

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 «Фундаментальная информатика»
I семестр
Задание 3
«Вещественный тип. Приближенные вычисления. Табулирование
функций»

Группа	М8О-109Б-22
Студент	Горохов М.С.
Преподаватель	Сысоев М.А.
Оценка	
Дата	

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 $\varepsilon * 10^k$, 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.

Вариант 22:

Ряд Тэйлора:

$$1 - \frac{x^2}{2} + \frac{x^3}{3} + \dots + (-1)^{n-1} \frac{n-1}{n!} x^n$$

Функция:

$$(1 + x)e^{-x}$$

Значения a и b : 0.0 и 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.

$$\sum_{n=0}^k \frac{f^{(n)}(a)}{n!} (x-a)^n = f(a) + f^{(1)}(a)(x-a) + \frac{f^{(2)}(a)}{2!} (x-a)^2 + \dots + \frac{f^{(k)}(a)}{k!} (x-a)^k$$

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 $1 + \varepsilon = 1$. 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 машинное эпсилон определено для следующих типов: float — $1.19 \cdot 10^{-7}$, double — $2.20 \cdot 10^{-16}$, long double — $1.08 \cdot 10^{-19}$.

Описание алгоритма

Рассмотрим алгоритм решения. Сперва нужно найти машинное эпсилон, на котором будет основываться точность вычисления. Это можно сделать просто деля 1 на 2.

Для каждой $N+1$ строки нужно просуммировать i членов формулы Тейлора, пока $|A_1 - A_2| > \varepsilon$. Для этого просто ищем каждый новый член из формулы Тейлора и суммируем с результатом

Использованные в программе переменные

Название переменной	Тип переменной	Смысл переменной
n	int64_t	То самое число N, на которое нужно разбить отрезок
k	int	То самое число K, используемое для вычисления точности.
FLT_EPSILON	float	То самое машинное эпсилон.
		1.192092896e-07F
step	long double	Формально разница между предыдущим значением из отрезка и следующим, если отрезок разбит на n равных частей.
currentX	long double	Переменная, для которой будем производить вычисления
Taylor_series (currentX, i)	double	То самое значение A ₁ , вычисленное с помощью формулы Тейлора
func(currentX)	double	То самое значение A ₂ , вычисленное с помощью встроенных функций языка
i	double	Счётчик члена формулы Тейлора + кол-во итераций

Исходный код программы:

```
#include <stdio.h>
#include <float.h>
#include <stdint.h>
#include <math.h>

int64_t factorial(int64_t n) {
    int64_t res = 1;

    for (int64_t i = 1; i <= n; ++i)
        res *= i;

    return res;
}

long double taylor_series(uint64_t n, long double x){
    long double result = 0;
    for (int i = 0; i <= n; ++i){
        result += powl(-1, (long double) i - 1) * (((long double) (i - 1))
/ ((long double) factorial(i))) * powl(x, (long double) i);
    }
    return result;
}

long double function(long double x){
    return (1 + x) * expl(-x);
}

int main(){
    long double a = 0.0;
    long double b = 1.0;

    uint64_t n;

    printf("Input N:");
    scanf_s("%lld", &n);
    printf("N = %lld\n", n);
    printf("Machine epsilon is equal to: %Lg\n\n", LDBL_EPSILON);

    printf("          Table of values of Taylor series and standard
function\n");

    printf("
    _____\n");
    printf("|  x  | sum of Taylor series | f(x) function value | number of
iterations | \n");

    printf("
    _____\n");

    long double x = 0;
    long double step = (b - a) / n;
    long double func = 1;
    int i = 0;
    while (fabsl(func) > LDBL_EPSILON && (i < 100) && (i < n)){
        i += 1;
        x += step;
        func = function(x);
    }
```

```
        printf("|%.3Lf|%.20Lf|%.19Lf|          %d          |\n", x,  
taylor_series(i, x), func, i);  
    }  
  
printf("_____\n");  
  
    return 0;  
}
```

Входные данные

Единственная строка содержит одно целое число N ($0 \leq N \leq 100$) – число разбиений отрезка на равные части

Выходные данные

Программа должна вывести значение машинного эпсилон, а затем $N+1$ строку.

В каждой строке должно быть значение x , для которого вычисляется функция, число A_1 — значение, вычисленное с помощью формулы Тейлора, A_2 — значение, вычисленное с помощью встроенных функций языка, i — количество итерация, требуемых для вычисления, и Δ — разница значений A_1 и A_2 по модулю. A_1 , A_2 и Δ должны быть выведены с точностью 16 знаков после запятой.

Протокол исполнения и тесты

Тест №1

Ввод:

3

Вывод:

```
N = 3
Machine epsilon is equal to: 1.0842e-19

      Table of values of Taylor series and standard function
-----
|  x  | sum of Taylor series | f(x) function value | number of iterations |
-----
|0.333|1.00000000000000000000|0.9553750807650523339|          1          |
|0.667|0.7777777777777777775|0.8556951983876533782|          2          |
|1.000|0.8333333333333333337|0.7357588823428846432|          3          |
-----
```

Process finished with exit code 0

Тест №2

Ввод:

100

Вывод:

```
N = 100
Machine epsilon is equal to: 1.0842e-19

      Table of values of Taylor series and standard function
-----
|  x  | sum of Taylor series | f(x) function value | number of iterations |
-----
|0.010|1.00000000000000000000|0.9999503320866597341|          1          |
|0.020|0.99980000000000000000|0.9998026467728904082|          2          |
|0.030|0.99955900000000000000|0.9995588995549634222|          3          |
|0.040|0.99922101333333333335|0.9992210167184161378|          4          |
|0.050|0.99879089583333333330|0.9987908957257497095|          5          |
|0.060|0.99827040559600000005|0.9982704055993036321|          6          |
|0.070|0.99766138729946380551|0.9976613872993646048|          7          |
|0.080|0.99696565409756371297|0.9969656540975666456|          8          |
```

Process finished with exit code 0

Тест №3

Ввод:

100000

Вывод:

```
N = 1000
Machine epsilon is equal to: 1.0842e-19

      Table of values of Taylor series and standard function
-----
|  x  | sum of Taylor series | f(x) function value | number of iterations |
-----
|0.001|1.00000000000000000000|0.9999995003332083666|          1          |
|0.002|0.99999800000000000001|0.9999980026646677329|          2          |
|0.003|0.99999550899999999998|0.9999955089898830949|          3          |
|0.004|0.99999202130133333337|0.9999920213013674382|          4          |
|0.005|0.99998754158864583337|0.9999875415886457249|          5          |
|0.006|0.99998207183825887597|0.9999820718382588764|          6          |
|0.007|0.99997561403376775065|0.9999756140337677506|          7          |
```

Process finished with exit code 0

Вывод

В работе описано определение машинного эпсилон, приведены его значения для разных переменных языка Си, описана формула Тейлора и составлен алгоритм реализации вычисления значения функции с заданной точностью для заданного числа точек на отрезке. На основе алгоритма составлена программа на языке Си, проведено её тестирование на различных тестах, составлен протокол исполнения программы. В целом, работа понравилась. Приятно применять знания из других областей для решения какой-либо задачи по программированию.

Список литературы

1. Машинный ноль – URL: https://ru.wikipedia.org/wiki/Машинный_ноль
2. Ряд Тейлора – URL: https://ru.wikipedia.org/wiki/Ряд_Тейлора