**«Московский Авиационный Институт»**

(Национальный Исследовательский Университет)

**Институт: №8 «Прикладная математика и информатика»**

**Кафедра: 806 «Вычислительная математика и программирование»**

Курсовой проект

по курсу «Вычислительные системы»

I семестр

Задание 3

«Вещественный тип. Приближенные вычисления. Табулирование функций»

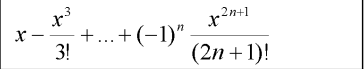
|  |  |
| --- | --- |
| Группа | М8О-107Б-20 |
| Студент | Чекменев В.А. |
| Преподаватель | Найдёнов И.Е. |
| Оценка |  |
| Дата |  |

Москва, 2020

**Постановка задачи**

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 \* 10k, 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 11:**

Ряд Тейлора: 

Функция:

Отрезок [a;b]: a = 0.0, b = 1.0

**Теоретическая часть**

**Формула Тейлора** - формула разложения функции в бесконечную сумму степенных функций. Формула широко используется в приближённых вычислениях, так как позволяет приводить трансцендентных функций к более простым. Сама она является следствием теоремы Лагранжа о среднем значении дифференцируемой функции. В случае a=0 формула называется рядом Маклорена.

**Машинный нол**ь - числовое значение с таким отрицательным порядком, которое воспринимается машиной как ноль.

**Машинный эпсилон** — числовое значение, меньше которого невозможно задавать относительную точность для любого алгоритма, возвращающего вещественные числа

В языке Си машинные эпсилон определено для следующих типов: float – 1.19 \* 10-7, double – 2.20 \* 10-16, long double – 1.08 \* 10-19.

**Описание алгоритма**

Найдём машинный эпсилон:

eps = 1.0;

while ((1.0 + eps / 2.0) > 1.0) {

eps = eps / 2.0;

}

Будем делить число пополам, пока оно не распознаётся машиной как ноль.

Найдём для каждой точки из отрезка значение функции с помощью встроенных функций языка Си. Для каждой точки будем искать приближённое значение функции, используя формулу Тейлора. Считаем члены формулы Тейлора, пока |t - f| > ε, где t - сумма ряда для функции, f - значение функции Вычисленное другим способом. Пока условие выполняется, ищем следующий член по формуле Тейлора и суммируем его с предыдущими членами.

**Код программы на Си**

/\* стандартная табличка \*/

#include <math.h>

#include <stdio.h>

double fact(double n)

{

if (n == 0 || n == 1) {

return 1;

} else {

return n \* fact(n - 1);

}

}

int main(void)

{

int n;

double eps = 1.0, x = 0, taylor\_series, func;

scanf("%d", &n);

double interval = 1.0 / n;

while ((1.0 + eps / 2.0) > 1.0) {

eps = eps / 2.0;

}

printf (" | Machine epsilon = %.8e |\n\n", eps);

for (int i = 0; i <= n; i++) {

taylor\_series = 0;

func = sin(x);

int counter = 0;

while (fabs(taylor\_series - func) >= eps \* 10) {

taylor\_series += pow(-1, counter) \* (pow(x, 2 \* counter + 1) / fact(2 \* counter + 1));

counter++;

}

printf ("| %.2lf | %.13lf | %.13lf | %d |\n", x, taylor\_series, func, counter);

x += interval;

}

return 0;

}

/\* красивая табличка \*/

#include <math.h>

#include <stdio.h>

double fact(double n)

{

if (n == 0 || n == 1) {

return 1;

} else {

return n \* fact(n - 1);

}

}

int main(void)

{

int n;

double eps = 1.0, x = 0, taylor\_series, func;

scanf("%d", &n);

double interval = 1.0 / n;

while ((1.0 + eps / 2.0) > 1.0) {

eps = eps / 2.0;

}

printf (" \*----------------------------------\*\n------| Machine epsilon = %.8e |------\n \*----------------------------------\*\n\*----------------------------------------------\*\n", eps);

for (int i = 0; i <= n; i++) {

taylor\_series = 0;

func = sin(x);

int counter = 0;

while (fabs(taylor\_series - func) >= eps \* 10) {

taylor\_series += pow(-1, counter) \* (pow(x, 2 \* counter + 1) / fact(2 \* counter + 1));

counter++;

}

printf ("| %.2lf | %.13lf | %.13lf | %d |\n", x, taylor\_series, func, counter);

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

x += interval;

}

return 0;

}

**Тесты программы с обычной табличкой**

Тест №1

Входные данные:

10

Выходные данные:

| Machine epsilon = 2.22044605e-16 |

| 0.00 | 0.0000000000000 | 0.0000000000000 | 0 |

| 0.10 | 0.0998334166468 | 0.0998334166468 | 5 |

| 0.20 | 0.1986693307951 | 0.1986693307951 | 5 |

| 0.30 | 0.2955202066613 | 0.2955202066613 | 6 |

| 0.40 | 0.3894183423086 | 0.3894183423087 | 6 |

| 0.50 | 0.4794255386042 | 0.4794255386042 | 7 |

| 0.60 | 0.5646424733950 | 0.5646424733950 | 7 |

| 0.70 | 0.6442176872377 | 0.6442176872377 | 8 |

| 0.80 | 0.7173560908995 | 0.7173560908995 | 8 |

| 0.90 | 0.7833269096275 | 0.7833269096275 | 8 |

| 1.00 | 0.8414709848079 | 0.8414709848079 | 9 |

Тест №2

Входные данные:

20

Выходные данные:

| Machine epsilon = 2.22044605e-16 |

| 0.00 | 0.0000000000000 | 0.0000000000000 | 0 |

| 0.05 | 0.0499791692707 | 0.0499791692707 | 4 |

| 0.10 | 0.0998334166468 | 0.0998334166468 | 5 |

| 0.15 | 0.1494381324736 | 0.1494381324736 | 5 |

| 0.20 | 0.1986693307951 | 0.1986693307951 | 5 |

| 0.25 | 0.2474039592545 | 0.2474039592545 | 6 |

| 0.30 | 0.2955202066613 | 0.2955202066613 | 6 |

| 0.35 | 0.3428978074555 | 0.3428978074555 | 6 |

| 0.40 | 0.3894183423086 | 0.3894183423087 | 6 |

| 0.45 | 0.4349655341112 | 0.4349655341112 | 7 |

| 0.50 | 0.4794255386042 | 0.4794255386042 | 7 |

| 0.55 | 0.5226872289307 | 0.5226872289307 | 7 |

| 0.60 | 0.5646424733950 | 0.5646424733950 | 7 |

| 0.65 | 0.6051864057360 | 0.6051864057360 | 7 |

| 0.70 | 0.6442176872377 | 0.6442176872377 | 8 |

| 0.75 | 0.6816387600233 | 0.6816387600233 | 8 |

| 0.80 | 0.7173560908995 | 0.7173560908995 | 8 |

| 0.85 | 0.7512804051403 | 0.7512804051403 | 8 |

| 0.90 | 0.7833269096275 | 0.7833269096275 | 8 |

| 0.95 | 0.8134155047894 | 0.8134155047894 | 8 |

| 1.00 | 0.8414709848079 | 0.8414709848079 | 9 |

Тест №3

Входные данные:

30

Выходные данные:

| Machine epsilon = 2.22044605e-16 |

| 0.00 | 0.0000000000000 | 0.0000000000000 | 0 |

| 0.03 | 0.0333271608368 | 0.0333271608368 | 4 |

| 0.07 | 0.0666172949234 | 0.0666172949234 | 4 |

| 0.10 | 0.0998334166468 | 0.0998334166468 | 5 |

| 0.13 | 0.1329386226223 | 0.1329386226223 | 5 |

| 0.17 | 0.1658961326934 | 0.1658961326934 | 5 |

| 0.20 | 0.1986693307951 | 0.1986693307951 | 5 |

| 0.23 | 0.2312218056343 | 0.2312218056343 | 6 |

| 0.27 | 0.2635173911435 | 0.2635173911435 | 6 |

| 0.30 | 0.2955202066613 | 0.2955202066613 | 6 |

| 0.33 | 0.3271946967962 | 0.3271946967962 | 6 |

| 0.37 | 0.3585056709286 | 0.3585056709286 | 6 |

| 0.40 | 0.3894183423086 | 0.3894183423087 | 6 |

| 0.43 | 0.4198983667038 | 0.4198983667038 | 7 |

| 0.47 | 0.4499118805560 | 0.4499118805560 | 7 |

| 0.50 | 0.4794255386042 | 0.4794255386042 | 7 |

| 0.53 | 0.5084065509313 | 0.5084065509313 | 7 |

| 0.57 | 0.5368227193940 | 0.5368227193940 | 7 |

| 0.60 | 0.5646424733950 | 0.5646424733950 | 7 |

| 0.63 | 0.5918349049588 | 0.5918349049588 | 7 |

| 0.67 | 0.6183698030697 | 0.6183698030697 | 7 |

| 0.70 | 0.6442176872377 | 0.6442176872377 | 8 |

| 0.73 | 0.6693498402505 | 0.6693498402505 | 8 |

| 0.77 | 0.6937383400791 | 0.6937383400791 | 8 |

| 0.80 | 0.7173560908995 | 0.7173560908995 | 8 |

| 0.83 | 0.7401768531960 | 0.7401768531960 | 8 |

| 0.87 | 0.7621752729138 | 0.7621752729138 | 8 |

| 0.90 | 0.7833269096275 | 0.7833269096275 | 8 |

| 0.93 | 0.8036082636944 | 0.8036082636944 | 8 |

| 0.97 | 0.8229968023633 | 0.8229968023633 | 8 |

| 1.00 | 0.8414709848079 | 0.8414709848079 | 9 |

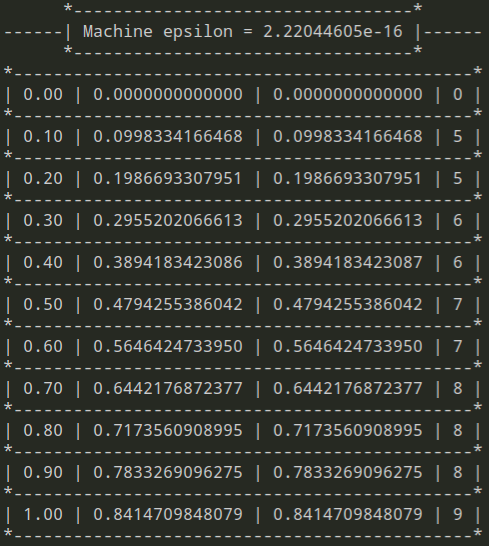
**Тесты программы с красивой табличкой**

Тест №1

Входные данные:

10

Выходные данные:

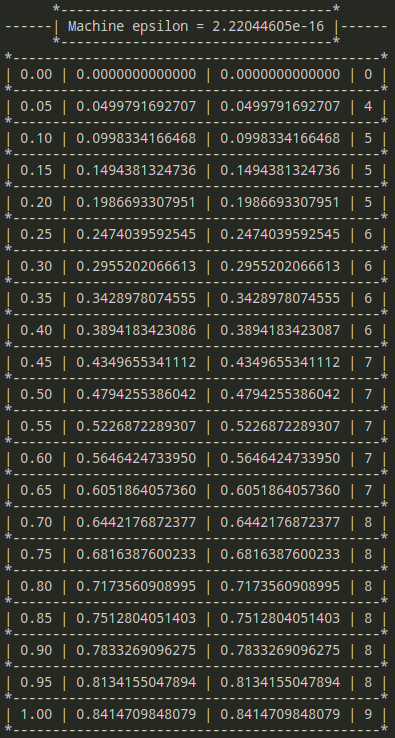


Тест №2

Входные данные:

20

Выходные данные:

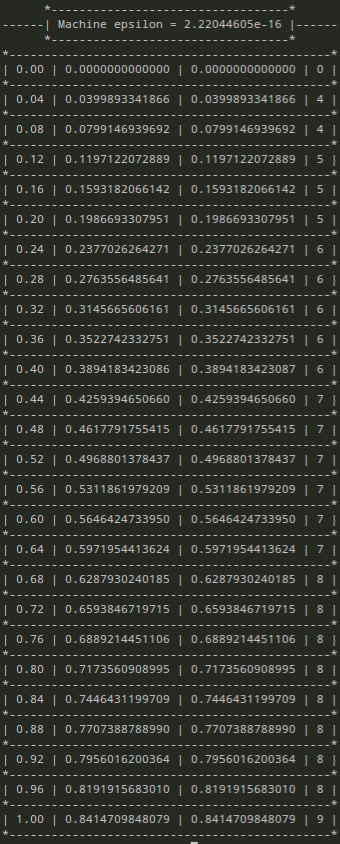


Тест №3

Входные данные:

25

Выходные данные:



**Вывод**:

В данном задание мне нужно было научиться работать с вещественными числами, точнее их представлением в компьютере, узнать что такое машинный эпсилон и машинный ноль. К тому же нужно было применить знания математического анализа, точнее одной темы - Ряды Тейлора.

Еще в прорамме был необычный вывод, нужно было применить табулирование функций, к тому же создать табличку.

После выполнения данного задния я смог понять основную суть и правила обращения с вещественными числами и понял как использовать математические знания в программировании.

**Список источников**

1. Машинный эпсилон – URL: https://qna.habr.com/q/160995
2. Ряд Тейлора – URL: http://www.mathprofi.ru/razlozhenie\_funkcij\_v\_stepennye\_ryady.html
3. Машинный ноль – URL: https://ru.wikipedia.org/wiki/Машинный\_ноль