Московский авиационный институт

(Национальный исследовательский университет)

Факультет прикладной математики

Курсовой проект

по курсу

«Фундаментальная информатика»

I семестр

Задание 3

Студент: Тимофеева Ирина

Группа: М8О-111Б-23

Руководитель: Никулин С.П.

Оценка:

Дата: 24.12.23

Москва

2023 г.

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, находящихся в рекомендованной области хорошей точности формулы Тейлора. Вычисления по формуле Тейлора проводить по экономной в сложностном смысле схеме с точностью $\varepsilon * k$, 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 20

Функция: e^{2x}

Отрезок: $[0.1; 0.6]$

Ряд: $1 + \frac{2x}{1!} + \frac{2x^2}{2!} + \ldots + \frac{(2x)^n}{n!}$

Алгоритм решения

1. Деля единицу пополам до тех пор, пока её значение не превратится в машинный ноль, вычислить машинное эпсилон. Распечатать результат.
2. Запросить у пользователя необходимое число разбиений отрезка.
3. Разделить отрезок значений x на соответствующее число частей и вычислить шаг изменения переменной x.
4. Распечатать заголовок таблицы.
5. В цикле for обработать все значения x. Для того, чтобы избежать вычисления факториала, предыдущее значение ряда будем умножать на 2*x/n.

6. Пока значение вычисленного члена по модулю не будет меньше машинного эпсилон, для каждого значения x вычислять сумму членов ряда Тейлора.
7. Вывод результатов.

Листинг программного кода

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

//встроенная функция языка Си
double func_c(double x)
{
	return exp(2 * x);
}

int main()
{
	int n = 20; //число разбиений диапазона
	double a = 0.1;//начало диапазона
	double b = 0.6;//конец диапазона
	double x, y, summa, sum_current;

	//найти машинное эпсилон
	double eps = 1.0;
	while (1.0 + eps / 2.0 > 1) {
		eps /= 2.0;
	}

	printf("Машинное эпсилон = %.20f\n\n", eps);

	printf("Введите число разбиений отрезка [%1.1f, %1.1f]: ", a, b);
	scanf("%d", &n);

	double step = (b - a) / n;


printf("\n=========================================================================
===\n");
	printf("|| Шаг\t  x\t      Сумма ряда                Функция       Итераций
||\n");

printf("||-------------------------------------------------------------------------|
|\n");
	x = a;
	for (int i = 0; i <= n; i++) {
		summa = 0;
		int n = 1;
		sum_current = 1;
		do {
			summa += sum_current;
			sum_current *= 2 * x / n;
			n++;
		} while (fabs(sum_current) > eps && n < 100);
		printf("|| %2d\t %.2lf\t %.18lf\t %.18lf\t %2d\t||\n", i, x, summa,
func_c(x), n - 1);
		x += step;
	}

printf("=========================================================================
=\n");
```

```
        return 0;
}
```

Результат работы программы

```
Машинное эпсилон = 0,0000000000000022204

Введите число разбиений отрезка [0,1, 0,6]: 10

=============================================================================
|| Шаг    x          Сумма ряда                Функция          Итераций  ||
||---------------------------------------------------------------------------||
||  0    0,10    1,221402758160169633    1,221402758160169855    12        ||
||  1    0,15    1,349858807576002961    1,349858807576003183    13        ||
||  2    0,20    1,491824697641270125    1,491824697641270348    14        ||
||  3    0,25    1,648721270700127750    1,648721270700128194    15        ||
||  4    0,30    1,822118800390509108    1,822118800390508886    16        ||
||  5    0,35    2,013752707470476633    2,013752707470476633    16        ||
||  6    0,40    2,225540928492467430    2,225540928492467430    17        ||
||  7    0,45    2,459603111156949851    2,459603111156949407    18        ||
||  8    0,50    2,718281828459045535    2,718281828459045091    18        ||
||  9    0,55    3,004166023946432951    3,004166023946432507    19        ||
|| 10    0,60    3,320116922736547238    3,320116922736547238    20        ||
=============================================================================
```

```
Машинное эпсилон = 0,0000000000000022204

Введите число разбиений отрезка [0,1, 0,6]: 20

=============================================================================
|| Шаг    x          Сумма ряда                Функция          Итераций  ||
||---------------------------------------------------------------------------||
||  0    0,10    1,221402758160169633    1,221402758160169855    12        ||
||  1    0,12    1,284025416687741616    1,284025416687741394    12        ||
||  2    0,15    1,349858807576002961    1,349858807576003183    13        ||
||  3    0,17    1,419067548593257566    1,419067548593257122    13        ||
||  4    0,20    1,491824697641270125    1,491824697641270348    14        ||
||  5    0,22    1,568312185490168487    1,568312185490168709    14        ||
||  6    0,25    1,648721270700127750    1,648721270700127972    15        ||
||  7    0,27    1,733253017867395052    1,733253017867395052    15        ||
||  8    0,30    1,822118800390509108    1,822118800390508886    16        ||
||  9    0,33    1,915540829013895774    1,915540829013896218    16        ||
|| 10    0,35    2,013752707470477077    2,013752707470476633    16        ||
|| 11    0,38    2,117000016612674784    2,117000016612674784    17        ||
|| 12    0,40    2,225540928492468318    2,225540928492467874    17        ||
|| 13    0,43    2,339646851925991200    2,339646851925991200    17        ||
|| 14    0,45    2,459603111156951183    2,459603111156950295    18        ||
|| 15    0,48    2,585709659315847730    2,585709659315846842    18        ||
|| 16    0,50    2,718281828459045979    2,718281828459045979    18        ||
|| 17    0,53    2,857651118063164297    2,857651118063164741    19        ||
|| 18    0,55    3,004166023946434727    3,004166023946433839    19        ||
|| 19    0,58    3,158192909689768513    3,158192909689768957    19        ||
|| 20    0,60    3,320116922736549014    3,320116922736549014    20        ||
=============================================================================
```

Заключение

Таблица показывает, что значения ряда Тейлора имеют отличия от встроенной функции примерно после 15 знака после запятой. Это означает, что, несмотря на точность данного метода задания функций, он не является совершенным.

Такого рода программы не имеют прикладного применения, так как вычисление значения функции по ряду Тейлора требует много процессорного времени, что неэффективно в перспективе глобального применения, однако дают представление о том, как задаются математические функции на языке Си и прочих языках программирования.