Московский Авиационный Институт

(Национальный исследовательский университет)

**Факультет информационных технологий и прикладной математики**

**Кафедра №806 Вычислительная математика и программирование**

# Курсовая работа

**по курсам  
«Архитектура компьютера», «Программные и аппаратные  
 средства информатики»  
I семестр**

**Задание 3**

**Вещественный тип. Приближенные вычисления. Табулирование функций.**

Студент: Былькова К. А.

Группа: М8О-108Б-22

Номер по списку: 2

Руководитель: Сахарин Н. А.

Оценка: <…>

Дата: <…>

Подпись преподавателя:

**СОДЕРЖАНИЕ**

1. Задание.…….……………………………………………………………….3
2. Вариант……………………………………………………………………...3
3. Общий метод решения……………………………………………………..3
4. Общие сведения о программе……………………………………………..4
5. Функциональное назначение.……………………………………………..4
6. Описание логической структуры……………………………………….....4
7. Описание переменных, констант и подпрограмм………………………..5
8. Протокол…………………...………………………………………………..6
9. Входные данные……...…………………………………………………….7
10. Выходные данные…………………...……………………………………..7
11. 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 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 сама определять машинное ε и обеспечивать корректные размеры генерируемой таблицы.

## Вариант

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ряд | a | b | Функция |
| 2 |  | 0.0 | 0.5 |  |

## 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 C «math.h»;
2. При помощи ряда Тейлора.

## Общие сведения о программе

Аппаратное обеспечение: домашний ноутбук

Операционная система: Linux Ubuntu, версия 22.04.1 LTS

Язык и система программирования: C, GNU

Местонахождение файлов: /home/Kristina

Компиляция программы: g++ -lm cp3.c

Запуск программы: ./a.out

## Функциональное назначение

Программа предназначена для высокоточного вычисления вещественного значения трансцедентных 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 ε\*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

Таблица 1. Описание функций программы

|  |  |  |
| --- | --- | --- |
| Функция | Входные аргументы | Описание |
| epsilon | long double x | Функция для подсчета машинного ε. Сравниваем (x + ε)/2 с x. Последнее число, при стремлении ε к нулю, при котором (x + ε)/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 2. Описание переменных

|  |  |
| --- | --- |
| Переменная | Значение |
| long double eps | Машинный эпсилон |
| const long double k | Эмпирический коэффициент для эпсилон |
| 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 |

## Протокол

**Код программы:**

#include <stdio.h>

#include <math.h>

#define MAX\_ITER 100

typedef double dbl;

typedef long double ldbl;

const ldbl k = 10e2;

ldbl epsilon(ldbl x) {

ldbl eps = 1.0;

while (x + eps / 2.0 != x)

eps /= 2.0;

return eps;

}

ldbl function(ldbl x) {

return (logl((1 + x) / ( 1 - x)));

}

ldbl Taylor(dbl x, int n) {

ldbl sum = 0;

for (ldbl i = 0.0; i <= n; i++) {

ldbl p = 2 \* i + 1;

sum = sum + pow(x,p) \* (2 / p);

}

return sum;

}

int main() {

ldbl a = 0.0, b = 0.5, x = 0.0, result;

int n, t;

printf("Enter the number of iterations: ");

scanf("%d", &n);

printf("\n");

ldbl step = (b - a) / n;

printf("Taylor series value table for function f(x) = ln((1+x)/(1-x))\n");

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \n");

printf("| Machine epsilon | x | sum of row | function | iter |\n");

printf("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|\n");

for (ldbl x = a; x <= b; x += step) {

ldbl eps = epsilon(x);

for (n = 0; n < MAX\_ITER; n++) {

result = Taylor(x, n);

if (fabs(result) < eps \* k) {

break;

}

}

printf("| %.20Lf | %.5Lf | %.20Lf | %.20Lf | %d\t|\n", eps, x, result, function(x), n);

result = 0.0;

}

printf("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_|\n");

return 0;

}

## Входные данные

На вход подается одно число n (5, 10, 12)

## Выходные данные

kristina@kristina-VirtualBox:~/Рабочий стол/course-w$ g++ -lm cp3.c && ./a.out

Enter the number of iterations: 5

Taylor series value table for function f(x) = ln((1+x)/(1-x))

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| Machine epsilon | x | sum of row | function |iter |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_|

|0.00000000000000000000|0.00000|0.00000000000000000000|0.00000000000000000000| 0 |

|0.00000000000000000000|0.10000|0.20067069546215117249|0.20067069546215116130|100|

|0.00000000000000000001|0.20000|0.40546510810816440511|0.40546510810816438199|100|

|0.00000000000000000003|0.30000|0.61903920840622340662|0.61903920840622343096|100|

|0.00000000000000000001|0.40000|0.84729786038720366666|0.84729786038720361369|100|

|0.00000000000000000005|0.50000|1.09861228866810969158|1.09861228866810969147|100|

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_|

kristina@kristina-VirtualBox:~/Рабочий стол/course-w$ g++ -lm cp3.c && ./a.out

Enter the number of iterations: 10

Taylor series value table for function f(x) = ln((1+x)/(1-x))

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| Machine epsilon | x | sum of row | function |iter |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_|

|0.00000000000000000000|0.00000|0.00000000000000000000|0.00000000000000000000| 0 |

|0.00000000000000000000|0.05000|0.10008345855698254206|0.10008345855698253650|100|

|0.00000000000000000000|0.10000|0.20067069546215117249|0.20067069546215116130|100|

|0.00000000000000000001|0.15000|0.30228087187293359921|0.30228087187293361054|100|

|0.00000000000000000001|0.20000|0.40546510810816440511|0.40546510810816438199|100|

|0.00000000000000000003|0.25000|0.51082562376599068336|0.51082562376599068319|100|

|0.00000000000000000003|0.30000|0.61903920840622340662|0.61903920840622343096|100|

|0.00000000000000000003|0.35000|0.73088750854279228755|0.73088750854279233823|100|

|0.00000000000000000003|0.40000|0.84729786038720366666|0.84729786038720361386|100|

|0.00000000000000000003|0.45000|0.96940055718810351097|0.96940055718810348327|100|

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_|

kristina@kristina-VirtualBox:~/Рабочий стол/course-w$ g++ -lm cp3.c && ./a.out

Enter the number of iterations: 12

Taylor series value table for function f(x) = ln((1+x)/(1-x))

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| Machine epsilon | x | sum of row | function |iter |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_|

|0.00000000000000000000|0.00000|0.00000000000000000000|0.00000000000000000000| 0 |

|0.00000000000000000000|0.04167|0.08338160893905105376|0.08338160893905105837|100|

|0.00000000000000000000|0.08333|0.16705408466316618269|0.16705408466316619204|100|

|0.00000000000000000001|0.12500|0.25131442828090607772|0.25131442828090607767|100|

|0.00000000000000000001|0.16667|0.33647223662121291147|0.33647223662121293050|100|

|0.00000000000000000001|0.20833|0.42285685082003358649|0.42285685082003356730|100|

|0.00000000000000000003|0.25000|0.51082562376599068336|0.51082562376599068319|100|

|0.00000000000000000001|0.29167|0.60077386042893020612|0.60077386042893016563|100|

|0.00000000000000000003|0.33333|0.69314718055994526769|0.69314718055994530937|100|

|0.00000000000000000001|0.37500|0.78845736036427016937|0.78845736036427016948|100|

|0.00000000000000000003|0.41667|0.88730319500090281976|0.88730319500090277509|100|

|0.00000000000000000001|0.45833|0.99039870402787689671|0.99039870402787694343|100|

|0.00000000000000000003|0.50000|1.09861228866810969158|1.09861228866810969136|100|

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_|

## Заключение

В результате выполнения данной курсовой работы 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 выделается 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.