МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

“Московский Авиационный Институт

(Национальный Исследовательский Университет)”

Факультет информационных технологий и прикладной математики

Кафедра 806 “Вычислительная математика и программирование”

Курсовая работа

по курсу “Вычислительные системы”

1 семестр

Задание 3. Вещественный тип. Приближенные вычисления. Табулирование функций

Студент: Немкова А.Р.

Группа: М8О-108Б-22,

№ по списку 14

Руководитель: Сахарин Н.А.

Дата:

Оценка:

Подпись преподавателя:

Москва, 2022

**Содержание**

ЗАДАЧА …………………………………………………………………...…. 3

ВАРИАНТ …………………………………………………………………..... 3

ОБЩИЙ МЕТОД РЕШЕНИЯ ……………………………………………… 3

ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ ……………………………………. 3

ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ……………………………………. 4

ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ ……………………………… 4

ОПИСАНИЕ ПЕРЕМЕННЫХ, КОНСТАНТ И ПОДПРОГРАММ ………. 4

ПРОТОКОЛ ………………………………………………………………….. 6

ВХОДНЫЙ ДАННЫЕ ……………………………………………………..... 7

ВЫХОДНЫЕ ДАННЫЕ ………………………………………………...…... 7

ВЫВОДЫ …………………………………………………………………….. 9

**1. Задача**

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.

**2. Вариант**

| № | Ряд | a | b | Функция |
| --- | --- | --- | --- | --- |
| 14 |  | 0,1 | 0,6 |  |

**3. Общий метод решения**

Вычисление значения функции в некоторой точке на отрезке от 0,1 до 0,6 двумя способами.

1 способ - использование программных средств, встроенных в стандартную математическую библиотеку языка Си “math.c”.

2 способ - при помощи ряда Тейлора.

**4. Общие сведения о программе**

Аппаратное обеспечение: домашний ноутбук

Операционная система: Linux Ubuntu, версия 22.04.1 LTS

Язык и система программирования: С, GNU

Местонахождение файлов: /home/anastasia

Компиляция программы: gcc -lm kp3.c

Вызов программы: ./a.out

**5. 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.

**6. Описание логической структуры**

Программа вычисляет значение функции в данной точке при помощи разложения по ряду Тейлора и с использованием программных средств языка программирования СИ. Ряд Тейлора мы преобразуем в функцию, которая вычисляет слагаемые ряда. Далее мы складываем полученные слагаемые, пока их количество не превысит 100 или значение одного из них не станет совсем мало (меньше ε\*л по модулю). В конце мы выводим таблицу с значением аргумента, значением функции, вычисленным с помощью ряда Тейлора и с использованием программной библиотеки, и номером итерации.

**7. Описание переменных, констант и подпрограмм**

Таблица 1. Описание функций программы

| Функция | Входные аргументы | Описание |
| --- | --- | --- |
| machine\_eps | - | Функция для подсчета машинного ε. Сравниваем 1+ε с 1. Последнее число, при стремлении ε к нулю, при котором 1+ε > 1 и будет машинным ε |
| func | long double x | Вычисляет значение входной функции при помощи встроенной библиотеки “math.c” |
| form\_teilor | long double x, int n, | Используя схему Горнера, считает сумму ряда по формуле Тейлора |

Таблица 2. Описание переменных

| Переменная | Значение |
| --- | --- |
| double abs\_eps | Абсолютное значение машинного эпсилон |
| double otn\_eps | Относительное значение машинного эпсилон |
| int MAX\_ITER | Максимальное число итераций |
| long double b, а | 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 |

**8. Протокол**

Код программы:

#include <stdio.h>

#include <math.h>

int MAX\_ITER = 100;

double machine\_eps() {

double eps = 1.0;

while ((1 + eps/2.0) != 1) {

eps /= 2.0;

}

return eps;

}

long double func(long double x) {

return (2 \* x - 3) / ((x - 1) \* (x - 1));

}

long double form\_teilor(long double x, int n){

long double sum = 0;

for (long double i = -(n + 3); i <= -3; i++){

sum \*= x;

sum += i;

}

return sum;

}

int main() {

long double b = 0.6, step, result;

long double a = 0.1;

int n;

double abs\_eps = machine\_eps();

double otn\_eps = sqrt(abs\_eps);

printf("Print the iterations: ");

scanf("%Lf", &step);

putchar('\n');

printf("Machine epsilon for double type = %.16e\n", abs\_eps);

printf("Taylor series values table for f(x) = (2x-3)/(x-1)^2\n");

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

printf("| x \t| sum of line\t\t | function\t\t | iter\t |\n");

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

for (long double x = a; x <= b; x += (b - a) / step) {

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

result = form\_teilor(x, n);

if (fabs(result) <= fmax(otn\_eps \* fabs(result), abs\_eps)) {

break;

}

}

if (fabs((int)result) >= 10) {

printf("| %.3Lf\t| %.20Lf| %.20Lf| %d\t |\n", x, result, func(x), n);

} else {

printf("| %.3Lf\t| %.20Lf | %.20Lf | %d\t |\n", x, result, func(x), n);

}

result = 0;

}

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

return 0;

}

**9. Входные данные**

На вход подается одно число n (3, 8, 15)

**10. Выходные данные**

anastasia@anastasia-VirtualBox:~$ g++ -lm kp3.c

anastasia@anastasia-VirtualBox:~$ ./a.out

Print the iterations: 3

Machine epsilon for double type = 2.2204460492503131e-16

Taylor series values table for f(x) = (2x-3)/(x-1)^2

---------------------------------------------------------------------------------------------

| x | sum of line | function | iter |

--------|------------------------------------|-----------------------------------|-------------

| 0.100 | -3.45679012345679015231 | -3.45679012345679015231 | 100 |

| 0.267 | -4.58677685950413219869 | -4.58677685950413219869 | 100 |

| 0.433 | -6.64359861591695479398 | -6.64359861591695479398 | 100 |

| 0.600 | -11.24999999999999902855| -11.24999999999999902855| 100 |

---------------------------------------------------------------------------------------------

anastasia@anastasia-VirtualBox:~$ ./a.out

Print the iterations: 8

Machine epsilon for double type = 2.2204460492503131e-16

Taylor series values table for f(x) = (2x-3)/(x-1)^2

---------------------------------------------------------------------------------------------

| x | sum of line | function | iter |

--------|------------------------------------|-----------------------------------|-------------

| 0.100 | -3.45679012345679015231 | -3.45679012345679015231 | 100 |

| 0.163 | -3.81376698596569393149 | -3.81376698596569393149 | 100 |

| 0.225 | -4.24557752341311133186 | -4.24557752341311133186 | 100 |

| 0.287 | -4.77685441674361337345 | -4.77685441674361337345 | 100 |

| 0.350 | -5.44378698224852060989 | -5.44378698224852060989 | 100 |

| 0.412 | -6.30149388863739230065 | -6.30149388863739230065 | 100 |

| 0.475 | -7.43764172335600874837 | -7.43764172335600874837 | 100 |

| 0.537 | -8.99926953981007979669 | -8.99926953981007979669 | 100 |

| 0.600 | -11.24999999999999902855| -11.24999999999999902855| 100 |

---------------------------------------------------------------------------------------------

anastasia@anastasia-VirtualBox:~$ ./a.out

Print the iterations: 15

Machine epsilon for double type = 2.2204460492503131e-16

Taylor series values table for f(x) = (2x-3)/(x-1)^2

---------------------------------------------------------------------------------------------

| x | sum of line | function | iter |

--------|------------------------------------|-----------------------------------|-------------

| 0.100 | -3.45679012345679015231 | -3.45679012345679015231 | 100 |

| 0.133 | -3.63905325443786984373 | -3.63905325443786984373 | 100 |

| 0.167 | -3.84000000000000001185 | -3.84000000000000001185 | 100 |

| 0.200 | -4.06250000000000000000 | -4.06250000000000000043 | 100 |

| 0.233 | -4.31001890359168240551 | -4.31001890359168240551 | 100 |

| 0.267 | -4.58677685950413219912 | -4.58677685950413219869 | 100 |

| 0.300 | -4.89795918367346933301 | -4.89795918367346933258 | 100 |

| 0.333 | -5.24999999999999991673 | -5.24999999999999991760 | 100 |

| 0.367 | -5.65096952908587245789 | -5.65096952908587245789 | 100 |

| 0.400 | -6.11111111111111094752 | -6.11111111111111094708 | 100 |

| 0.433 | -6.64359861591695479571 | -6.64359861591695479528 | 100 |

| 0.467 | -7.26562499999999970206 | -7.26562499999999970249 | 100 |

| 0.500 | -7.99999999999999960188 | -7.99999999999999960188 | 100 |

| 0.533 | -8.87755102040816273211 | -8.87755102040816273298 | 100 |

| 0.567 | -9.94082840236686318854 | -9.94082840236686318940 | 100 |

| 0.600 | -11.24999999999999902855| -11.24999999999999902855| 100 |

---------------------------------------------------------------------------------------------

**11. Вывод**

В ходе выполнения данной работы были получены навыки вычисления заданной функции при помощи разложения по ряду Тейлора и с помощью встроенной библиотеки СИ “math.c”, было изучено вычисление и использование машинного эпсилон. После составление таблицы значений заданной функции, можно увидеть, что значения различаются приблизительно после 16 знака после запятой. Это происходит из-за ограниченности разрядной сетки при представлении вещественных чисел, так как для данных чисел выделяется ограниченное количество памяти в компьютере, что приводит к тому, что в окрестностях границ данного диапазона возникают погрешности.

Вычисление трансцендентных функций при помощи формулы Тейлора не применяется на практике ввиду большой ресурсоемкости и значительной погрешности.