

Министерство образования и науки Российской Федерации
Федеральное государственное образовательное учреждение высшего профессионального
образования

«Московский авиационный институт»

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

1 семестр

Курсовой проект

По курсу «Вычислительные системы»

задание 3

Работу выполнил : студент группы М8О-104Б-22

Жарков Александр Алексеевич

Руководитель : Потенко Максим Алексеевич

Москва, 2023

Содержание

ЦЕЛЬ	
РАБОТЫ.....	3
ОБЩИЙ МЕТОД	
РЕШЕНИЯ.....	4
ФУНКЦИОНАЛЬНОЕ	
НАЗНАЧЕНИЕ.....	5
ОПИСАНИЕ	
ПРОГРАММЫ.....	6
ТЕКСТ	
ПРОГРАММЫ.....	7
ТЕСТЫ.....	1
0	
ЗАКЛЮЧЕНИЕ.....	

ЦЕЛЬ РАБОТЫ

Составить программу на языке программирования Си, которая печатает таблицу значений элементарной функции, вычисленной двумя способами: по формуле Тейлора и с помощью функций из стандартной библиотеки языка Си. В качестве аргументов таблицы взять точки разбиения $[a, b]$ на n равных частей ($n+1$ точка включая концы отрезка), находящихся в рекомендованной области достаточной точности формулы Тейлора. Вычисления по формуле Тейлора проводить по экономной в сложностном смысле схеме с точностью $\varepsilon \cdot k$, где ε – машинное эпсилон аппаратно реализованного 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.

11 вариант задания:

Отрезок - $[0.1, 0.6]$

Функция: $(1 - (x^2 / 2)) * \cos(x) - (x / 2) * \sin(x)$

Разложение в ряд: $1 - (3 / 2) * x^2 + \dots + (-1)^n * (2n^2 + 1) / (2n)! * x^{2n}$

За количество x -ов на отрезке $[0.0, 1.0]$ взято число 15.

ОБЩИЙ МЕТОД РЕШЕНИЯ

Общий метод решения заключается в нахождении значения функции в некоторой точке при помощи двух способов.

Первый способ заключается в использовании функций, имеющихся в стандартной библиотеке «math.h» языка Си.

$$f(x) = f(a) + \frac{f'(a)}{1!} \cdot (x - a) + \frac{f''(a)}{2!} \cdot (x - a)^2 + \dots + \frac{f^{(n)}(a)}{n!} \cdot (x - a)^n + \dots$$

Основополагающей вещью в вычислении данной функции является наличие, так называемого, машинного эпсилон, которое является критерием точности вычислений на заданной ЭВМ.

Машинное эпсилон — минимальное число, выразимое на конечной вычислительной машине.

Его можно найти путём сравнения $1 + \varepsilon$ с 1 ($1 + \varepsilon = 1$). Последнее число, при стремлении к нулю, при котором данное выражение выдаст false и будет машинным эпсилон.

Я буду вычислять на каждом шаге итерации n -ное слагаемое ряда Тейлора и, в случае если данное слагаемое будет меньше $k \cdot \varepsilon$ (где k — экспериментально подобранный коэффициент), то далее вычислять ряд Тейлора является бессмысленным, т.к. члены ряда дошли до максимальной точности компьютера.

ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Программа предназначена для выполнения вещественных вычислений значений трансцендентных функций в алгебраической форме с использованием ряда Тейлора.

Ряд Тейлора – это разложение функции в бесконечную сумму степенных функций. Если функция $f(x)$ имеет непрерывные производные до $(n + 1)$ порядка, то ее можно разложить по формуле Тейлора.

Ранее данный метод использовался для аппаратного вычисления подобных функций, так как в то время компьютеры были способны только на сложение, вычитание и умножение. Но на сегодняшний день аппаратное обеспечение позволяет вычислять трансцендентные функции другими способами, которые более эффективны во всех смыслах.

ОПИСАНИЕ ПРОГРАММЫ

Программа работы:

- Подключаем заголовки «math.h» и «stdio.h»
- Определяем функцию вычисления машинного эпсилон
- Определяем функцию для вычисления члена ряда Тейлора
- Определяем функцию для вычисления функции при помощи встроенных функций
- Вычисляем машинное эпсилон и выводим.
- Печатаем таблицу аргументов функций, значений полученных средствами языка С и ряда Тейлора, количество итераций запрошенное машиной для вычисления значения функции

Название функции	Входные аргументы	Описание функции
compute_epsilon	-	Функция считает машинный epsilon, методом, описанным выше, а именно сравнивая $1+\epsilon$ и 1. Пока выражение $1 < 1 + \epsilon$ возвращает true, функция делит epsilon пополам.
inner_func	long double x	Функция вычисляет функцию, данную в задаче при помощи встроенных в язык программирования С средств. Используется функция powl, которая вычисляет степень числа для long double типа.
factorial	long long n	Функция вычисляет факториал числа n, данное во входных аргументах, путём итерирования от 2 до n включительно и умножения ans на i, где ans — ответ, а i — число, которое пробегает от 2 до n.
teylor_series	long double x, int n	Функция вычисляет ряд Тейлора в конкретной точке x.

printtab	long double k, long double a, long double b, int steps, int max_iters	Функция выводит таблицу с подсчитанными значениями
----------	---	--

long double k	Эмпирический коэффициент для eps
long double eps	Машинный эpsilon
long double a,b	Границы отрезка
int n	Кол-во итераций
int steps	Кол-во отрезков
int max_iters	Максимальное кол-во итераций
long double cur_member	I-ое слагаемое ряда
long double sum	Сумма ряда

Описание переменных и констант

Текст программы

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

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

```
/*
```

```
Жарков Александр М8О-104Б-22
```

```
Курсовой проект задание 3 вариант 11
```

```
*/
```

```
// функция подсчета машинного эpsilon. Пока единица меньше Эpsilon + 1, делим
эpsilon пополам и повторяем
```

```
long double compute_epsilon() {
```

```
    long double eps = 1;
```

```
    while (1 < 1 + eps) {
```

```
        eps /= 2;
```

```
    }
```

```
    return eps;
```

```
}
```

```

// функция, считающая значение заданной функции в точке икс
long double inner_func(long double x) {
    return (1 - powl(x, 2) / 2) * cos(x) - x / 2 * sin(x);
}

// функция подсчета факториала числа
int factorial(long long n) {
    long double ans = 1;
    for (long long i = 2; i <= n; ++i) {
        ans *= i;
    }
    return ans;
}

// функция, вычисляющая ряд Тейлора в этой точке икс
long double teilor_series(long double x, int n) {
    long double v = pow(-1, n);
    v *= (2 * n * n + 1);
    v /= 2 * (long double)factorial(n);
    v *= powl(x, 2 * n);
    return v;
}

// функция, выводящая таблицу с нужными значениями
void printtab(long double k, long double a, long double b, int steps, int max_iters) {
    long double step = (b - a) / steps; // считаем количество шагов
    long double eps = compute_epsilon(); // присваиваем переменной eps значение
    подсчитанного машинного эпсилон
    printf("Machine epsilon for long double for this system is %.20Lf\n", eps); // выводим
    машинный эпсилон

    printf("_____
    _____\n");
}

```



```

scanf("%d", &steps);

int max_iters = 100; // максимальное число итераций

printtab(k, a, b, steps, max_iters); // вывод

return 0;
}

```

Тесты

Введите количество разбиений отрезка 3

Machine epsilon for long double for this system is 0.00000000000000000005

x	Sum	$(1 - (x^2 / 2)) * \cos(x) - (x / 2) * \sin(x)$	n
0.10	0.4852234235204672631	0.9850374736192942837264066580083010649104835	27
0.27	0.4041589542080639156	0.8952201612476724878835442067437355717629544	47
0.43	0.2879932804240183933	0.7313823083746465693799274077058214516000589	74
0.60	0.1880935375088899656	0.5073824622074256174351740422689971410363796	100

Введите количество разбиений отрезка 10

Machine epsilon for long double for this system is 0.00000000000000000005

x	Sum	$(1 - (x^2 / 2)) * \cos(x) - (x / 2) * \sin(x)$	n
0.10	0.4852234235204672631	0.9850374736192942837264066580083010649104835	27
0.15	0.4673712023236472403	0.9664395433737418254381210114267730659776134	33
0.20	0.4435004051127123935	0.9405983132049106720893455468868182833830360	39
0.25	0.4146627972575107962	0.9077084136253717183695499670648132450878620	45
0.30	0.3821146285619005049	0.8680183161157527798744600044944519368073088	52
0.35	0.3472525971022231940	0.8218290178807729460543594901533737129284418	59
0.40	0.3115437692460468695	0.7694924460209241900846346973175116090715164	68
0.45	0.2764534786499768448	0.7114095880644415493296132957379285244314815	77
0.50	0.2433752447098153300	0.6480283570030254125526880670804530382156372	89
0.55	0.2135665116599015907	0.5798412001420741645812614861732470217248192	100
0.60	0.1880935375088899656	0.5073824622074256173809639336447219193360070	100

Введите количество разбиений отрезка 20

Machine epsilon for long double for this system is 0.00000000000000000005

x	Sum	$(1 - (x^2 / 2)) * \cos(x) - (x / 2) * \sin(x)$	n
0.10	0.4852234235204672631	0.9850374736192942837264066580083010649104835	27
0.12	0.4771058172499354702	0.9766539521175231419389262121910633140942082	30
0.15	0.4673712023236472403	0.9664395433737418254381210114267730659776134	33
0.18	0.4561278750493750483	0.9544134517385120074347767959999089271150297	36
0.20	0.4435004051127123935	0.9405983132049106720893455468868182833830360	39
0.23	0.4296280181687232918	0.9250201654939393254269489053065456118929433	42
0.25	0.4146627972575107962	0.9077084136253717183695499670648132450878620	45
0.28	0.3987677297412814405	0.8886957910114547034164528271915628465649206	48
0.30	0.3821146285619005049	0.8680183161157527798744600044944519368073088	52
0.33	0.3648819582345746802	0.845715244724270625025115688533614354749443	55
0.35	0.3472525971022231940	0.8218290178807729460543594901533737129284418	59
0.38	0.3294115679629383401	0.7964052055429707687589369058400734502356499	63
0.40	0.3115437692460468695	0.7694924460209241900846346973175116090715164	68
0.43	0.2938317384566893435	0.7411423812636447779316622996859109662182163	72
0.45	0.2764534786499768448	0.7114095880644415493296132957379285244314815	77
0.48	0.2595803772580791814	0.6803515052600526125917319852920428502329742	83
0.50	0.2433752447098153300	0.6480283570030254125526880670804530382156372	89
0.53	0.2279904979927962970	0.6145030721911576949371275313538376394717488	96
0.55	0.2135665116599015907	0.5798412001420741645812614861732470217248192	100
0.57	0.2002301558274473088	0.5441108226051924916049327973777138822697452	100
0.60	0.1880935375088899656	0.5073824622074256174893841508932723627367523	100
0.62	0.1772529582356356434	0.4697289874329606150157867539007838786346838	100

ЗАКЛЮЧЕНИЕ

В процессе выполнения данного курсового проекта были получены навыки вычисления и дальнейшего использования так называемого «машинного эпсилон». После генерации таблицы значений заданной функции можно увидеть, что значения совпадают до 10-14 знака после запятой. Из-за того, что существует понятие ограниченности разрядной сетки, вещественные числа имеют диапазон представления в памяти компьютера, что неизбежно приводит к тому, что в вычислениях в окрестности границ этого диапазона возникают погрешности.

На данный момент использование ряда Тейлора для вычисления трансцендентных функций является не оправданным, т. к. они требуют намного больше ресурсов, чем современные методы и имеют меньшую точность.

Полученные навыки и знания будут полезны в дальнейшем изучении программирования. Совмещение программирования с математическим аппаратом помогает решать множество разнообразных задач и отлично развивает мышление.

Источники

- 1) Численные методы. Линейная алгебра и нелинейные уравнения. Учебное пособие. — Directmedia, 2014-05-20. — 432 с.
- 2) Ильин В. А., Садовничий В. А., Сендов Б. Х. Математический анализ, ч. 1, изд. 3, ред. А. Н. Тихонов. М.: Проспект, 2004.
- 3) Романов Е. Си/Си++. От дилетанта до профессионала.