

Московский авиационный институт
(национальный исследовательский университет)

Институт №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Курсовая работа

по курсу «Фундаментальная информатика»

1 семестр

Задание 3 «Вещественный тип. Приближенные вычисления. Табулирование функций»

Студент:	Дробышев Е. П.
Группа:	М8О-114БВ-24
Преподаватель:	Никулин С.П.
Подпись:	
Оценка:	
Дата сдачи:	7.12.2024

Содержание

Введение и формулировка задания	3
Вариант задания	3
Использованное оборудование и ПО	4
Описание алгоритма.....	4
Код программы.....	5
Протокол выполнения программы	6
Выводы.....	8

Введение и формулировка задания

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 $\text{eps} \cdot k$, где eps — машинное эpsilon 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 eps и обеспечивать корректные размеры генерируемой таблицы.

Вариант задания

Вариант 3:

3	$x - \frac{5}{2}x^2 + \dots + \frac{(-1)^{n+1} \cdot 2^n - 1}{n} x^n$	-0.2	0.3	$\ln(1 + x - 2x^2)$
---	---	------	-----	---------------------

Использованное оборудование и ПО

Оборудование ПЭВМ студента (лабораторное):

Процессор Intel Core i5, ОП 8ГБ, SSD 256ГБ, монитор 1920x1080 ~60Hz. Другие устройства не использовались.

Программное обеспечение ПЭВМ студента (лабораторное):

Операционная система семейства Linux, наименование Ubuntu версия 24.04

Интерпретатор команд GNU bash версия 5.2.21(1).

Редактор текстов: emacs версия 27.2

Прикладные системы и программы: emacs

Местоположение файлов /home/tru

Описание алгоритма

Вычисление машинного эпсилон: проверяем результат деления eps на два. Если он больше нуля, делим eps на два и продолжаем. Если результат меньше или равен нулю, останавливаем цикл. Далее получаем данные n и k от пользователя. Печатаем значение машинного эпсилон и верхнюю часть таблицы с заголовками.

Цикл `for` по значениям x : проходим все точки на отрезке $[a, b]$, поделенном на n равных частей. Присваиваем x текущее значение точки. Инициализируем переменные S и p (первый член ряда Тейлора).

Цикл `while`: выполняется, пока количество итераций не достигнет 100: увеличиваем количество итераций. Вычисляем следующий член ряда Тейлора. Если модуль этого члена больше $\text{eps} * k$, добавляем его к сумме S . Если модуль этого члена меньше $\text{eps} * k$, прерываем цикл `while`.

Печатаем строку таблицы для текущего значения x , сумма ряда Тейлора S , точное значение функции и количество итераций.

После завершения цикла `for`, выводим нижнюю строку таблицы и завершаем выполнение программы.

Код программы

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

int main() {
    double a = -0.2, b = 0.3, eps = 1.0, x, S, p;
    int n, k, iterations;

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

    printf("Введите количество частей n, на которые разбивается интервал: ");
    scanf("%d", &n);
    printf("Введите значение коэффициента точности k: ");
    scanf("%d", &k);

    printf("Машинное эpsilon для типа double: EPS = %.30f.\n", eps);
    printf("-----\n");
    printf("----\n");
    printf("|      x      |   Сумма ряда Тейлора   |   Значение\n");
    printf("функции      |   Итерации   |\n");
    printf("-----\n");
    printf("----\n");

    for (x = a; x < b + (b - a) / n; x += (b - a) / n) {
        S = 0;

        iterations = 1;

        while (iterations < 100) {
            p = (((pow(-1.0, iterations + 1) * pow(2.0, iterations)) - 1) /
iterations * pow(x, iterations));
            if (fabs(p) > eps) {
                S += p;
            } else {
                break;
            }

            iterations++;
        }

        printf("| %7.4f | %22.15f | %22.15f | %9d |\n", x, S, log(1 + x - 2 * (x
* x)), iterations);
    }
```

```

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

```

Протокол выполнения программы

Введите количество частей n , на которые разбивается интервал: 10
Введите значение коэффициента точности k : 1
Машинное эpsilon для типа double: EPS = 0.000000000000000222044604925031.

x	Сумма ряда Тейлора	Значение функции	Итерации
-0.2000	-0.328504066972036	-0.328504066972036	36
-0.1500	-0.216913001563574	-0.216913001563574	28
-0.0500	-0.056570351488394	-0.056570351488394	15
-0.0000	0.000000000000000	0.000000000000000	1
0.0500	0.044016885416774	0.044016885416774	15
0.1000	0.076961041136128	0.076961041136128	21
0.1500	0.099845334969716	0.099845334969716	28
0.2000	0.113328685307003	0.113328685307003	36
0.2500	0.117783035656383	0.117783035656383	47
0.3000	0.113328685307003	0.113328685307003	63

Введите количество частей n , на которые разбивается интервал: 5
Введите значение коэффициента точности k : 1
Машинное эpsilon для типа double: EPS = 0.000000000000000222044604925031.

x	Сумма ряда Тейлора	Значение функции	Итерации
-0.2000	-0.328504066972036	-0.328504066972036	36
-0.1000	-0.127833371509885	-0.127833371509885	21
0.0000	0.000000000000000	0.000000000000000	1
0.1000	0.076961041136128	0.076961041136128	21
0.2000	0.113328685307003	0.113328685307003	36
0.3000	0.113328685307003	0.113328685307003	63

Введите количество частей n , на которые разбивается интервал: 20
Введите значение коэффициента точности k : 1
Машинное эpsilon для типа double: EPS = 0.000000000000000222044604925031.

x	Сумма ряда Тейлора	Значение функции	Итерации
-0.2000	-0.328504066972036	-0.328504066972036	36
-0.1750	-0.269514768496332	-0.269514768496332	32
-0.1500	-0.216913001563574	-0.216913001563574	28
-0.1250	-0.169899036795397	-0.169899036795397	24
-0.1000	-0.127833371509885	-0.127833371509885	21
-0.0750	-0.090198267918149	-0.090198267918149	18
-0.0500	-0.056570351488394	-0.056570351488394	15
-0.0250	-0.026600681797179	-0.026600681797179	12
-0.0000	0.000000000000000	0.000000000000000	1
0.0250	0.023472356185142	0.023472356185142	12
0.0500	0.044016885416774	0.044016885416774	15
0.0750	0.061800400905447	0.061800400905447	18
0.1000	0.076961041136128	0.076961041136128	21
0.1250	0.089612158689687	0.089612158689687	24
0.1500	0.099845334969716	0.099845334969716	28
0.1750	0.107732699802882	0.107732699802882	32
0.2000	0.113328685307003	0.113328685307003	36
0.2250	0.116671306803693	0.116671306803693	41
0.2500	0.117783035656383	0.117783035656383	47
0.2750	0.116671306803693	0.116671306803693	54
0.3000	0.113328685307003	0.113328685307003	63

Выводы

В ходе выполнения данного задания курсового проекта я освоил реализацию вычислений значений функции с использованием ряда Тейлора. Я научился выводить данные в виде таблицы и оптимизировать форматирование чисел с плавающей точкой. Также я разобрался в алгоритмах, позволяющих вычислить машинное эпсилон.