
    return 0;
}

void tabulation(const dbl a, const dbl b, const uint n) {
    const dbl delta = (b - a) / n;
    for (uint i = 0; i <= n; ++i) {
```



```

const dbl x = a + i * delta;

    printf("%.6f %.15f %.15f\n", x, (1 - (x * x / 2)) * cos(x) - (x / 2) * sin(x),
TaylorSeries(x));

    }

}

```

```

dbl epsilon() {
    dbl x = 1.0;
    dbl eps = 1.0;
    while (x + eps / 2.0 != x)
        eps /= 2.0;
    return eps;
}

```

```

dbl TaylorSeries(const dbl x) {
    dbl result = 1.0, member = 1.0;
    for (uint n = 1; n < MAX_ITER; ++n) {
        member *= -1.0 * (2.0 * n * n + 1.0) / (4.0 * n * n * n - 10.0 * n * n +
10.0 * n - 3.0) * x * x / (2.0 * n);
        result += member;
    }
    return result;
}

```

```

mimik@mimik-VirtualBox:~$ gcc kp31.c -lm -Wall -pedantic -std=c99 -o c4.out && ./c4.out
dbl epsilon is 2.22044604925031308085e-16
0.100000 0.985037473619294 0.985037473619294
0.125000 0.976653952117523 0.976653952117523
0.150000 0.966439543373742 0.966439543373742

```

0.175000 0.954413451738512 0.954413451738512
0.200000 0.940598313204911 0.940598313204911
0.225000 0.925020165493939 0.925020165493939
0.250000 0.907708413625372 0.907708413625372
0.275000 0.888695791011455 0.888695791011455
0.300000 0.868018316115753 0.868018316115753
0.325000 0.845715244724271 0.845715244724270
0.350000 0.821829017880773 0.821829017880773
0.375000 0.796405205542971 0.796405205542971
0.400000 0.769492446020924 0.769492446020924
0.425000 0.741142381263645 0.741142381263645
0.450000 0.711409588064441 0.711409588064441
0.475000 0.680351505260053 0.680351505260053
0.500000 0.648028357003025 0.648028357003025
0.525000 0.614503072191158 0.614503072191158
0.550000 0.579841200142074 0.579841200142074
0.575000 0.544110822605193 0.544110822605192
0.600000 0.507382462207426 0.507382462207426

ВЫВОД

В процессе выполнения данного курсового проекта были получены навыки вычисления и дальнейшего использования так называемого «машинного эпсилон». После генерации таблицы значений заданной функции можно увидеть, что значения совпадают до 10-14 знака после запятой. Из-за того, что существует понятие ограниченности разрядной сетки, вещественные числа имеют диапазон представления в памяти компьютера, что неизбежно приводит к тому, что в вычислениях в окрестности границ этого диапазона возникают погрешности.

На данный момент использование ряда Тейлора для вычисления трансцендентных функций является не оправданным, т. к. они требуют намного больше ресурсов, чем современные методы и имеют меньшую точность.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

- 1) Численные методы. Линейная алгебра и нелинейные уравнения. Учебное пособие. — Directmedia, 2014-05-20. — 432 с. [20.01.23]
- 2) Ильин В. А., Садовничий В. А., Сендов Б. Х. Математический анализ, ч. 1, изд. 3, ред. А. Н. Тихонов. М.: Проспект, 2004. [20.01.23]
- 3) Романов Е. Си/Си++. От дилетанта до профессионала. ermak.cs.nstu.ru. Проверено 25 мая 2015. [20.01.23]

