

Московский Авиационный Институт
(Национальный исследовательский университет)

Факультет информационных технологий и прикладной математики
Кафедра №806 Вычислительная математика и программирование

Курсовой проект

по курсам

**«Фундаментальная информатика», «Архитектура компьютера и
информационных систем»**

I семестр

Задание 3

Вещественный тип. Приближенные вычисления.

Табулирование функций.

Студент: Ибрагимов Д. М.

Группа: М8О-108Б-22

Руководитель: Сахарин Н. С.

Оценка:

Дата:

Подпись преподавателя:

Москва, 2022

СОДЕРЖАНИЕ

Задание.....	3
Вариант	3
Общий метод решения	3
Оборудование	4
Программное обеспечение	4
Функциональное назначение.....	4
Описание логической структуры.....	5
Описание переменных, констант и подпрограмм.....	5
Протокол.....	7
Входные данные	8
Выходные данные.....	8
Заключение.....	9

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 сама определять машинное ε и обеспечивать корректные размеры генерируемой таблицы.

Вариант

№	Ряд	a	b	Функция
2	$-\left(1 + \frac{2}{3}\right) - \left(1 + \frac{2}{3^2}\right)x - \dots - \left(1 + \frac{2}{3^{n+1}}\right)x^n$	0.0	0.5	$\frac{3x - 5}{x^2 - 4x + 3}$

Таблица 1

Общий метод решения

Необходимо вычислить значение заданной функции на отрезке от 0.0 до 0.5 двумя способами:

1. Используя программные средства, встроенные в стандартную библиотеку языка С «math.h»;
2. При помощи ряда Тейлора.

Оборудование

Процессор	AMD Ryzen 5 5600H (12) @ 3.600GHz
ОП	32GiB 3200 MHz LPDDR4
SSD	512 GiB
Монитор	27-дюймовый (1920 x 1080)
Графика	AMD Radeon™ RX 6600M 2177 MHz 8GiB GDDR6

Таблица 2 – оборудование

Программное обеспечение

Операционная система семейства	Microsoft Windows 11 Pro
Компилятор	GNU Compiler Collection
Текстовый редактор	Visual Studio Code версия 1.73.0

Таблица 3 – программное обеспечение

Функциональное назначение

Программа предназначена для высокоточного вычисления вещественного значения трансцендентных функций в алгебраической форме с использованием ряда Тейлора и при помощи встроенных программных функций библиотеки языка Си.

Описание логической структуры

Программа вычисляет значение функции в данной точке, используя программные средства языка программирования C, и при помощи разложения по ряду Тейлора. Ряд Тейлора мы преобразуем в отдельную функцию, которая вычисляет каждое слагаемое ряда. Затем складываем полученные слагаемые, пока их количество не превысит 100 или значение одного из них не станет совсем мало (меньше $\varepsilon \cdot k$ по модулю). В конце мы выводим таблицу с значением машинного эпсилона, значением аргумента, значением функции, вычисленным с помощью ряда Тейлора и с использованием программной библиотеки, а также номером итерации.

Описание переменных, констант и подпрограмм

Функция	Входные аргументы	Описание
epsilon	long double x	Функция для подсчета машинного ε . Сравниваем $(x + \varepsilon)/2$ с x . Последнее число, при стремлении ε к нулю, при котором $(x + \varepsilon)/2 = x$ и будет машинным ε
function	long double x	Вычисляет значение входной функции при помощи встроенной библиотеки «math.h»
Taylor	long double x, int n,	Считает сумму ряда по формуле Тейлора

Таблица 4 – описание функций программы

Переменная	Значение
double abs_eps	Абсолютный эпсилон
double relative_eps	Относительный эпсилон
const long double k	Эмпирический коэффициент для эпсилона
int MAX_ITER	Максимальное число итераций
long double a, b	Границы отрезка
long double step	Количество отрезков
long double n	Количество частей, на которые разбивается отрезок [a, b]
long double result, sum	Сумма ряда
long double x	Значение аргумента функции
int n	Текущая итерация

Таблица 5 – описание переменных

Протокол

Код программы:

```
#include <stdio.h>
#include <math.h>
#define MAX_ITER 100
typedef long double ldbl;
const ldbl k = 10e2;
ldbl epsilon() {
    ldbl eps = 1.0;
    while (1 + eps / 2.0 != 1) eps /= 2.0;
    return eps;
}
ldbl function(ldbl x) {
    return (3*x - 5)/(x*x - 4*x + 3);
}
ldbl Taylor(ldbl x, int n) {
    ldbl sum = 0;
    for (ldbl i = 0.0; i <= n; i++) sum -= (1 + 2.0/pow(3.0, i + 1))*pow(x, i);
    return sum;
}
int main() {
    ldbl abs_eps = epsilon();
    ldbl relative_eps = sqrt(abs_eps);
    ldbl a = 0.0, b = 0.5, result;
    int n;
    printf("Enter the number of iterations: ");
    scanf("%d", &n);
    printf("Machine epsilon for long double = %.16Le\n", abs_eps);
    ldbl step = (b - a) / n;
    printf("Taylor series value table for function f(x) = (3*x - 5)/(x^2 - 4*x + 3)\n");
    printf("|_____|\n");
    printf("|      x      |      sum of row      |      function      |      iter      |\n");
    printf("|_____|\n");
    for (ldbl x = a; x <= b + epsilon(); x += step) {
        for (n = 0; n < MAX_ITER; n++) {
            result = Taylor(x, n);
            if (fabs(result) <= fmax(relative_eps * fabs(result), abs_eps)) {
                break;
            }
        }
        printf("| %.5Lf | %.20Lf | %.20Lf | %d\t|\n", x, result, function(x), n);
        result = 0.0;
    }
    printf("|_____|\n");
    return 0;
}
```

Входные данные

На вход подается одно число n (5, 10, 15)

Выходные данные

```
PS C:\Users\lock_R\Documents\cc> gcc .\kp3.c -pedantic -Wall
PS C:\Users\lock_R\Documents\cc> .\a.exe
Enter the number of iterations: 5
Machine epsilon for long double = 1.0842021724855044e-19
Taylor series value table for function  $f(x) = (3*x - 5)/(x^2 - 4*x + 3)$ 
```

x	sum of row	function	iter
0.00000	-1.66666666666666651864	-1.666666666666666663	100
0.10000	-1.80076628352490407772	-1.80076628352490421454	100
0.20000	-1.96428571428571416572	-1.96428571428571428575	100
0.30000	-2.16931216931216917917	-2.16931216931216931209	100
0.40000	-2.43589743589743585326	-2.43589743589743589749	100
0.50000	-2.79999999999999988178	-2.799999999999999996	100

```
PS C:\Users\lock_R\Documents\cc> .\a.exe
Enter the number of iterations: 8
Machine epsilon for long double = 1.0842021724855044e-19
Taylor series value table for function  $f(x) = (3*x - 5)/(x^2 - 4*x + 3)$ 
```

x	sum of row	function	iter
0.00000	-1.66666666666666651864	-1.666666666666666663	100
0.06250	-1.74751773049645375846	-1.74751773049645390070	100
0.12500	-1.83850931677018619801	-1.83850931677018633538	100
0.18750	-1.94188034188034174910	-1.94188034188034188039	100
0.25000	-2.06060606060606047720	-2.060606060606060601	100
0.31250	-2.19873150105708232158	-2.19873150105708245233	100
0.37500	-2.36190476190476179126	-2.36190476190476190466	100
0.43750	-2.55826558265582649988	-2.5582655826558265822	100
0.50000	-2.79999999999999988178	-2.799999999999999996	100

```
PS C:\Users\lock_R\Documents\cc> .\a.exe
Enter the number of iterations: 10
Machine epsilon for long double = 1.0842021724855044e-19
Taylor series value table for function  $f(x) = (3*x - 5)/(x^2 - 4*x + 3)$ 
```

x	sum of row	function	iter
0.00000	-1.66666666666666651864	-1.666666666666666663	100
0.05000	-1.73059768064228353254	-1.73059768064228367522	100
0.10000	-1.80076628352490407772	-1.80076628352490421454	100
0.15000	-1.87822497420020626132	-1.87822497420020639847	100
0.20000	-1.96428571428571416572	-1.96428571428571428575	100
0.25000	-2.06060606060606047720	-2.060606060606060601	100
0.30000	-2.16931216931216917917	-2.16931216931216931209	100
0.35000	-2.29317851959361376757	-2.29317851959361393324	100
0.40000	-2.43589743589743585326	-2.43589743589743589749	100
0.45000	-2.60249554367201419705	-2.60249554367201426036	100
0.50000	-2.79999999999999988178	-2.8000000000000000039	100

Рис.1 – выходные данные

Заключение

В результате выполнения данного курсового проекта были получены навыки вычисления заданной функции при помощи встроенной библиотеки языка C «math.h» и с помощью разложения по ряду Тейлора. Также было изучено вычисление машинного эпсилона. Оценивая полученную таблицу, можно заметить, что значения функции различаются приблизительно после 16 знака после запятой. Так происходит, потому что для вещественных чисел выделяется ограниченное место в памяти компьютера, соответственно в окрестностях границ данного диапазона возникает погрешность.

Необходимо отметить, что вычисление трансцендентных функций при помощи формулы Тейлора не применяется на практике ввиду большой ресурсоемкости и значительной погрешности.