Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский Авиационный Институт»

Национальный Исследовательский Университет

Факультет №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

КУРСОВОЙ ПРОЕКТ

По дисциплине «Вычислительные системы»

По курсам: «Основы информатики», «Алгоритмы структуры и данных»

Задание: «Вещественный тип. Приближенные вычисления. Табулирование функций»

Студент:	Хренникова А. С.
Группа:	М80-108-19
Преподаватель:	Поповкин А. В.
Подпись:	
Оценка:	
Дата:	

Москва, 2019

Содержание

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 берутся 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 проводятся по экономной в сложностном смысле схеме с точностью ε*k, где ε – машинное эпсилон аппаратно реализованного вещественного типа для данной ЭВМ, а k – экспериментально подбираемый коэффициент, обеспечивающий приемлемую сходимость. Число итераций ограничено сверху числом порядка 100. Программа сама должна определять машинное ε и обеспечивать корректные размеры генерируемой таблицы.

Таблица 1 – Задание 23 варианта

№	Ряд	a	b	Функция
23	$x - \frac{x^3}{3} + \cdots + (-1)^n \frac{x^{2n+1}}{2n+1}$	0,0	0,5	arctg(x)

Общий метод решения

Необходимо вычислить значения заданной функции в нескольких точках. Используются два метода: 1) по формуле Тейлора, 2) непосредственно с помощью арифметических функций.

Рассмотрим элемент ряда Тейлора как дробь:

$$T_i = \frac{U_i}{D_i}$$

где T – элемент ряда Тейлора, U и D – составные части дроби, i – номер элемента в ряде Тейлора.

Таблица 2 – Анализ значений U и D в зависимости от i

I	0	1	2	3	4	5	6	…	p
U	x	$-x^3$	x^5	$-x^7$	x^9	$-x^{11}$	x^{13}	…	$(-1)^n x^{(2n+1)}$
D	1	3	5	7	9	11	13	…	2n+1

Очевидно, что

$U_{i+1}=U_i*(-1)*x^2$, $U_0=x$;

а $D_{i+1}=D_i+2$, $D_0=1$.

На каждом этапе высчитываем элемент по указанной рекуррентной формуле, что позволяет сократить временные издержки вычислений. Чтобы достичь максимальной точности, будем продолжать вычисления до выполнения ($\frac{U_i}{D_i} < \varepsilon$) и ($i \geq 100$), где Ɛ - машинное эпсилон, минимальное число, выразимое на конечной вычислительной машине. Число итераций должно ограничиваться сверху числом порядка 100 по условию выполнения работы.

На печать нужно вывести столбец аргументов, соответствующие им столбцы значений функции, вычисленные с помощью формулы Тейлора и вычисленные арифметически. Для каждой строчки надо указать, на каком шаге закончились вычисления по формуле Тейлора – по ней результат

вычисляется с какой-либо точностью. Точность зависит от переменной k – чем она меньше, тем больше n-ное слагаемое приближено к ε*k.

Для рисования границ таблицы нужно использовать символы -|_ и др.

Для проведения арифметических вычислений достаточно подставлять в цикле текущее значение аргумента в исходную формулу функции. А значение по формуле Тейлора надо вычислять во вложенном цикле – каждый раз считается новое слагаемое, уточняющее результат.

Общие сведения о программе

Необходимое программное и аппаратное обеспечение: ОС семейства UNIX (Linux Ubuntu), среда программирования Си (язык Си, компилятор gcc), процессор с 64-битной архитектурой (как на лабораторном компьютере).

Система программирования: GUN C.

Строк в программе: 60.

Местонахождение файлов на домашнем компьютере: /home/lina_tucha/dir/kp.c – файл с программой. Сам файл компилируется с помощью написания «gcc kp.c -lm -o 123» в командной строке интерпретатора команд. Запуск файла осуществляется вызовом исполняемого файла «./123». -lm - это математическая библиотека, -o задает имя исполняемого файла.

Функциональное назначение

Программа предназначена для выполнения вещественных вычислений значений неалгебраических функций в алгебраической форме с помощью разложения по методу Тейлора. Ряд Тейлора — разложение функции в бесконечную сумму степенных функций. Если функция *f*(*x*) имеет непрерывные производные вплоть до (n+1)-го порядка, то ее можно разложить в степенной ряд по формуле Тейлора: $f(x) = \sum_{i=0}^{n} \frac{f^{(i)}(x_0)}{i!}(x - x_0)^i$. Также следует учесть, что точность вычислений зависит от диапазона вещественных чисел в ЭВМ.

Описание логической структуры

- Объявляем переменные.
- Создаём вспомогательные функции (для рисования линий, для арифметического вычисления заданной функции).
- Цикл - вычисляем машинное эпсилон.
- Запускаем цикл, последовательно проходящий по каждому значению отрезка [a;b] (в соответствии с введённым n). Каждый раз вычисляется функция – арифметическим способом и по формуле Тейлора. Для вычисления по формуле Тейлора запускаем вложенный цикл – он прекращается, когда будет достигнута приемлемая точность вычислений, либо когда количество итераций (точнее, членов последовательности) превысит 100.
- И каждый раз после всех вычислений выводим на печать очередную строку таблицы.

Описание переменных, функций, входные и выходные данные

Переменные и константы вспомогательных функций

Таблица 3 - Переменные и константы функции main(основы программы)

Имя	Тип	Назначение
a	double	Начало отрезка
b	double	Конец отрезка
h	double	Шаг (по отрезку [a;b])
x	double	Текущая точка на отрезке, аргумент функции
t	double	Результат вычисления функции по формуле Тейлора
cur	double	Текущее слагаемое в формуле Тейлора
eps	double	Машинное эпсилон
n	int	Количество частей, на которые делится отрезок
it	int	Номер слагаемого в формуле Тейлора
iter	double	Счётчик для расчёта формулы Тейлора
i	int	Запасной счётчик
k	int	Константа, регулирующая точность вычислений по формуле Тейлора
s	int	Знак каждого i-го слагаемого в формуле Тейлора

Вспомогательные функции

Таблица 4 - Функция line, рисующая внутренние границы таблицы

Имя	Тип	Вид	Назначение
i	int	Локальная переменная	Счётчик цикла
line()	void	Возвращаемое значение	Результат функции

Таблица 5 - Функция clearline, рисующая внешние границы таблицы

Имя	Тип	Вид	Назначение
i	int	Локальная переменная	Счётчик цикла
clearline()	void	Возвращаемое значение	Результат функции

Таблица 6 - Функция, вычисляющая арифметическое значение заданной по условию функции

Имя	Тип	Вид	Назначение
a	double	Входной параметр	Аргумент вычисляемой функции f(x)
f(double a)	double	Возвращаемое значение	Результат вычисляемой функции

Встроенные функции

Функция atan: вычисляет арктангенс и возвращает значение арктангенса параметра с плавающей точкой в интервале [-pi/2,+pi/2] . В тригонометрии, арктангенс является обратной тригонометрической функцией тангенса. В Си, определён только один прототип этой функции, с типом данных double.

Функция fabs: вычисляет абсолютное значение (модуль) и возвращает его |x|. В Си, определён только один прототип данной функции, с типом данных double.

Входные данные:

Единственная вводимая переменная – n типа int. После приглашения нужно вводить значения, допускаемые типом int.

Выходные данные:

Даётся приглашение на ввод. После вычисления машинного эпислон оно выводится на экран, рядом же печатается значение k.

Остальные выходные данные представляются в виде таблицы с заголовком, нарисованной символами. Она включает в себя аргументы x, значения функции, вычисленные по формуле Тейлора, значения функции, вычисленные арифметически, количество слагаемых в формуле Тейлора – оно же количество итераций, сделанное для достижения определённой точности вычислений.

Выходные данные имеют форматирование вывода – количество позиций под всё число и количество позиций под дробную часть (для чисел со спецификатором %lf).

Таблица значений

Таблица 7 – Значения функций при разбиении интервала на 30 частей

x	Taylor	f(x)	iter
0.0000	0.000000000000000	0.000000000000000	1
0.0167	0.016665123199537	0.016665123713941	4
0.0333	0.033320979417331	0.033320995878247	5
0.0500	0.049958270721509	0.049958395721943	6
0.0667	0.066567637021072	0.066568163775824	6
0.0833	0.083139624335084	0.083141231888441	7
0.1000	0.099664652268924	0.099668652491162	7
0.1167	0.116132980413911	0.116141626879990	8
0.1333	0.132534673369327	0.132551532296674	8
0.1500	0.148859564063533	0.148889947609497	9
0.1667	0.165097215022727	0.165148677414627	9
0.1833	0.181236877201322	0.181319774401490	10
0.2000	0.197267445945918	0.197395559849881	10
0.2167	0.213177413614167	0.213368642151808	11
0.2333	0.228954818309129	0.229231933276995	11
0.2500	0.244587188117005	0.244978663126864	12
0.2667	0.260061480149298	0.260602391747341	12
0.2833	0.275364013586606	0.276097019395436	13
0.3000	0.290480395796888	0.291456794477867	14
0.3167	0.305395440451976	0.306676319401683	14
0.3333	0.320093076386694	0.321750554396642	15
0.3500	0.334556245728604	0.336674819386727	16
0.3667	0.348766789564400	0.351444794003552	16
0.3833	0.362705319090480	0.366056515847418	17
0.4000	0.376351069806398	0.380506377112365	18
0.4167	0.389681735832882	0.394791119699761	19

Продолжение таблицы 7

0.4333	0.402673280847994	0.408907828950925	19
0.4500	0.415299721405948	0.422853926132941	20
0.4667	0.427532877494093	0.436627159813541	21
0.4833	0.439342084043170	0.450225596260715	22
0.5000	0.450693855665945	0.463647609000806	23

Протокол

```
lina_tucha@LAPTOP-44CRFC1U:~$ cd dir
lina_tucha@LAPTOP-44CRFC1U:~/dir$ nano kp.c
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc kp.c -lm -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ cat kp.c
#include <stdio.h>
#include <math.h>

void line() {
	for(int i=0;i<64;i++) {
		if(i!=10 && i!=33 && i!=56 && i!=63) {
		   printf("_");
		}
		else {
		   printf("|");
		}
	}
	printf("\n");
}

void clearline() {
	for(int i=0;i<63;i++) {
	   printf("_");
	}
	printf("\n");
}

double f(double a) {
   return atan(a);
```

```
}

int main(){
double a=0.0,b=0.5,h,x=a,t,iter,cur,eps=1.0;
int n,i,k=10,p,s,it;
printf("Введите количество частей отрезка: ");
scanf("%d",&n);
h=(b-a)/n;

while ((eps/2+1)>1) {eps/=2;}

printf("Машинное эпсилон для double = %.30lf с точностью вычислений по формуле Тейлора k = %d\n",eps,k);
printf("Таблица значений ряда Тейлора и стандартной функции для f(x)=arctg(x)\n");

clearline();
printf("          x  |                  Taylor  |                  f(x)  |  iter |\n");
line();

while(x<=b)
   { it=0; t=x-a; p=1; s=1;
     for(iter=1;iter<=100;iter++){
          s*=-1;
          p*=iter;
          cur=((s*pow(x,2*iter+1))/(2*iter+1));
          t+=cur; s*=-1;
          it++;
          if(fabs(cur)<eps*k) break;
```

```
            }
        printf("%9.4lf |%21.15lf |%21.15lf |%4d  |\n",x,t,f(x),it);
        line();
        x+=h;
        }
}
```

```
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
Введите количество частей отрезка: 1
Машинное эпсилон для double = 0.000000000000000222044604925031 с
точностью вычислений по формуле Тейлора k = 10
Таблица значений ряда Тейлора и стандартной функции для f(x)=arctg(x)
_____________________________________________________________
        x  |                 Taylor  |                   f(x)  |   iter  |
___________|_________________________|_________________________|_________|
   0.0000  |  0.000000000000000      |  0.000000000000000      |    1    |
___________|_________________________|_________________________|_________|
   0.5000  |  0.450693855665945      |  0.463647609000806      |   22    |
___________|_________________________|_________________________|_________|
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
Введите количество частей отрезка: 2
Машинное эпсилон для double = 0.000000000000000222044604925031 с
точностью вычислений по формуле Тейлора k = 10
Таблица значений ряда Тейлора и стандартной функции для f(x)=arctg(x)
_____________________________________________________________
        x  |                 Taylor  |                   f(x)  |   iter  |
___________|_________________________|_________________________|_________|
   0.0000  |  0.000000000000000      |  0.000000000000000      |    1    |
___________|_________________________|_________________________|_________|
   0.2500  |  0.244587188117005      |  0.244978663126864      |   11    |
```

```
_____|_____|_____|_____|
  0.5000 | 0.450693855665945 | 0.463647609000806 | 22 |
_____|_____|_____|_____|
```

lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123

Введите количество частей отрезка: 3

Машинное эпсилон для double = 0.000000000000000222044604925031 с точностью вычислений по формуле Тейлора k = 10

Таблица значений ряда Тейлора и стандартной функции для f(x)=arctg(x)

```
_____________________________________________________
        x |                Taylor |                f(x) |  iter |
_____|_____|_____|_____|
   0.0000 | 0.000000000000000 | 0.000000000000000 |  1 |
_____|_____|_____|_____|
   0.1667 | 0.165097215022727 | 0.165148677414627 |  9 |
_____|_____|_____|_____|
   0.3333 | 0.320093076386694 | 0.321750554396642 | 14 |
_____|_____|_____|_____|
   0.5000 | 0.450693855665945 | 0.463647609000806 | 22 |
_____|_____|_____|_____|
```

lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123

Введите количество частей отрезка: 5

Машинное эпсилон для double = 0.000000000000000222044604925031 с точностью вычислений по формуле Тейлора k = 10

Таблица значений ряда Тейлора и стандартной функции для f(x)=arctg(x)

```
_____________________________________________________
        x |                Taylor |                f(x) |  iter |
_____|_____|_____|_____|
   0.0000 | 0.000000000000000 | 0.000000000000000 |  1 |
_____|_____|_____|_____|
   0.1000 | 0.099664652268924 | 0.099668652491162 |  7 |
```

0.2000	0.197267445945918	0.197395559849881	10
0.3000	0.290480395796888	0.291456794477867	13
0.4000	0.376351069806398	0.380506377112365	17
0.5000	0.450693855665945	0.463647609000806	22

lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123

Введите количество частей отрезка: 10

Машинное эпсилон для double = 0.000000000000000222044604925031 с точностью вычислений по формуле Тейлора k = 10

Таблица значений ряда Тейлора и стандартной функции для f(x)=arctg(x)

x	Taylor	f(x)	iter
0.0000	0.000000000000000	0.000000000000000	1
0.0500	0.049958270721509	0.049958395721943	5
0.1000	0.099664652268924	0.099668652491162	7
0.1500	0.148859564063533	0.148889947609497	8
0.2000	0.197267445945918	0.197395559849881	10
0.2500	0.244587188117005	0.244978663126864	11
0.3000	0.290480395796888	0.291456794477867	13

0.3500	0.334556245728604	0.336674819386727	14
0.4000	0.376351069806398	0.380506377112365	17
0.4500	0.415299721405948	0.422853926132941	19
0.5000	0.450693855665945	0.463647609000806	22

Заключение

Цель задания достигнута – мы вычислили значения функции двумя способами и составили таблицу, научились понимать разницу между машинным и реальным вычислением функций. Судя по выходным данным, можно сказать, что программа работает корректно. В выводимой таблице можно видеть, что вычисленные значения функции по Тейлору приближены к арифметически вычисляемым значениям. Также, изменяя k можно менять и точность вычислений. Всё это помогает понять смысл и принципы машинной арифметики и вещественного представления чисел в ЭВМ.

Из-за того, что существует понятие ограниченности в разрядной сетке, вещественные числа имеют диапазон представления в памяти компьютера, что неизбежно приводит к тому, что в вычислениях в окрестности этих границ этого диапазона возникают погрешности.

На базе этой программы можно создать много аналогичных программ, выполняющих те же действия с другими функциями, но они не имеют прикладного применения, так как вычисление значения функции по ряду Тейлора требует много процессорного времени, что неэффективно в перспективе глобального применения.

Список использованных источников

1. РосДиплом, Оформление таблиц в дипломной работе, особенности и требования ГОСТ/Электронный диплом/Режим доступа: https://www.rosdiplom.ru/rd/pubdiplom/view.aspx?id=288
2. Диплом Журнал, Оформление курсовой работы по ГОСТу 2019(образец)/Электронный диплом/Режим доступа: https://journal.duplom.ru/kursovaya/oformlenie-kursovoj-raboty-po-gostu-2019-obrazec/
3. Vyuchit.work – универсальная методичка/Электронный диплом/Режим доступа: https://vyuchit.work/samorazvitie/sekretyi/oformlenie-risunkov-po-gostu.html
4. Архив вопросов и ответов для программистов/Электронный диплом/Режим доступа: https://qarchive.ru/320864_parametry_gcc__lm__lz__lrt___o_chem_oni_
5. Компилятор GCC/Электронный диплом/Режим доступа: http://parallel.uran.ru/book/export/html/25
6. StudFiles – файловый архив студентов/Электронный диплом/Режим доступа: https://studfile.net/preview/6262840/page:12/
7. АБИУМ24/Электронный диплом/Режим доступа: https://abium24.ru/gost-oformleniya-spiska-literatury#h50sjpy2l58g1vpe64bsnc9r51rskb1o
8. Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание. :Пер. с англ. – М. : Издательский дом «Вильямс», 2009. – 304 с. : ил. – Парал. тит. англ.