

Московский Авиационный институт
(государственный технический университет)

Факультет прикладной математики и информационных технологий

Кафедра вычислительной математики и программирования

Курсовая работа

По курсу

«Информатика»

I семестр

Задание III. Вещественный тип. Приближенные вычисления. Табулирование функций

Студент: Арешин Станислав Олегович

Группа: М8О-106Б, №1 по списку

Руководитель: Дубинин А.В.

Оценка:

Дата:

Москва, 2017

Задача

Составить программу на Си, которая печатает таблицу значений элементарной функции, вычисленной двумя способами: по формуле Тейлора и с помощью встроенных функций языка программирования. Программа должна сама определять машинное эпсилон и обеспечивать корректные размеры генерируемой таблицы.

Вариант №22

22	$1 - \frac{x^2}{2} + \frac{x^3}{3} + \dots + (-1)^{n-1} \frac{x^{n-1}}{(n-1)!}$	0.0	1.0	$(1+x)e^{-x}$
----	---	-----	-----	---------------

Теория

Ряд Тейлора

Ряд Тейлора — разложение функции в бесконечную сумму степенных функций.

Ряд Тейлора был известен задолго до публикаций Тейлора — его использовали ещё в XVII веке Грегори, а также Ньютон.

Пусть функция $f(x)$ бесконечно дифференцируема в некоторой окрестности точки a .
Формальный ряд

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!} (x-a)^k$$

называется рядом Тейлора функции f в точке a .

То есть рядом Тейлора функции $f(x)$ в точке a называется ряд по положительным степеням

$$\begin{aligned} f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \dots = \\ = \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!}(x-a)^k. \end{aligned}$$

двучлена $(x-a)$

Свойства:

- Если f есть аналитическая функция в любой точке a , то её ряд Тейлора в любой точке a области определения f сходится к f в некоторой окрестности a .

- Существуют бесконечно дифференцируемые функции, ряд Тейлора которых сходится, но при этом отличается от функции в любой окрестности a . Коши

предложил такой пример:

$$f(x) = \begin{cases} 0, & x = 0 \\ e^{-\frac{1}{x^2}}, & x \neq 0 \end{cases}, \quad a = 0.$$

У этой функции все производные в нуле равны нулю, поэтому коэффициенты ряда Тейлора в точке $a = 0$ равны нулю.

Машинный эпсилон

Машинное эпсилон (англ. *Machine epsilon*) — 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).^[2] Формально машинное эпсилон обычно определяют как минимальное из чисел eps , для которого $1+\text{eps} > 1$ при машинных расчетах с числами данного типа^[3]. Альтернативное определение — максимальное положительное eps , для которого справедливо равенство $1+\text{eps}=1$.

Практическая важность машинного эпсилон связана с тем, что два (отличных от нуля) числа являются одинаковыми с точки зрения машинной арифметики, если их относительная разность по модулю меньше (при определении первого типа) или не превосходит (при определении второго типа) машинного эпсилон.

Тип double

Ключевое слово `double` обозначает простой тип, используемый для хранения 64-разрядных значений с плавающей запятой. В приведенной ниже таблице представлен точный и приблизительный диапазон значений для типа `double`.

Тип	Приблизительный диапазон значений	Точность
<code>double</code>	от $\pm 5,0 \times 10^{-324}$ до $\pm 1,7 \times 10^{308}$	15–16 знаков

Код на С

```
#include <stdio.h>

#include <math.h>

#define abs(x) ( (x) >= 0 ? (x) : (-x) )

double epsilon(){
    double a=1;
    while (1+a/2>1){
        a=a/2;
    }
    return(a);
}

int factorial(int n){
    if (n == 0)
        return 1;
    else
        return factorial(n - 1) * n;
}

int main(){
    double eps=2.71828182845904523536;
    double e,k;
    k=100000000;
    e=epsilon();
    printf("k=%lf\n", k);
    printf("Mashinnoe epsilon = %e\n", e);
    e=e*k;
    double a=0.0, b=1.0;
    int n,i=0;
```

```

printf("Vvedite n\n");

scanf("%d",&n);

printf(" Tablica znachenii\n");

printf("| x | Summa ryada |znachenie f(x) | raznica | iteracii | \n");

double x;

x=a;

while (i<=n){

    double function;

    function= (1+x)*(pow(eps,-1.0*x));

    double s=1.0;

    int it=1;

    double tay =100.0;

    while (!(s<=(function+e) && s>(function-e))){

        if (it%2==0)

            tay=-1*( (it-1) * (pow(x,it)) ) / (factorial(it));

        else

            tay=( (it-1) * (pow(x,it)) ) / (factorial(it));

        s=s+tay;

        it++;

    }

    double r;

    r=function-s;

    printf("|%5.2lf|%15.13lf|%15.13lf|%15.13lf|%15d|\n", x, s, function, abs(r), it);

    x+=(b-a)/n;

    i++;

}

}

```

Результаты

k=10000.000000

Mashinnoe epsilon = 2.220446e-16

Vvedite n

20

Tablica znachenii

x	Summa ryada	znachenie f(x)	raznica	iteracii	
0.00	1.0000000000000000	1.0000000000000000	0.0000000000000000	1	
0.05	0.9987908957248	0.9987908957257	0.000000000000009	7	
0.10	0.9953211598413	0.9953211598396	0.00000000000017	8	
0.15	0.9898141728880	0.9898141728888	0.00000000000008	9	
0.20	0.9824769036938	0.9824769036936	0.00000000000002	10	
0.25	0.9735009788392	0.9735009788393	0.00000000000001	11	
0.30	0.9630636868858	0.9630636868862	0.00000000000004	11	
0.35	0.9513289211203	0.9513289211203	0.00000000000001	12	
0.40	0.9384480644503	0.9384480644499	0.00000000000004	12	
0.45	0.9245608198531	0.9245608198516	0.00000000000015	12	
0.50	0.9097959895687	0.9097959895690	0.00000000000002	13	
0.55	0.8942722060890	0.8942722060898	0.00000000000008	13	
0.60	0.8780986177506	0.8780986177504	0.00000000000001	14	
0.65	0.8613755316560	0.8613755316557	0.00000000000003	14	
0.70	0.8441950164464	0.8441950164454	0.00000000000010	14	
0.75	0.8266414672966	0.8266414672968	0.00000000000001	15	
0.80	0.8087921354106	0.8087921354110	0.00000000000004	15	
0.85	0.7907176241043	0.7907176241051	0.00000000000009	15	
0.90	0.7724823535051	0.7724823535071	0.00000000000021	15	
0.95	0.7541449957366	0.7541449957363	0.00000000000003	16	
1.00	0.7357588823436	0.7357588823429	0.00000000000007	16	

k=100000.000000

Mashinnoe epsilon = 2.220446e-16

Vvedite n

20

Tablica znachenii

x	Summa ryada	znachenie f(x)	raznica	iteracii	
0.00	1.000000000000000	1.000000000000000	0.000000000000000	1	
0.05	0.9987908957248	0.9987908957257	0.000000000000009	7	
0.10	0.9953211598413	0.9953211598396	0.000000000000017	8	
0.15	0.9898141728880	0.9898141728888	0.000000000000008	9	
0.20	0.9824769036825	0.9824769036936	0.000000000000110	9	
0.25	0.9735009788416	0.9735009788393	0.000000000000023	10	
0.30	0.9630636869004	0.9630636868862	0.000000000000142	10	
0.35	0.9513289211179	0.9513289211203	0.000000000000023	11	
0.40	0.9384480644398	0.9384480644499	0.000000000000101	11	
0.45	0.9245608198531	0.9245608198516	0.000000000000015	12	
0.50	0.9097959895743	0.9097959895690	0.000000000000054	12	
0.55	0.8942722061066	0.8942722060898	0.000000000000168	12	
0.60	0.8780986177480	0.8780986177504	0.000000000000024	13	
0.65	0.8613755316489	0.8613755316557	0.000000000000068	13	
0.70	0.8441950164277	0.8441950164454	0.000000000000177	13	
0.75	0.8266414672993	0.8266414672968	0.000000000000025	14	
0.80	0.8087921354172	0.8087921354110	0.000000000000062	14	
0.85	0.7907176241196	0.7907176241051	0.000000000000144	14	
0.90	0.7724823535051	0.7724823535071	0.000000000000021	15	
0.95	0.7541449957316	0.7541449957363	0.000000000000047	15	
1.00	0.7357588823329	0.7357588823429	0.000000000000100	15	

Заключение

Проанализировав полученные результаты можно сделать следующие выводы:

- Результаты, полученные вычислениями по формуле Тейлора незначительно, но отличаются от результатов, полученных с помощью встроенных функций языка программирования;
- Это в свою очередь указывает на наличие погрешности в расчетах по формуле Тейлора;
- Чем больше k , тем больше разница между вычислениями, то есть больше погрешность.

Список информационных ресурсов

- https://ru.wikipedia.org/wiki/%D0%A0%D1%8F%D0%B4_%D0%A2%D0%B5%D0%B9%D0%BB%D0%BE%D1%80%D0%B0%D0%A4%D0%BE%D1%80%D0%BC%D1%83%D0%BB%D0%B0_%D0%A2%D0%B5%D0%B9%D0%BB%D0%BE%D1%80%D0%B0 (Ряд Тейлора)
- http://ru.math.wikia.com/wiki/%D0%A0%D1%8F%D0%B4_%D0%A2%D0%B5%D0%B9%D0%BB%D0%BE%D1%80%D0%B0 (Ряд Тейлора)
- http://cyclowiki.org/wiki/%D0%A0%D1%8F%D0%B4_%D0%A2%D0%B5%D0%B9%D0%BB%D0%BE%D1%80%D0%B0 (Формулы для ряда Тейлора)
- https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BD%D0%BE%D0%BB%D1%8C (Машинный эпсилон)
- <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/double> (Тип double)