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 «Вычислительные системы»
I семестр
Задание 3
«Вещественный тип. Приближенные вычисления. Табулирование функций»

Группа:	М8О-107Б-20
Студент:	Алапанова Эльза Халилевна
Преподаватель:	
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 ε * 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 16:
Ряд Тейлора — $x + 3x^2 + \cdots + \frac{2n+1}{n!}x^{2n}$
Значения a и b – 0.0, 1.0
Функция — $(1 + x^2)e$

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.

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

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 + ε = 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 машинные эпсилон определено для следующих типов: float – 1.19 * 10^{-7}, double – 2.20 * 10^{-16}, long double – 1.08 * 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 за $O(\log(10^{16}))\sim O(1)$ Для каждой N+1 строки нужно просуммировать i членов формулы Тейлора, пока $|A_1-A_2|> \varepsilon$. Проще всего задать рекуррентное соотношение между членами: $T_{i+1} = (-1)\frac{T_i x^2}{n(n-1)}$ где, T_i – значение i-того члена формулы Тейлора, n – счётчик факториала, x – значение для которого проводятся вычисления. В общем случае сумма может быть вычислена за O(log k). Асимптотика решения O(1 + (n+1) log k) ~ O(n log k)

Описание программы

Использованные в программе переменные

Название переменной	Тип переменной	Смысл переменной
n	int	То самое число N, на которое нужно разбить отрезок
k	int	То самое число K, используемое для вычисления точности. Так же обозначает, что вывод должен быть с точность до K знаков после запятой
e0	double	То самое машинное эпсилон. В случае с double $\varepsilon = 2.20 * 10^{-16}$
d	double	Формально разница между предыдущим значением из отрезка и следующим, если отрезок разбит на n равных частей.
x	double	Переменная, для которой будем производить вычисления
ans1	double	То самое значение A_1, вычисленное с помощью формулы Тейлора
ans2	double	То самое значение A_2, вычисленное с помощью встроенных функций языка
count	double	Счётчик члена формулы Тейлора. Так как для sin(x) все члены нечётные, то будем пропускать чётные
cur	double	То самое значение T_i. Текущие значение члена формулы Тейлора. При чём T_0=x
step	int	Число итераций, которые затрачены на вычисление A_2

Использованные в программе функции

Название функции	Тип возвращаемой переменной	Смысл функции
do_nothing	double	Копирует значение в память, где double выделяется 64 бита, а не 80 бит

Программа

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

double do_nothing(double x) {
   return x;
}

int main()
{
   int n, k;
   scanf("%d %d", &n, &k);
   double e0 = 1.0, d = 1.0 / (double)n;
   while (do_nothing(1.0 + e0 / 2.0) > 1.0) {
      e0 = e0 / 2.0;
   }
   printf("Machine epsilon equals %.8e\n", e0);
   printf("-------------------------------------\n");
   double x = 0.0, ans1, ans2, count, cur, cur0;
   double e = 2.7182818;
   int step;
   for (int i = 0; i <= n; i++) {
      ans1 = 0;
      ans2 = (1 + 2 * x * x) * pow(e, x * x);
      step = 0;
      count = 1.0;
      cur = 1.0;
      double copy = x;
      if (copy == 0) {
         step++;
         ans1 = 1;
      } else if (step == 0) {
         ans1 = 1;
      }
      while(fabs(ans2) > fabs(ans1)) {

         cur *= count;
         cur0 = ((2 * count + 1) * pow(x, 2 * count)) / cur;
         ans1 += cur0;
         step++; count++;
      }
      printf("%.2f | %.*f | %.*f | %d | %.*f\n", x, k+1, ans1, k+1, ans2, step, k+1, fabs(ans2-
ans1));
      x += d;
   }
   return 0;
}
```

Входные и выходные данные

Входные данные

Единственная строка содержит два целых числа N ($0 \le N \le 100$) – число разбиений отрезка на равные части, K ($0 \le K \le 16$) — коэффициент для вычисления точности формулы Тейлора.

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 K знаков после запятой.

Протокол исполнения и тесты

Тест #1

Ввод:
2 8
Вывод:
Machine epsilon equals 2.22044605e-016

0.00 | 1.000000000 | 1.000000000 | 1 | 0.000000000
0.50 | 1.926038125 | 1.926038120 | 8 | 0.000000005
1.00 | 8.154845428 | 8.154845400 | 11 | 0.000000028

Тест #2

Ввод:
10 6
Вывод:
Machine epsilon equals 2.22044605e-016

0.00 | 1.0000000 | 1.0000000 | 1 | 0.0000000
0.10 | 1.0302512 | 1.0302512 | 4 | 0.0000000
0.20 | 1.1240756 | 1.1240756 | 5 | 0.0000000
0.30 | 1.2911257 | 1.2911257 | 6 | 0.0000000
0.40 | 1.5490343 | 1.5490343 | 7 | 0.0000000
0.50 | 1.9260381 | 1.9260381 | 8 | 0.0000000
0.60 | 2.4653266 | 2.4653266 | 8 | 0.0000000
0.70 | 3.2319861 | 3.2319861 | 9 | 0.0000000
0.80 | 4.3239764 | 4.3239764 | 10 | 0.0000000
0.90 | 5.8895189 | 5.8895189 | 11 | 0.0000000
1.00 | 8.1548454 | 8.1548454 | 11 | 0.0000000

Тест #3

Ввод:
20 8
Вывод:
Machine epsilon equals 2.22044605e-016

0.00 | 1.000000000 | 1.000000000 | 1 | 0.000000000
0.05 | 1.007515643 | 1.007515643 | 3 | 0.000000000
0.10 | 1.030251170 | 1.030251170 | 4 | 0.000000000
0.15 | 1.068779011 | 1.068779010 | 5 | 0.000000000
0.20 | 1.124075636 | 1.124075636 | 5 | 0.000000000
0.25 | 1.197556266 | 1.197556265 | 6 | 0.000000001
0.30 | 1.291125655 | 1.291125654 | 6 | 0.000000001
0.35 | 1.407247303 | 1.407247303 | 6 | 0.000000001
0.40 | 1.549034350 | 1.549034347 | 7 | 0.000000002
0.45 | 1.720366418 | 1.720366416 | 7 | 0.000000002

0.50 | 1.926038125 | 1.926038120 | 8 | 0.000000005
0.55 | 2.171946470 | 2.171946464 | 8 | 0.000000006
0.60 | 2.465326588 | 2.465326584 | 8 | 0.000000004
0.65 | 2.815047899 | 2.815047888 | 9 | 0.000000011
0.70 | 3.231986111 | 3.231986099 | 9 | 0.000000012
0.75 | 3.729491127 | 3.729491124 | 9 | 0.000000003
0.80 | 4.323976400 | 4.323976376 | 10 | 0.000000024
0.85 | 5.035662616 | 5.035662595 | 10 | 0.000000021
0.90 | 5.889518921 | 5.889518875 | 11 | 0.000000045
0.95 | 6.916456255 | 6.916456206 | 11 | 0.000000049
1.00 | 8.154845428 | 8.154845400 | 11 | 0.000000028

Тест #4

Ввод:
4 16
Вывод:
Machine epsilon equals 2.22044605e-016

0.00 | 1.00000000000000000 | 1.00000000000000000 | 1 | 0.00000000000000000
0.25 | 1.19755626627140566 | 1.19755626549897864 | 6 | 0.00000000077242701
0.50 | 1.92603812482621950 | 1.92603811999044927 | 8 | 0.00000000483577023
0.75 | 3.72949112660596072 | 3.72949112407731631 | 9 | 0.00000000252864440
1.00 | 8.15484542849126193 | 8.15484540000000102 | 11 | 0.00000002849126091

Тест #5

Ввод:
50 12
Вывод:
Machine epsilon equals 2.22044605e-016

0.00 | 1.0000000000000 | 1.0000000000000 | 1 | 0.0000000000000
0.02 | 1.0012004000747 | 1.0012004000705 | 3 | 0.0000000000042
0.04 | 1.0048064047787 | 1.0048064047643 | 3 | 0.0000000000144
0.06 | 1.0108324544950 | 1.0108324544569 | 4 | 0.0000000000380
0.08 | 1.0193027064638 | 1.0193027063965 | 4 | 0.0000000000673
0.10 | 1.0302511704167 | 1.0302511703180 | 4 | 0.0000000000987
0.12 | 1.0437218997723 | 1.0437218996719 | 4 | 0.0000000001004
0.14 | 1.0597692400659 | 1.0597692398495 | 5 | 0.0000000002164
0.16 | 1.0784581354878 | 1.0784581352039 | 5 | 0.0000000002839
0.18 | 1.0998644974495 | 1.0998644970974 | 5 | 0.0000000003521
0.20 | 1.1240756360533 | 1.1240756356570 | 5 | 0.0000000003963
0.22 | 1.1511907587465 | 1.1511907583972 | 5 | 0.0000000003494
0.24 | 1.1813215394160 | 1.1813215393694 | 5 | 0.0000000000467
0.26 | 1.2145927638798 | 1.2145927630396 | 6 | 0.0000000008402
0.28 | 1.2511430486361 | 1.2511430476640 | 6 | 0.0000000009722
0.30 | 1.2911256546280 | 1.2911256535556 | 6 | 0.0000000010724
0.32 | 1.3347093833723 | 1.3347093822979 | 6 | 0.0000000010744

0.34	1.3820795745245	1.3820795736865	6	0.0000000008380
0.36	1.4334392080462	1.4334392079630	6	0.0000000000832
0.38	1.4890101239304	1.4890101217605	7	0.0000000021699
0.40	1.5490343495244	1.5490343471144	7	0.0000000024101
0.42	1.6137755864884	1.6137755839122	7	0.0000000025762
0.44	1.6835208198365	1.6835208172767	7	0.0000000025598
0.46	1.7585820947597	1.7585820926039	7	0.0000000021557
0.48	1.8392984633211	1.8392984623307	7	0.0000000009904
0.50	1.9260381248262	1.9260381199904	8	0.0000000048358
0.52	2.0192007440560	2.0192007387568	8	0.0000000052992
0.54	2.1192200391257	2.1192200334806	8	0.0000000056451
0.56	2.2265665729401	2.2265665672207	8	0.0000000057194
0.58	2.3417508307224	2.3417508254746	8	0.0000000052477
0.60	2.4653265875068	2.4653265837519	8	0.0000000037549
0.62	2.5978945972674	2.5978945968310	8	0.0000000004364
0.64	2.7401066519766	2.7401066410280	9	0.0000000109486
0.66	2.8926699558200	2.8926699441156	9	0.0000000117044
0.68	3.0563520532852	3.0563520411994	9	0.0000000120858
0.70	3.2319861106583	3.2319860989314	9	0.0000000117269
0.72	3.4204767649720	3.4204767549586	9	0.0000000100134
0.74	3.6228065304564	3.6228065245224	9	0.0000000059341
0.76	3.8400428534734	3.8400428317031	10	0.0000000217703
0.78	4.0733457523696	4.0733457290011	10	0.0000000233685
0.80	4.3239764003035	4.3239763758426	10	0.0000000244609
0.82	4.5933063788178	4.5933063542738	10	0.0000000245440
0.84	4.8828279314372	4.8828279086475	10	0.0000000227896
0.86	5.1941652234875	5.1941652056278	10	0.0000000178597
0.88	5.5290867290679	5.5290867214432	10	0.0000000076247
0.90	5.8895189206306	5.8895188751475	11	0.0000000454831
0.92	6.2775610878829	6.2775610398406	11	0.0000000480423
0.94	6.6955021277149	6.6955020785279	11	0.0000000491871
0.96	7.1458386154812	7.1458385677404	11	0.0000000477408
0.98	7.6312949322332	7.6312948904070	11	0.0000000418262
1.00	8.1548454284913	8.1548454000000	11	0.0000000284913

Вывод

В работе описано определение машинного эпсилон, приведены его значения ля 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.

Список литературы

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

Протокол исполнения и тесты