ГУАП

КАФЕДРА № 34

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ст. препод. |  |  |  | К.А. Жиданов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1 |
|  |
| по курсу: ЯЗЫКИ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 3145 |  |  |  | Е.М. Гаврилов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Вариант 3.**

Вычислить значение тангенса с заданной точностью. (Ряд Тейлора)

**Цель работы:**

Изучить методы разработки консольных приложений, способы их запуска и обработки кодов возврата.

**Ход работы:**

1. Реализовал на языке Си функцию, выполняющую вычисление тангенса через формулы синуса и косинуса для ряда Тейлора:

Код для нахождения синуса по ряду Тейлора:

double mysin(double x, double ep)

{

double S = 0, S\_1; // S - сумма ряда, S\_1 - предыдущее значение суммы ряда

int zn = 1, n = 1, sign = 1; // sign - для убирания периодов

// при x < 0 идем в положительную часть синуса

if (x < 0)

{

sign = -1; //меняем знак на минус

x = fabs(x); //угол делаем по модулю

}

//если угол больше 2Пи, то идём в область от 0 до 2Пи

x = fmod(x, 2 \* pi);//делим угол на 2Пи, получая остаток от деления

//если x больше Пи, то идем в область от 0 до Пи

if (x > pi)

{

x = x - pi;

sign = sign \* (-1); // меняем знак на противоположный. (если был -, будет +)

}

//x > (pi/2) peremeshtaem v interval [0..pi/2]

if (x > (pi / 2))

x = pi - x; // пример: если из Пи вычесть 3/4Пи получим числый угол Пи/4 (синус(3/4Пи) = синус(Пи/4))

do

{

S\_1 = S;

S = S + (zn \* pow(x, 2 \* n - 1) / fact(2 \* n - 1));

n = n + 1;

zn = zn \* (-1);

} while (fabs(S - S\_1) > ep); //остаемся в цикле до тех пор пока выполняется условие

S = S \* sign;

return S;

}

Код для нахождения косинуса по ряду Тейлора:

double mycos(double x, double ep)

{

double S = 0, S\_1;

int zn = 1, n = 1, sign = 1;

//если х меньше 0, берем по модулю

if (x < 0) x = fabs(x);

//ecli x>2Pi peremeshtaem v interval [0..2pi]

x = fmod(x, 2 \* pi);//ostatok ot deleniya x/(2\*pi) , ubiraem period

//x > pi peremeshtaem v interval [0..pi]

if (x > pi)

{

x = 2 \* pi - x; // чтобы получить угол от 0 до пи, вычитаем из 2пи найденный угол >пи

}

//x > (pi/2) peremeshtaem v interval [0..pi/2]

if (x > (pi / 2))

{

x = pi - x;

sign = sign \* (-1); //т.к от 0 до пи/2 знак + а от пи/2 до пи знак -, то меняем знак

}

do

{

S\_1 = S; //predydushee znachenie summy

S = S + (zn \* pow(x, 2 \* n - 2) / fact(2 \* n - 2));

n = n + 1;

zn = zn \* (-1);

} while (fabs(S - S\_1) > ep);

S = S \* sign;

return S;

}

Путём деления синуса на косинус, найдем тангенс:

double mytan(double x, double ep) //ищем тангенс из деления двух функций

{

double s, c; // s -синус, c - косинус

s = mysin(x, ep); //идёт в функцию синуса

c = mycos(x, ep); //идёт в функцию косинуса

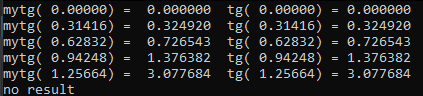
return(s/c);

1. Составляю набор эталонных параметров и результатов для проверки работы программы. Для этого вывожу в консоль значение тангенса через встроенную функцию tan().

printf("mytg(%8.5f) = %9.6f tg(%8.5f) =%9.6f\n", g, mytan(g, ep), g, tan(g))

mytan – значение тангенса по ряду Тейлора  
 tan – значение от встроенной функции

При выводе от 0<= g<=Пи/2 с шагом Пи/10 получил следующие значения:



1. Тестирующая функция:

int main(void)

{

double ep = 0.000001;

double g; // для подсчёта угла

for (g = -2 \* pi; g <= 2 \* pi; g = g + pi / 10.) // с шагом 1/10 pi идём до 6pi от -2pi (перебираем разные углы, чтобы посчитать тангенс)

{

if (fabs(fmod(fabs(g), pi) - pi / 2) >= ep) //все точки pi/2 этой функцией убираем

{

printf("mytg(%8.5f) = %9.6f tg(%8.5f) =%9.6f\n", g, mytan(g, ep), g, tan(g)); // печать двух вещ. чисел с заданной точностью ep

}

В зависимости от изменения значения ep, количество знаков выводимых чисел в консоль меняется.

1. Вывод ошибки:

Выводим в консоль “no result” если при вычитании от модуля угла без периода, мы получаем угол, близкий к Pi/2 (тангенс не существует).

if (fabs(fmod(fabs(g), pi) - pi / 2) >= ep) //все точки pi/2 этой функцией убираем

{

printf("mytg(%8.5f) = %9.6f tg(%8.5f) =%9.6f\n", g, mytan(g, ep), g, tan(g)); // печать двух вещ. чисел с заданной точностью с жёстким порядком

}

else

{

printf("no result\n"); // разница между числами меньше EP

}

1. Компилирую программу, работает корректно. Значения тангенса через ряд Тейлора совпадают со значениями встроенной функции. При значениях, близких к Pi/2 по модулю программа выводит “no result”