

Московский авиационный институт
(государственный технический университет)

Факультет прикладной математики

Кафедра вычислительной математики и программирования

КУРСОВАЯ РАБОТА

По курсам

**«Вещественный тип. Приближенные вычисления.
Табулированные функции»**

I семестр

Задание 3

«Вычисление функций по формуле Тейлора»

Студент: Чурилов С.Э. Группа:

М8О-103Б-20

Руководитель: Титов В.К.

Оценка:_____

Дата:_____

Москва, 2020

Содержание

Введение.....	3
Постановка задачи.....	3
Решение	4
Теоретическое обоснование.....	4
Программная реализация	5
Протокол	8
Вывод.....	10

Введение

В задании 3 необходимо составить программу на языке Си, которая печатает таблицу значений элементарной функции, вычисленной двумя способами: по формуле Тейлора и с помощью встроенных функций языка программирования.

Постановка задачи

Для вычисления значений элементарной функции в качестве аргументов таблицы необходимо 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 необходимо проводить по экономной в сложностном смысле схеме с точностью $\varepsilon \cdot k$, 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 обеспечить корректные размеры генерируемой таблицы.

№	ряд	a	b	функция
1	$\frac{x}{9} - \frac{x^3}{9^2} + \dots + (-1)^n \frac{x^{2n+1}}{9^{n+1}}$	-1.0	1.0	$\frac{x}{9 + x^2}$

Теоретические сведения

Машинное 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).

Если функция $f(x)$ имеет $n + 1$ производную на отрезке с концами a и x , то для произвольного положительного числа p найдётся точка ξ , лежащая между a и x , такая, что

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

Это формула Тейлора с остаточным членом в общей форме.

Решение

Теоретическое обоснование

Вначале необходимо вычислить машинное значение ε . Т.к. компьютер имеет ограниченную точность вычисления, то при достаточно маленьких значениях значения будут совпадать с некоторой погрешностью. Чтобы определить эту погрешность, т.е. эпсилон (переменная `eps` в коде), он будет вычисляться как:

```
while (1.+ eps > 1.) eps /= 2.;
```

В таком случае, каждый раз уменьшая эпсилон в два раза, можно получить наименьшее вещественное значение, которое означает минимальную разницу, которую компьютер в состоянии различить.

Следующим шагом, для определения количества точек разбиения отрезка возьмём $n = 10$, тогда шаг h для получения новой точки может быть вычислен как $h = \frac{(b-a)}{n} = 0.1$.

В каждой точке x отрезка будем вычислять значение по формуле Тейлора. Сперва необходимо определить как устроен член последовательности, т.к. вычисление значений заново для каждого члена последовательности увеличивает сложность вычислений многократно, из-за вычисления факториала и возведения в степень x . Чтобы определить

«разницу» между текущим и следующим членом последовательности

достаточно текущий разделить на предыдущий. Так мы получим значение, умножая на которое текущий член (слагаемое) последовательности можно получить следующий в формуле Тейлора. «Разница» элементов: $-x^*x/9$

Программная реализация

Далее необходимо написать программный код. Он будет содержать переменные и константы разных типов:

Переменная/константа	Тип	Значение
x	double	Неизвестная переменная
d		Вспомогательная переменная; счётчик суммы
S		Значение суммы ряда
S1		Значение заданной функции
eps		Машинное эpsilon
n	int	Количество итераций
a	const double	Начало отрезка
b		Конец отрезка
h		Вспомогательная константа

Подключение заголовочных файлов, функция взятия по модулю, объявление переменных и констант, инициализация некоторых из них:

```
#include <stdio.h>
#include <math.h>
double fabs(double d);
int main() {
    double x, d, S, S1, eps=1., a=-1.0, b=1.0, h=(b-
    a)/10.;
    int n;
    ...
}
```

Сначала выполняется вычисление машинного эпсилон, затем, выводится заголовок таблицы:

```
while(1.+eps>1.) eps/=2.;
printf("Машинное эпсилон = %.21f\n",eps);
printf("-----\n");
printf("|   x   |      S      |      S1      | n |\n");
printf("|-----|-----|-----|----|\n");
```

Затем объявляется цикл for, в котором последовательно берутся все значения x в отрезке [a; b] с шагом h.

```
for(x=a;x<0.001;x+=h) {...}
```

Внутри цикла переменным S, d и n присваиваются значения, для дальнейшего вычисления:

```
S=d=x/9; n=1;
```

После чего следует цикл вычисляющий непосредственно значения: S – функции по формуле Тейлора и d – новое слагаемое. В цикле также увеличивается значение n .

```
while(fabs(d)>eps){
    d=-d*x*x/9;
    S+=d; n++;
}
```

Условие цикла строится на том, что результат, вычисленный с помощью формулы Тейлора и встроенных функций, не должен отличаться более чем на εk . Если условие не выполняется, значит значение найдено с некой приемлемой погрешностью, после чего выводится новая часть таблицы с новыми значениями.

```
printf("| %.3f | %.17f | %.17f | %2d \n", x, S, x/(9+x*x), n);
```

После внешнего цикла (который идёт в отрезке $[a; b]$) выводится последняя часть таблицы и программа завершается командой возврата 0.

```
printf("-----\n");
return 0;
```

Компиляция программы происходит с помощью gcc с флагами -o, -lm полученный файл – main.exe. После запуска командой ./main.exe в терминальное окно выводится результат работы программы:

Машинное ϵ = 0.000000000000000111022

x	S	S1	n
-----	-----	-----	-----
-1.000	-0.09999999999999998	-0.10000000000000001	17
-0.800	-0.08298755186721991	-0.08298755186721991	14
-0.600	-0.06410256410256411	-0.06410256410256411	12
-0.400	-0.04366812227074236	-0.04366812227074236	10
-0.200	-0.02212389380530974	-0.02212389380530974	8
-0.000	-0.00000000000000001	-0.00000000000000001	1
0.000	0.00000000000000000	0.00000000000000000	1
0.200	0.02212389380530974	0.02212389380530974	8
0.400	0.04366812227074236	0.04366812227074236	10
0.600	0.06410256410256411	0.06410256410256411	12
0.800	0.08298755186721991	0.08298755186721991	14
1.000	0.09999999999999998	0.10000000000000001	17

```

azali@azali-VirtualBox:~/kurs$ cat Kurs3.cpp
#include <math.h>
#include <stdio.h>
int main()
{ double x, d, S, S1, eps=1., a=-1.0, b=1.0, h=(b-a)/10.;
  int n;
  while(1.+eps>1.) eps/=2.;
  printf("Машинное эпсилон = %.21f\n",eps);
  printf("-----\n");
  printf("|   x   |      S      |      S1      | n |\n");
  printf("|-----|-----|-----|----|\n");
  for(x=a;x<0.001;x+=h)
  { S=d=x/9; n=1;
    while(fabs(d)>eps)
    { d=-d*x*x/9;
      S+=d; n++;
    }
    printf("|  %.3f | %.17f | %.17f | %2d |\n", x, S, x/(9+x*x), n);
  }
  for(x=0;x<b+0.001;x+=h)
  {
    S=d=x/9; n=1;
    while(fabs(d)>eps)
    { d=-d*x*x/9;
      S+=d; n++;
    }
    printf("|  %.3f | %.17f | %.17f | %2d |\n", x, S, x/(9+x*x), n);
  }
  printf("-----\n");
  return 0;
}
azali@azali-VirtualBox:~/kurs$ c++ Kurs3.cpp
azali@azali-VirtualBox:~/kurs$ ./a.out

```

Протокол

9

Машинное ϵ = 0.000000000000000111022

x	S	S1	n
----- ----- ----- -----			
-1.000	-0.09999999999999998	-0.10000000000000001	17
-0.800	-0.08298755186721991	-0.08298755186721991	14
-0.600	-0.06410256410256411	-0.06410256410256411	12
-0.400	-0.04366812227074236	-0.04366812227074236	10
-0.200	-0.02212389380530974	-0.02212389380530974	8
-0.000	-0.00000000000000001	-0.00000000000000001	1
0.000	0.00000000000000000	0.00000000000000000	1
0.200	0.02212389380530974	0.02212389380530974	8
0.400	0.04366812227074236	0.04366812227074236	10
0.600	0.06410256410256411	0.06410256410256411	12
0.800	0.08298755186721991	0.08298755186721991	14
1.000	0.09999999999999998	0.10000000000000001	17

ВЫВОД

В процессе выполнения курсовой работы я написала программу на языке Си, которая выводит таблицу значений элементарной функции