# Московский Авиационный Институт (национальный исследовательский университет)

# Факультет Компьютерные науки и прикладная математика Кафедра вычислительной математики и программирования

# КУРСОВАЯ РАБОТА по курсам «Архитектура компьютера», «Программные и аппаратные средства информатики»

# I семестр Задание 3. Вещественный тип. Приближенные вычисления. Табулирование функций

# Студент: Галкин А.Д. Группа: М8О-108Б-22, № 3 по списку Руководитель: Сахарин Н.А. Оценка:<………> Дата:<……….>

# Москва, 2022

**Содержание**

1. **ЗАДАЧА** …………………………………………………………………...…. 3

2. **ОБЩИЙ МЕТОД РЕШЕНИЯ** …………………………………………… 3

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

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

5. **ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ** …………………………… 4

6. **ОПИСАНИЕ ПЕРЕМЕННЫХ, КОНСТАНТ**……………………………. 4

7. **ПРОТОКОЛ** ………………………………………………………………….. 5

8. **ВХОДНЫЙ ДАННЫЕ** ……………………………………………………..... 7

9. **ВЫХОДНЫЕ ДАННЫЕ** ……………………………………………...…... 7

10. **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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ряд | a | b | Функция |
| 3 |  | -0.2 | 0.3 |  |

**2. Общий метод решения**

Вычисление значения функции в некоторой точке на отрезке от -0,2 до 0,3 двумя способами.

1 способ - при помощи ряда Тейлора.

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

**3. Общие сведения о программе**

Аппаратное обеспечение: Компьютер

Операционная система: Linux Ubuntu, версия 22.04.1 LTS

Язык и система программирования: С, GNU

Число строк: 68

Местонахождение файлов: /home/Alexey

Имя файла: KP3.c

Компиляция программы: gcc –std=c18 KP3.c - lm

Вызов программы: ./a.out

**4. Функциональное назначение**

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

**5. 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\*k 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.

**6. Описание переменных, констант**

Таблица 1. Описание функций программы

|  |  |  |
| --- | --- | --- |
| Функция | Входные аргументы | Описание |
| mach\_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 |
| logfunc | long double x | Вычисляет функцию при помощи встроенных в язык Си программных средств |
| logTaylor | long double x, int n, | Используя схему Горнера, считает сумму ряда по формуле Тейлора |
| table | long double a, long double b, int n | Выводит значения функций в виде таблицы |

Таблица 2. Описание переменных

|  |  |
| --- | --- |
| Переменная | Значение |
| long double eps | Машинный эпсилон |
| const long double k | Коэффициент для эпсилон |
| int MAX\_ITER | Максимальное число итераций |
| long double a, b | Границы отрезка |
| long double delta | Количество отрезков |
| long double n | Количество частей, на которые разбивается отрезок [a, b] |
| long double result | Сумма ряда |
| long double x | Значение аргумента функции |
| int n | Текущая итерация |

**7. Протокол**

Код программы:

#include <stdio.h>

#include <math.h>

#include <stdbool.h>

#define MAX\_ITER 100

typedef double dbl;

typedef long double ldbl;

const ldbl k = 10e2;

ldbl mach\_eps();

ldbl logfunc(ldbl x);

ldbl logTaylor(ldbl x, int n);

void table (dbl a, dbl b, int n);

int main(){

int n;

dbl a = -0.2, b = 0.3;

scanf ("%d", &n);

table (a, b, n);

return 0;

}

void table (dbl a, dbl b, int n){

ldbl result, delta = (b - a) / n;

ldbl eps = mach\_eps();

printf("Machine epsilon = %.20Lf\n", eps);

printf("Taylor for f(x) = (ln(1 + x - 2\*x\*\*2))\n");

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

printf(" x | Taylor | function | iterations |\n");

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

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

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

result = logTaylor(x,n);

}

printf("| %.6Lf | %.20Lf | %.20Lf | %d |\n", x, result, logfunc(x), n);

result = 0.0;

}

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

}

ldbl mach\_eps(){

ldbl eps = 1;

while (1 + eps / 2.0 > 1) {

eps /= 2.0;

}

return eps;

}

ldbl logfunc(ldbl x){

return (log(1 + x - 2\*x\*x));

}

ldbl logTaylor(ldbl x, int n){

ldbl result = 0, eps = mach\_eps();

ldbl member = x, substruction = 1, mult = 2, sum = 0;

for (int i = 1; i <= n; ++i){

substruction = mult - 1;

mult \*= -2;

sum = substruction \* member / i;

member \*= x;

result += sum;

if (fabs(result) <= (fmax(sqrt(eps) \* fabs(result), eps))) {

break;

}

}

return result;

}

**8. Входные данные**

На вход подается одно число n (2, 9, 23): количество точек, значения функции которых должна вычислить программа двумя способами.

**9. Выходные данные**

Программа должна вывести значение машинного эпсилона, таблицу со значениями функции, найденные разными способами.

**Протокол**

alexey@alexey-VirtualBox:~$ gcc -std=c18 KP3.c -lm

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

2

Machine epsilon = 0.00000000000000000011

Taylor for f(x) = (ln(1 + x - 2\*x\*\*2))

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

x | Taylor | function | iterations |

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

| -0.200000 | -0.32850406697203609490 | -0.32850406697203610173 | 100 |

| 0.050000 | 0.04401688541677431847 | 0.04401688541677425959 | 100 |

| 0.300000 | 0.11332868530700317301 | 0.11332868530700326848 | 100 |

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

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

9

Machine epsilon = 0.00000000000000000011

Taylor for f(x) = (ln(1 + x - 2\*x\*\*2))

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

x | Taylor | function | iterations |

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

| -0.200000 | -0.32850406697203609490 | -0.32850406697203610173 | 100 |

| -0.144444 | -0.20600726907122253215 | -0.20600726907122260401 | 100 |

| -0.088889 | -0.11058676878578848889 | -0.11058676878578853342 | 100 |

| -0.033333 | -0.03620304866396060462 | -0.03620304866396058946 | 100 |

| 0.022222 | 0.02101225608768021405 | 0.02101225608768012704 | 100 |

| 0.077778 | 0.06361216627744043726 | 0.06361216627744041563 | 100 |

| 0.133333 | 0.09328793442355704983 | 0.09328793442355702570 | 100 |

| 0.188889 | 0.11112166609289785312 | 0.11112166609289778063 | 100 |

| 0.244444 | 0.11772816446648794366 | 0.11772816446648796951 | 100 |

| 0.300000 | 0.11332868530700319124 | 0.11332868530700326848 | 100 |

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

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

23

Machine epsilon = 0.00000000000000000011

Taylor for f(x) = (ln(1 + x - 2\*x\*\*2))

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

x | Taylor | function | iterations |

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

| -0.200000 | -0.32850406697203609490 | -0.32850406697203610173 | 100 |

| -0.178261 | -0.27682752320257483973 | -0.27682752320257486556 | 100 |

| -0.156522 | -0.23006727603772489737 | -0.23006727603772494928 | 100 |

| -0.134783 | -0.18767423112043841916 | -0.18767423112043843392 | 100 |

| -0.113043 | -0.14919762307474315023 | -0.14919762307474307428 | 100 |

| -0.091304 | -0.11426371588465767914 | -0.11426371588465771001 | 100 |

| -0.069565 | -0.08256005121949272978 | -0.08256005121949275272 | 100 |

| -0.047826 | -0.05382360465441779915 | -0.05382360465441783648 | 100 |

| -0.026087 | -0.02783175003169168844 | -0.02783175003169163775 | 100 |

| -0.004348 | -0.00439527837015645574 | -0.00439527837015645396 | 100 |

| 0.017391 | 0.01664705509736980754 | 0.01664705509736974509 | 100 |

| 0.039130 | 0.03543282983634422660 | 0.03543282983634429839 | 100 |

| 0.060870 | 0.05207937475939166386 | 0.05207937475939163757 | 100 |

| 0.082609 | 0.06668649614053241391 | 0.06668649614053248631 | 100 |

| 0.104348 | 0.07933866463382738578 | 0.07933866463382732714 | 100 |

| 0.126087 | 0.09010677034863092538 | 0.09010677034863093238 | 100 |

| 0.147826 | 0.09904952788953091783 | 0.09904952788953097498 | 100 |

| 0.169565 | 0.10621459317381359102 | 0.10621459317381365395 | 100 |

| 0.191304 | 0.11163943864698732739 | 0.11163943864698741559 | 100 |

| 0.213043 | 0.11535202180357872169 | 0.11535202180357879975 | 100 |

| 0.234783 | 0.11737127267477980297 | 0.11737127267477986958 | 100 |

| 0.256522 | 0.11770741843074342637 | 0.11770741843074344901 | 100 |

| 0.278261 | 0.11636215691085279028 | 0.11636215691085274870 | 100 |

| 0.300000 | 0.11332868530700317301 | 0.11332868530700326848 | 100 |

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

alexey@alexey-VirtualBox:~$

**10. Вывод**

В ходе выполнения работы была составлена программа на языке Си, которая вычисляет значение функции 2 способами: разложением по ряду Тейлора и с помощью библиотеки Си «math.h». Было выяснено, что значения различаются примерно 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. Было изучено нахождение машинной эпсилон и способ вывода информации в виде таблицы.