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	Любарский И.В.
Преподаватель	Сысоев М.А.
Оценка	
Дата	

Москва, 2022

Введение

Цель: Составить программу на Си, которая печатает таблицу значений элементарных функций, 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 точность ξ*k, где ξ — 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 ξ и обеспечивать корректные размеры генерируемой таблицы. (Вариант 10)

Задачи:

- Создать функцию, вычисляющую данную математическую функцию по формуле Тейлора.
- Создать функцию, вычисляющую данную математическую функцию при помощи встроенных математических функций.
- Настроить точность вычислений.
- Вычислить значения точек отрезка (значения принимаемой аргументом заданной функции).
- Скомпоновать готовые функции и данные в программу вывода таблицы.

10	$\frac{2x^2}{2!} - \frac{2^3 x^4}{4!} + \ldots + (-1)^{n-1} \frac{2^{2n-1} x^{2n}}{(2n)!}$	0.0	1.0	$\sin^2 x$

Дополнительная информация

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. Частный случай разложения в ряд Тейлора в нулевой точке называется **рядом Маклорена:**

Машинный эпсилон - числовое значение, меньше которого невозможно задавать относительную точность для любого алгоритма, возвращающего вещественные числа. Абсолютное значение «машинного эпсилон» зависит от разрядности сетки применяемой ЭВМ, типа (разрядности) используемых при расчетах чисел, и от принятой в конкретном трансляторе структуры представления вещественных чисел (количества бит, отводимых на мантиссу и на порядок). Формально машинный эпсилон обычно определяют как минимальное из чисел ε, для которого $1+\varepsilon>1$ при машинных расчетах с числами данного типа. Альтернативное определение — максимальное ε, для которого справедливо равенство $1+\varepsilon=1$.

В языке Си существуют предельные константы FLT_EPSILON, DBL_EPSILON и LDBL_EPSILON являющиеся «машинными эпсилон», соответствующими первому определению: FLT_EPSILON = $2^{-23} \approx 1.19e\text{-}07$ — это машинный эпсилон для чисел типа float (32 бита), DBL_EPSILON = $2^{-52} \approx 2.20e\text{-}16$ — для типа double (64 бита), и LDBL_EPSILON = $2^{-63} \approx 1.08e\text{-}19$ — для типа long double (80 бит).

При альтернативном определении соответствующие машинные эпсилон будут вдвое меньше: 2^{-24}, 2^{-53} и 2^{-64}.В некоторых компиляторах Си (например gcc, Intel’s C/C++ compiler) допускается использование переменных четверной точности (_float128, _Quad). Соответствующие машинные эпсилон равны $2^{-112} \approx 1.93e\text{-}34$ и $2^{-113} \approx 9.63e\text{-}35$.

Практическая часть

Для решения поставленных задач необходимо инициализировать описанные функции:

- Вычисления по встроенным функциям — требуется использовать встроенные в math.h функцию sin.
- Вычисления по формуле Тейлора — требуется создать цикл суммирования членов ряда Тейлора, пока нововычисленный член не станет равным или меньшим машинному эпсилону.

Также нужно определить зависимые от пользователя переменные и учесть их при написании программы. (кол-во равных частей отрезка, коэффициент точности, начало и конец отрезка)

Название переменной	Тип переменной	Значение переменной
eps	long double	Переменная машинного нуля
i	int	Счетчик итераций
LDBL_EPSILON	long double	Встроенный машинный ноль
n	int	Количество равных частей отрезка
k	int	Коэффициент точности
a	long double	Начало отрезка
b	long double	Конец отрезка
x	long double	Текущее значение аргумента функции
Fx	long double	Аргумент расчет через встроенную функцию
Fn	int	Аргумент расчета факториала
Sk	int	Передаваемый коэффициент точности в функцию
Sx	long double	Передаваемое значение

		аргумента в функцию
sum	long double	Суммы членов ряда Тейлора
temp	long double	Переменная текущего члена ряда Тейлора

Алгоритм выполнения:

1. Вычисление машинного эпсилона.
2. Считывание зависимых переменных (n, k, a ,b).
3. Вычисление значения функции в каждой точке разбитого на равные части отрезка двумя способами.
4. Вывод полученных значений.

Исходный код

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

long double Function(long double Fx); //Расчет при помощи встроенных функций
long long Factorial(int Fn); //Расчет факториала
long double sumT(int Sn, long double Sx); //Расчет по формуле Тейлора

long double eps = 1; //Переменная машинного нуля
int i; //Переменная-счетчик итераций

void test() {

	assert(Factorial(2)==2);
	assert(Function(0)==0);
	assert(sumT(1, 0)==0);

}


int main() {

	while (LDBL_EPSILON != eps) { eps /= 2; } //Вычисление машинного нуля

	test();

	int n, k; //Кол-во частей отрезка, коэффициент точности
	long double a, b; //Начало и конец отрезка
	printf("Enter n k a b : ");
	scanf_s("%i%i%lf%lf", &n, &k, &a, &b);
	printf("x\t\tTaylor's Row\tFunction\titerations\n");

	long double x = a;

	for(int j = 0; j < 51; j++){

		printf("%lf\t%lf\t%lf\t%i\n", x, sumT(k, x), Function(x), i);
		x += (b - a) / n;

	}

	return 0;
}

long double Function(long double Fx) { return sin(Fx) * sin(Fx); }

long long Factorial(int Fn) {
```

```
        if (Fn == 0) { return 1; }
        return Fn*Factorial(Fn-1);

}

long double sumT(int Sk, long double Sx) {

        long double sum = 0;
        long double temp;

        for (i = 1; 1; i++) {

                temp = (pow(-1, i - 1) * pow(2, 2 * i - 1) * pow(Sx, 2 * i)) / Factorial(2 * i);
                if (pow(-1, i - 1)*temp <= eps) { return sum; }
                sum += temp;

        }
        return sum;

}
```

Входные данные: числа n, k, a, b соответственно.

Выходные данные: таблица с n + 1 строкой и 4мя столбцами, значение аргумента функции, значение функции, вычисленное по ряду Тейлора, значение функции, вычисленное встроенными мат. функциями, и количество потребовавшихся итераций соответственно.

Протокол исполнения программы

Тест I

Ввод: 5 5 0 1

Выход:

Enter n k a b : 5 5 0 1

x	Taylor's Row	Function	iterations
0.000000	0.000000	0.000000	1
0.200000	0.039470	0.039470	7
0.400000	0.151647	0.151647	9
0.600000	0.318821	0.318821	10
0.800000	0.514600	0.514600	11
1.000000	0.708073	0.708073	11

Тест II

Ввод: 10 10 0 1

Выход:

Enter n k a b : 10 10 0 1

x	Taylor's Row	Function	iterations
0.000000	0.000000	0.000000	1
0.100000	0.009967	0.009967	6
0.200000	0.039470	0.039470	7
0.300000	0.087332	0.087332	8
0.400000	0.151647	0.151647	9
0.500000	0.229849	0.229849	9
0.600000	0.318821	0.318821	10
0.700000	0.415016	0.415016	10
0.800000	0.514600	0.514600	11
0.900000	0.613601	0.613601	11
1.000000	0.708073	0.708073	11

Тест III

Ввод: 12 12 0 1

Выход:

Enter n k a b : 12 12 0 1

x	Taylor's Row	Function	iterations
0.000000	0.000000	0.000000	1
0.083333	0.006928	0.006928	6
0.166667	0.027522	0.027522	7
0.250000	0.061209	0.061209	8
0.333333	0.107056	0.107056	8
0.416667	0.163794	0.163794	9
0.500000	0.229849	0.229849	9
0.583333	0.303391	0.303391	10
0.666667	0.382381	0.382381	10
0.750000	0.464631	0.464631	11
0.833333	0.547862	0.547862	11
0.916667	0.629766	0.629766	11
1.000000	0.708073	0.708073	11

Тест IV

Ввод: 50 50 0 1

Выход:

Enter n k a b : 50 50 0 1

x	Taylor's Row	Function	iterations
0.000000	0.000000	0.000000	1
0.020000	0.000400	0.000400	4
0.040000	0.001599	0.001599	5
0.060000	0.003596	0.003596	5
0.080000	0.006386	0.006386	6
0.100000	0.009967	0.009967	6
0.120000	0.014331	0.014331	6
0.140000	0.019472	0.019472	7
0.160000	0.025382	0.025382	7
0.180000	0.032052	0.032052	7
0.200000	0.039470	0.039470	7
0.220000	0.047624	0.047624	7
0.240000	0.056503	0.056503	7

0.260000	0.066090	0.066090	8
0.280000	0.076372	0.076372	8
0.300000	0.087332	0.087332	8
0.320000	0.098952	0.098952	8
0.340000	0.111214	0.111214	8
0.360000	0.124097	0.124097	8
0.380000	0.137582	0.137582	9
0.400000	0.151647	0.151647	9
0.420000	0.166269	0.166269	9
0.440000	0.181424	0.181424	9
0.460000	0.197090	0.197090	9
0.480000	0.213240	0.213240	9
0.500000	0.229849	0.229849	9
0.520000	0.246890	0.246890	9
0.540000	0.264336	0.264336	10
0.560000	0.282159	0.282159	10
0.580000	0.300330	0.300330	10
0.600000	0.318821	0.318821	10
0.620000	0.337602	0.337602	10
0.640000	0.356642	0.356642	10
0.660000	0.375912	0.375912	10
0.680000	0.395381	0.395381	10
0.700000	0.415016	0.415016	10
0.720000	0.434788	0.434788	11
0.740000	0.454664	0.454664	11
0.760000	0.474613	0.474613	11
0.780000	0.494602	0.494602	11
0.800000	0.514600	0.514600	11
0.820000	0.534574	0.534574	11
0.840000	0.554493	0.554493	11
0.860000	0.574325	0.574325	11
0.880000	0.594038	0.594038	11
0.900000	0.613601	0.613601	11
0.920000	0.632982	0.632982	11

0.940000	0.652150	0.652150	11
0.960000	0.671075	0.671075	11
0.980000	0.689726	0.689726	11
1.000000	0.708073	0.708073	11

Заключение

Были изучены теории по темам машинного эпсилона, формулы Тейлора, описаны и созданы функции вычисления математической функции при помощи встроенных в язык программирования функций и при помощи формулы Тейлора. Составлена и собрана программа, собирающая зависимые переменные, вычисляющая и выводящая значения заданной функции.

Работа позволяет поближе познакомиться с точными вычислениями реализованных на программном уровне. Она развивает понимания и умения в сфере математического анализа и „точного“ программирования.

Источники

- Ряд Тейлора - https://ru.wikipedia.org/wiki/Ряд_Тейлора
- Машинный эпсилон - https://ru.wikipedia.org/wiki/Машинный_ноль