Московский авиационный институт  
(национальный исследовательский университет)  
  
Институт №8 «Компьютерные науки и прикладная математика»  
[Кафедра 806 «Вычислительная математика и программирование»](https://mai.ru/content/org/index.php?SECTION_ID=&ID=5042)

КУРСОВОЙ ПРОЕКТ  
по курсу "Архитектура компьютера и информационных систем"  
I семестр  
на тему «Вещественный тип. Приближенные вычисления. Табулирование функций»

Студент: Клименко В.М.  
Группа: М8О-103Б-22, № 11  
Руководитель: Никулин С.П., доцент 806 кафедры  
Москва, 2022

Оглавление

[Введение 3](#_Toc121042700)

[Представление вещественных чисел 4](#_Toc121042701)

[Практика 5](#_Toc121042702)

[Задание 5](#_Toc121042703)

[Описание программы 5](#_Toc121042704)

[Переменные 7](#_Toc121042705)

[Тесты 8](#_Toc121042706)

[Заключение 10](#_Toc121042707)

[Список литературы 10](#_Toc121042708)

## Введение

Вещественный тип данных очень важен в различных сферах жизни людей. Любая задача, связанная с математикой, практически всегда решается с применением этого представления чисел.

Вещественные числа очень важны в экономических, физических расчетах, благодаря чему всегда есть высокий спрос на людей, умеющих с ними работать, особенно высок спрос на программистов. Без знаний математики и знании о представлении чисел в компьютерной памяти невозможно стать хорошим программистом.

Цель данного проекта – изучение того, как представлены в памяти вещественные числа в языке программирования Си, что такое машинный эпсилон, узнать, отличаются ли значения, полученные рядом Тейлора, от значений, полученных встроенными функциями Си.

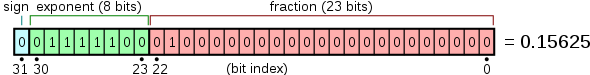
Проект будет реализован составлением программы на Си, которая будет печатать таблицу значений элементарной функции, заданной вариантом, вычисленной двумя способами: по формуле Тейлора и с помощью встроенных функций языка программирования.

## Представление вещественных чисел

В языке программирования Си вещественные числа представлены типами *float* и *double*, соответствующие стандартам IEEE 754 binary32 и IEEE 754 binary64.

1. В памяти компьютера стандарт IEEE 754 binary32 представляется при помощи:

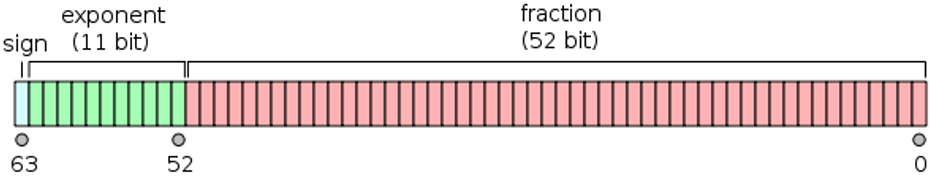
* Одного бита для знака
* Восьми бит для порядка
* Двадцати трех бит для значения

Представление числа, выраженного типом *float*:

Float в таком представлении вычисляется по формуле:

2. В памяти компьютера стандарт IEEE 754 binary64 представляется при помощи:

* Одного бита для знака
* Одиннадцати бит для порядка
* Пятидесяти трех бит для значения

Представление числа, выраженного типом *double*:

Double в таком представлении вычисляется по формуле:

где **e** – это значение бит порядка в десятичной системе счисления.

|  |  |  |
| --- | --- | --- |
| Тип | максимальное полож. значение | минимальное полож. значение |
| float | 1.175494351e-38 | 3.402823466е+38 |
| double | 2.2250738585072014е-308 | 1.7976931348623158е+308 |

## Практика

### 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 взять точки разбиения отрезка [a, b] на n равных частей, находящихся в рекомендованной области хорошей точности формулы Тейлора. Вычисления по формуле Тейлора проводить до экономной в сложностном смысле схеме с точностью 𝜀 ∗ 𝑘, где 𝜀 - машинный 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.

### Описание программы

Программы состоит из списка функций:

* *main* – главная функция программы, в которой происходит ввод и вывод

int main() {

    unsigned long long n\_t, n;

    long double eps = epsilon(), k = 1.5, a = 0.1, b = 0.6, x = a;

    printf("\nEnter number of x's iterations: ");

    scanf("%lld", &n);

    long double dx = (b - a) / (long double) n;

    printf("\n|x       |taylor's row           |native functions       |itrs\n");

    for (; x <= b + eps; x += dx) {

        printf("|%.5Lf ", x);

        printf("|%.20Lf ", taylor(x, &n\_t, eps, k));

        printf("|%.20Lf ", f(x));

        printf("|%3lld ", n\_t);

        printf("\n");

    }

    printf("\nMachine-calculated epsilon = %.30Lf\n", eps);

    printf("Coefficient = %Lf\n\n", k);

    return 0;

}

* *f* – функция, вычисляющая заданную математическую функцию при помощи встроенных в Си функций

long double f(long double x) {

    return (1 - x \* x/2) \* cos(x) - x/2 \* sin(x);

}

* *taylor* – функция, вычисляющая заданную математическую функцию при помощи ряда Тейлора

long double taylor(long double x, unsigned long long \*iters, long double eps, long double k) {

    long double sum = 0, curr, next = 1, temp;

    for (unsigned long long i = 0; i <= 1000; ++i) {

        \*iters = i;

        curr = next;

        next = pow(-1, i + 1) \* (2 \* pow(i + 1, 2) + 1) / f\_factorial(2 \* (i + 1)) \* pow(x, 2 \* (i + 1));

        temp = mabs(curr - next);

        if (temp < k \* eps) {

            break;

        }

        sum += curr;

    }

    return sum;

}

* *epsilon* – возвращает машинный эпсилон – минимальное положительное числовое значение данного типа, отличное от нуля. Формально, машинный эпсилон (*ε*) – это минимальное число, для которого выполняется условие 1 < 1 + *ε*.

long double epsilon() {

    long double e = 1;

    long double eps;

    while(1 < (1 + e)){

        eps = e;

        e /= 2;

    }

    return eps;

}

* вспомогательные функции *factorial* – возвращает факториал числаи *mabs* – возвращает абсолютное значение переменной типа *long double*

unsigned long long factorial(unsigned long long a) {

    unsigned long long answ = 1;

    for (unsigned long long i = 1; i <= a; ++i) {

        answ \*= i;

    }

    return answ;

}

long double mabs(long double a) {

    if (a < 0) {

        return -a;

    }

    return a;

}

### Переменные

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Тип | Начальное значение | Назначение |
| *n\_t* | *unsigned long long* | - | Счетчик итераций ряда Тейлора |
| *n* | *unsigned long long* | вводится пользователем | Кол-во шагов от *a* до *b* |
| *eps* | *long double* | *epsillon()* | Машинный эпсилон |
| *k* | *long double* | 1.5 | Коэффициент *eps* |
| *a* | *long double* | 0.1 | Начало заданного отрезка |
| *b* | *long double* | 0.6 | Конец заданного отрезка |
| *x* | *long double* | *a* | Значение *x* |
| *dx* | *long double* | (*b* - *a*) / (*long double*) *n* | Длина шага |

## Тесты

$ ./main.out

Enter number of x's iterations: 10

|x |taylor's row |native functions |itrs

|0.10000 |0.98503747361929422659 |0.98503747361929428291 | 6

|0.15000 |0.96643954337374187065 |0.96643954337374182490 | 7

|0.20000 |0.94059831320491066558 |0.94059831320491067209 | 8

|0.25000 |0.90770841362537176846 |0.90770841362537171940 | 8

|0.30000 |0.86801831611575281733 |0.86801831611575278231 | 8

|0.35000 |0.82182901788077299745 |0.82182901788077295028 | 9

|0.40000 |0.76949244602092421112 |0.76949244602092420749 | 9

|0.45000 |0.71140958806444156901 |0.71140958806444155806 | 9

|0.50000 |0.64802835700302537661 |0.64802835700302542388 | 9

|0.55000 |0.57984120014207430851 |0.57984120014207430358 | 10

|0.60000 |0.50738246220742561407 |0.50738246220742563473 | 10

Machine-calculated epsilon = 0.000000000000000000108420217249

Coefficient = 1.500000

$ ./main.out

Enter number of x's iterations: 25

|x |taylor's row |native functions |itrs

|0.10000 |0.98503747361929422659 |0.98503747361929428291 | 6

|0.12000 |0.97847768123838324903 |0.97847768123838325006 | 7

|0.14000 |0.97074386142465676969 |0.97074386142465671836 | 7

|0.16000 |0.96184531761927924523 |0.96184531761927922604 | 7

|0.18000 |0.95179276335662967419 |0.95179276335662970498 | 7

|0.20000 |0.94059831320491066558 |0.94059831320491067209 | 8

|0.22000 |0.92827547251790921349 |0.92827547251790923675 | 8

|0.24000 |0.91483912600503500403 |0.91483912600503505911 | 8

|0.26000 |0.90030552512758650521 |0.90030552512758650586 | 8

|0.28000 |0.88469227433001282763 |0.88469227433001286325 | 8

|0.30000 |0.86801831611575281733 |0.86801831611575278231 | 8

|0.32000 |0.85030391497804104511 |0.85030391497804101990 | 8

|0.34000 |0.83157064019686939674 |0.83157064019686934703 | 9

|0.36000 |0.81184134751408849331 |0.81184134751408844745 | 9

|0.38000 |0.79114015969941950071 |0.79114015969941950130 | 9

|0.40000 |0.76949244602092421112 |0.76949244602092420738 | 9

|0.42000 |0.74692480063425298071 |0.74692480063425293853 | 9

|0.44000 |0.72346501990575105460 |0.72346501990575100440 | 9

|0.46000 |0.69914207868525518781 |0.69914207868525518879 | 9

|0.48000 |0.67398610554515477072 |0.67398610554515473548 | 9

|0.50000 |0.64802835700302537661 |0.64802835700302542377 | 9

|0.52000 |0.62130119074586004916 |0.62130119074586007165 | 10

|0.54000 |0.59383803787463801242 |0.59383803787463805497 | 10

|0.56000 |0.56567337418866520604 |0.56567337418866512505 | 10

|0.58000 |0.53684269052980918886 |0.53684269052980917205 | 10

|0.60000 |0.50738246220742561407 |0.50738246220742563462 | 10

Machine-calculated epsilon = 0.000000000000000000108420217249

Coefficient = 1.500000

$ ./main.out

Enter number of x's iterations: 3

|x |taylor's row |native functions |itrs

|0.10000 |0.98503747361929422659 |0.98503747361929428291 | 6

|0.26667 |0.89522016124767253586 |0.89522016124767248935 | 8

|0.43333 |0.73138230837464651598 |0.73138230837464657719 | 9

|0.60000 |0.50738246220742561407 |0.50738246220742563473 | 10

Machine-calculated epsilon = 0.000000000000000000108420217249

Coefficient = 1.500000

$ ./main.out

Enter number of x's iterations: 1

|x |taylor's row |native functions |itrs

|0.10000 |0.98503747361929422659 |0.98503747361929428291 | 6

|0.60000 |0.50738246220742561407 |0.50738246220742563473 | 10

Machine-calculated epsilon = 0.000000000000000000108420217249

Coefficient = 1.500000

$ ./main.out

Enter number of x's iterations: 15

|x |taylor's row |native functions |itrs

|0.10000 |0.98503747361929422659 |0.98503747361929428291 | 6

|0.13333 |0.97345170366349904591 |0.97345170366349909882 | 7

|0.16667 |0.95862212006676651509 |0.95862212006676650425 | 7

|0.20000 |0.94059831320491066558 |0.94059831320491067209 | 8

|0.23333 |0.91944065584523185930 |0.91944065584523189649 | 8

|0.26667 |0.89522016124767253586 |0.89522016124767248935 | 8

|0.30000 |0.86801831611575281733 |0.86801831611575278231 | 8

|0.33333 |0.83792688872011576235 |0.83792688872011579699 | 8

|0.36667 |0.80504771256567969596 |0.80504771256567963860 | 9

|0.40000 |0.76949244602092421112 |0.76949244602092420738 | 9

|0.43333 |0.73138230837464651598 |0.73138230837464657719 | 9

|0.46667 |0.69084779283153373915 |0.69084779283153372831 | 9

|0.50000 |0.64802835700302537661 |0.64802835700302542377 | 9

|0.53333 |0.60307209149411254153 |0.60307209149411252972 | 10

|0.56667 |0.55613536722985524869 |0.55613536722985522321 | 10

|0.60000 |0.50738246220742561407 |0.50738246220742563473 | 10

Machine-calculated epsilon = 0.000000000000000000108420217249

Coefficient = 1.500000

## Заключение

В ходе данного проекта я узнал, как представляются числа с плавающей точкой в компьютере, в частности в языке программирования Си, изучил несколько стандартов представления вещественных чисел в ЭВМ. Научился работать с типами данных *float* и *double*. Вспомнил, что такое машинный эпсилон и где он используется. Получил опыт работы с рядами Тейлора в сфере программирования. Осознал важность знания математики в программировании.

## Список литературы

<https://wikipedia.org> – интернет-ресурс с описанием стандартов вещественных чисел и определением машинного эпсилон