МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский Авиационный Институт» (Национальный Исследовательский Университет)

Институт: №8 «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Курсовая работа

по курсу «Вычислительные системы» I семестр

Курсовая работа №3

«Вещественный тип. Приближенные вычисления. Табулирование функций»

|  |  |
| --- | --- |
| Группа | М8О-107Б-21 |
| Студент | Брюханов З.Д. |
| Преподаватель | Аносова Н.П. |
| Оценка |  |
| Дата |  |

Москва, 2022

Оглавление

[*Постановка задачи* 3](#_Toc92588921)

[*Вариант 5*: 3](#_Toc92588922)

[*Теоретическая часть* 4](#_Toc92588923)

[Формула Тейлора 4](#_Toc92588924)

[Машинное эпсилон 4](#_Toc92588925)

[*Описание алгоритма* 5](#_Toc92588926)

[*Описание переменных и констант* 6](#_Toc92588927)

[*Входные данные* 8](#_Toc92588928)

[*Выходные данные* 8](#_Toc92588929)

[*Протокол с тестами* 8](#_Toc92588930)

[*Вывод* 10](#_Toc92588931)

[*Список литературы* 10](#_Toc92588932)

# *Постановка задачи*

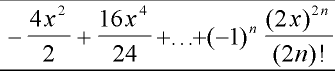
Написать программу на языке Си, печатающую таблицу значений элементарной функции, вычисленной двумя способами:

1. По формуле Тейлора для x0=0
2. С помощью встроенных функций языка программирования

В качестве аргументов таблицы взять точки разбиения отрезка [a;b] на n равных отрезков. 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 5*:

Ряд Тейлора:



Функция:



Значения а и b, соответственно:



# *Теоретическая часть*

## ­­Формула Тейлора ( Ряд Тейлора )

— разложение функции в бесконечную сумму степенных функций. Частный случай разложения в ряд Тейлора в нулевой точке называется рядом Маклорена. Ряд Тейлора был известен задолго до публикаций Брука Тейлора — его использовали ещё в XIV веке в Индии, а также в XVII веке Грегори и Ньютон. Ряды Тейлора применяются при аппроксимации функции многочленами. В частности, линеаризация уравнений происходит путём разложения в ряд Тейлора и отсечения всех членов выше первого порядка.



## Машинное эпсилон

— числовое значение, меньше которого невозможно задавать относительную точность для любого алгоритма, возвращающего вещественные числа. Абсолютное значение для машинного эпсилон зависит от разрядности сетки применяемой ЭВМ и от разрядности используемых при расчёте чисел. Формально это машинное эпсилон определяют как число, удовлетворяющее равенству 1 + ε = 1. Фактически, два отличных от нуля числа являются равными с точки зрения машинной арифметики, если их модуль разности меньше или не превосходит машинное эпсилон.

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*

Рассмотрим алгоритм решения.

Для начала нужно найти машинное эпсилон, на котором будет основываться точность вычисления. Реализуем функцию, которую можно сделать просто делением 1 на 2, пока 1 + ε больше 1, используя тип данных double.

Далее программа вычисляет значение функции в данной точке с помощью ряда Тейлора и при помощи программных средств языка программирования на отрезке от 0 до 0,5. Ряд Тейлора преобразуется в функцию, которая вычисляет слагаемые ряда. Следом идет сложение полученных слагаемых ряда до тех пор, пока одно из них не станет незначительным (по модулю меньше ε).

В итоге выводится таблица с текущим значением аргумента, значениями функции, вычисленными с помощью ряда Тейлора и с помощью подключаемой библиотеки и номером члена в ряде Тейлора.

# *Описание переменных и констант*

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Назначение |
| x | double | Текущее значение аргумента |
| a | Левая граница отрезка |
| b | Правая граница отрезка |
| k | Числитель дроби при вычислении текущего значения элемента ряда Тейлора |
| t | Текущее значение элемента ряда Тейлора |
| e | Значение машинного ε |
| ans1 | Ответ получаемый рядом Тейлора |
| ans2 | Ответ получаемый встроенной функцией |
| i | int | Номер шага, номер члена в ряде Тейлора |

***Исходный код программы:***

#include<stdio.h>

#include<math.h>

double taylor(double x, double n, double t) {

return ((t \* (-1) \* 4 \* x \* x) / ((2 \* n + 1) \* (2 \* n + 2)));

}

double f(double x) {

return -2 \* (sin(x) \* sin(x));

}

double eps(double e){

while (1 + e / 2 > 1) {

e /= 2;

}

return e;

}

int main() {

int i = 0;

double e = exp(1), a = 0, b = 0.5;

printf("Машинное эпсилон для типа double = %.16e\n", e);

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

printf(" Таблица значений ряда Тейлора и стандартной функции для f(x) = 2 \* (cos(x) \* cos(x) - 1) \n");

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

printf("Номер\t x\t\t Ряд тейлора\t\t\t f(x)\n");

for (double x = 0; x <= 0.5; x += 0.01) {

double ans1 = 0, ans2, t = 1;

for (int k = 1; k < 100; k++) {

ans1 += taylor(x, k - 1, t);

t = taylor(x, k - 1, t);

if(fabs(t) <= eps(ans1)){

break;

}

}

ans2 = f(x);

if (ans1 == 0) {

printf("%d\t %lf\t %.16e\t %.16e\t\n", i + 1, x, -ans1, ans2);

} else {

printf("%d\t %lf\t %.16e\t %.16e\t\n", i + 1, x, ans1, ans2);

}

i++;

}

}

# *Входные данные*

Целое число p (0≤N≤100) – количество частей отрезка, разделенного на равные части.

# *Выходные данные*

Программа должна вывести с 1 по p строку.

В каждой строке должно быть значение x – текущее значение аргумента, число tayVal — значение, вычисленное с помощью ряда Тейлора, значение, вычисленное с помощью встроенных функций языка программирования, n – количество итераций, требуемых для вычисления, оно же номер члена в ряде Тейлора, а также точность вычисления.

# *Протокол с тестами*

br\_zahar@Zahar-Lenovo-ideapad-330-15IKB:~/Programs/C++/Lessons$ gcc -g kurs.c -lm -o kurs

br\_zahar@Zahar-Lenovo-ideapad-330-15IKB:~/Programs/C++/Lessons$ ./kurs

Машинное эпсилон для типа double = 2.7182818284590451e+00

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

Таблица значений ряда Тейлора и стандартной функции для f(x) = 2 \* (cos(x) \* cos(x) - 1)

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

Номер x Ряд тейлора f(x)

1 0.000000 -0.0000000000000000e+00 -0.0000000000000000e+00

2 0.010000 -1.9999333342222162e-04 -1.9999333342222159e-04

3 0.020000 -7.9989333902205975e-04 -7.9989333902205975e-04

4 0.030000 -1.7994600647958344e-03 -1.7994600647958344e-03

5 0.040000 -3.1982936973806156e-03 -3.1982936973806151e-03

6 0.050000 -4.9958347219742349e-03 -4.9958347219742341e-03

7 0.060000 -7.1913641461337496e-03 -7.1913641461337488e-03

8 0.070000 -9.7840037873628289e-03 -9.7840037873628289e-03

9 0.080000 -1.2772716624373053e-02 -1.2772716624373053e-02

10 0.090000 -1.6156307211878582e-02 -1.6156307211878582e-02

11 0.100000 -1.9933422158758367e-02 -1.9933422158758363e-02

12 0.110000 -2.4102550669394505e-02 -2.4102550669394505e-02

13 0.120000 -2.8662025147970383e-02 -2.8662025147970390e-02

14 0.130000 -3.3610021865486762e-02 -3.3610021865486762e-02

15 0.140000 -3.8944561689229044e-02 -3.8944561689229044e-02

16 0.150000 -4.4663510874393984e-02 -4.4663510874393977e-02

17 0.160000 -5.0764581917559129e-02 -5.0764581917559143e-02

18 0.170000 -5.7245334471653787e-02 -5.7245334471653787e-02

19 0.180000 -6.4103176322065178e-02 -6.4103176322065164e-02

20 0.190000 -7.1335364423489769e-02 -7.1335364423489769e-02

21 0.200000 -7.8939005997114925e-02 -7.8939005997114939e-02

22 0.210000 -8.6911059687691777e-02 -8.6911059687691763e-02

23 0.220000 -9.5248336780036622e-02 -9.5248336780036635e-02

24 0.230000 -1.0394750247447482e-01 -1.0394750247447482e-01

25 0.240000 -1.1300507722071586e-01 -1.1300507722071587e-01

26 0.250000 -1.2241743810962732e-01 -1.2241743810962735e-01

27 0.260000 -1.3218082032235021e-01 -1.3218082032235018e-01

28 0.270000 -1.4229131863617595e-01 -1.4229131863617595e-01

29 0.280000 -1.5274488898658395e-01 -1.5274488898658398e-01

30 0.290000 -1.6353735008481315e-01 -1.6353735008481318e-01

31 0.300000 -1.7466438509032184e-01 -1.7466438509032181e-01

32 0.310000 -1.8612154333746622e-01 -1.8612154333746619e-01

33 0.320000 -1.9790424211570751e-01 -1.9790424211570753e-01

34 0.330000 -2.1000776850263508e-01 -2.1000776850263508e-01

35 0.340000 -2.2242728124907221e-01 -2.2242728124907221e-01

36 0.350000 -2.3515781271551175e-01 -2.3515781271551178e-01

37 0.360000 -2.4819427085910517e-01 -2.4819427085910520e-01

38 0.370000 -2.6153144127041233e-01 -2.6153144127041233e-01

39 0.380000 -2.7516398925909502e-01 -2.7516398925909508e-01

40 0.390000 -2.8908646198772292e-01 -2.8908646198772292e-01

41 0.400000 -3.0329329065283489e-01 -3.0329329065283489e-01

42 0.410000 -3.1777879271238674e-01 -3.1777879271238668e-01

43 0.420000 -3.3253717415869222e-01 -3.3253717415869216e-01

44 0.430000 -3.4756253183594848e-01 -3.4756253183594854e-01

45 0.440000 -3.6284885580142012e-01 -3.6284885580142007e-01

46 0.450000 -3.7839003172933583e-01 -3.7839003172933589e-01

47 0.460000 -3.9417984335653755e-01 -3.9417984335653761e-01

48 0.470000 -4.1021197496890216e-01 -4.1021197496890216e-01

49 0.480000 -4.2648001392754370e-01 -4.2648001392754376e-01

50 0.490000 -4.4297745323378312e-01 -4.4297745323378307e-01

# *Вывод*

В ходе выполнения курсовой работы я расширил и углубил свои знания относительно Ряда Тейлора, а также составил программу на языке Си с вычислением функции двумя способами, и убедился, что значения функций практически совпадают (примерно до 16 символа после запятой).

# *Список литературы*

1. Электронный справочник, статья про машинный ноль

[https://ru.wikipedia.org/wiki/Машинный\_ноль](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BD%D0%BE%D0%BB%D1%8C)

1. Электронный справочник, статья про Ряд Тейлора

[https://ru.wikipedia.org/wiki/Ряд\_Тейлора](https://ru.wikipedia.org/wiki/%D0%A0%D1%8F%D0%B4_%D0%A2%D0%B5%D0%B9%D0%BB%D0%BE%D1%80%D0%B0)