

Московский Авиационный Институт
(Национальный Исследовательский Университет)
Факультет информационных технологий и прикладной математики
Кафедра 806 “Вычислительная математика и программирование”

Курсовая работа
по курсу “Архитектура компьютера”
1 семестр

Задание 3. Вещественный тип.
Приближенные вычисления.
Табулирование функций

Студент: Старостина А.А.

Группа: М8О-108Б-22

№ по списку: 19

Руководитель: Сахарин Н.А.

Дата: 08.01.2023

Оценка: _____

г. Москва, 2023

Содержание

1. Задание	3
2. Вариант	3
3. Общий метод решения.....	3
4. Общие сведения о программе	4
5. Функциональное назначение	4
6. Описание логической структуры	4
7. Описание переменных, констант и подпрограмм.....	5
8. Протокол.....	6
9. Входные данные	8
10. Выходные данные	8
11. Выводы	10

Задание

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 $\varepsilon \cdot k$, 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 19

№	Ряд	a	b	Функция
19	$1 + \frac{x^2}{2} \dots + \frac{x^{2n}}{(2n)!}$	0.1	0.6	$\sinh x$

Общий метод решения

Вычисление значения функции в некоторой точке на отрезке от 0.1 до 0.6 двумя способами:

1: использование программных средств, встроенных в стандартную математическую библиотеку языка Си “math.c”.

2 : при помощи ряда Тейлора.

Общие сведения о программе

Аппаратное обеспечение: домашний ноутбук

Операционная система: Linux Ubuntu, версия 22.04.1 LTS

Язык и система программирования: C, GNU

Местонахождение файлов: /home/ann

Компиляция программы: gcc -lm kp3.c

Вызов программы: ./a.out

Функциональное назначение

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

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. В конце мы выводим таблицу с значением аргумента, значением функции, вычисленным с помощью ряда Тейлора и с использованием программной библиотеки, и номером итерации.

Описание переменных, констант и подпрограмм

Функция	Входные аргументы	Описание
computer_epsilon n		Функция для подсчета машинного ϵ . Сравниваем $1+\epsilon/2$ с 1. Последнее число, при стремлении ϵ к нулю, при котором $1+\epsilon/2 = 1$ и будет машинным ϵ
func	long double x	Вычисляет значение входной функции
taylor	long double x	Считает сумму ряда по формуле Тейлора
tabulation	const ld a, const ld b, const uint steps	Печатает значения в таблицу

Таблица 1. Описание функций программы

Переменная	Значение
long double eps	Машинный эпсилон (абсолютный)
MAX_ITER	Максимальное число итераций

const long double a,b	Границы отрезка
const long double step	Отрезок
long double n, steps	Количество разделителей, которые разбивают отрезок [a, b]
long double result	Сумма ряда
long double x	Значение аргумента функции
int n	Текущая итерация
long double cur_member	Текущий член
long double ch	Вычисляет числитель
long double fac	Вычисляет знаменатель

Таблица 2. Описание переменных

Протокол

Код программы:

```
#include <math.h>
```

```
#include <stdio.h>
```

```
#define MAX_ITER 100
```

```
typedef long double ld;
```

```
typedef unsigned uint;
```

```
ld computer_epsilon() {
```

```

    ld eps = 1.0;
    while (1.0 + eps / 2.0 != 1.0)
        eps *= 0.5;
    return eps;
}

ld func(ld x) {
    return cosh(x);
}

ld taylor(ld x) {
    ld eps = computer_epsilon();
    int n = 0.0;
    ld cur_member = 1.0, ch = 1.0, fac = 1.0, result = 0.0;
    while ((fabs1(cur_member) > sqrt(eps) && n <= MAX_ITER)) {
        cur_member *= ch / fac;
        ch *= x*x;
        fac *= 2*(n + 1);
        result += cur_member;
        n++;
    }
    return result;
}

void tabulation(const ld a, const ld b, const uint steps) {
    const ld step = b/steps - a/steps;
    for (ld x = a; x <= b; x += step)
        printf("|%.3Lf|%.19Lf|%.19Lf\n", x, func(x), taylor(x));
}

```

```
int main() {
    ld eps = computer_epsilon();
    printf("Machine epsilon: %.45Lf\n", eps);
    printf("_____ \n");
    printf("| x | function | Taylor | \n");
    printf("|_____|_____|\n");
    tabulation(0.1, 0.6, 14);
    printf("|_____|_____|\n");
}
```

Входные данные

Не подаются. В программе указано выполнить действия для 14 итераций.

Выходные данные

```
ann@ann:~$ gcc kp3.c -lm
```

```
ann@ann:~$ ./a.out
```

[illegible]

x	sum of row	function
0.100	1.00500416805580359996	1.00500416805580705670
0.138	1.00960112319658304086	1.00960112319658978919
0.177	1.01569175524130606770	1.01569175524131716237
0.215	1.02328507511142135520	1.02328507511143773634
0.254	1.03239231691340289173	1.03239231691342592256
0.292	1.04302695455929943150	1.04302695455932981083

0.331	1.05520472170101423210	1.05520472170105339416
0.369	1.06894363500780754697	1.06894363500785605225
0.408	1.08426402082146028772	1.08426402082151840389
0.446	1.10118854522853490890	1.10118854522860010192
0.485	1.11974224759422623283	1.11974224759428864218
0.523	1.13995257760741697584	1.13995257760744506115
0.562	1.16184943589175203644	1.16184943589166156386
0.600	1.18546521824282886838	1.18546521824240169707
_____	_____	_____

Вывод

В ходе выполнения данной работы были получены навыки вычисления заданной функции при помощи разложения по ряду Тейлора и с помощью встроенной библиотеки “math.c”, было изучено вычисление и использование машинного эпсилон., как абсолютного, так и относительного. При сравнении данных в таблице, можно увидеть, что значения различаются приблизительно после 13 знака после запятой. Это происходит из-за ограниченного количества памяти в компьютере для представления вещественных чисел, что приводит к тому, что в окрестностях границ данного диапазона возникают погрешности.

И что важно, вычисление трансцендентных функций при помощи формулы Тейлора не применяется на практике ввиду большой ресурсоемкости и значительной погрешности.