Московский Авиационный Институт

(Национальный Исследовательский Университет)

Кафедра вычислительной математики и программирования

**Курсовая работа**

**по курсу «Фундаментальная информатика»**

**1 семестр**

**Задание №3:**

**Вещественный тип. Приближенные вычисления.**

**Табулирование функции.**

|  |  |
| --- | --- |
| Студент: | Кондратьев Е.А. |
| Группа: | М80-106Б-19 |
| Преподаватель: | Дубинин А.В. |
| Оценка: |  |
| Дата: |  |

**Оглавление**

[**Введение** 3](#_Toc28637873)

[**IEEE 754** 4](#_Toc28637874)

[**Стандарт описывает**: 4](#_Toc28637875)

[**Разработка стандарта** 4](#_Toc28637876)

[**Формат** 5](#_Toc28637877)

[**Представление и кодирование в памяти** 6](#_Toc28637878)

[**Форматы обмена** 7](#_Toc28637879)

[**Предикат общего порядка** 8](#_Toc28637880)

[**Ряд Тейлора** 8](#_Toc28637881)

[**Описание программы** 9](#_Toc28637882)

[**Заключение** 10](#_Toc28637883)

[**Список Литературы** 10](#_Toc28637884)

# **Введение**

Число с плавающей запятой (или число с плавающей точкой) — [экспоненциальная форма](https://ru.wikipedia.org/wiki/Экспоненциальная_запись) представления [вещественных (действительных) чисел](https://ru.wikipedia.org/wiki/Вещественное_число), в которой число хранится в виде [мантиссы](https://ru.wikipedia.org/wiki/Экспоненциальная_запись) и [порядка](https://ru.wikipedia.org/wiki/Экспоненциальная_запись) ([показателя степени](https://ru.wikipedia.org/wiki/Показатель_степени)). При этом число с плавающей запятой имеет фиксированную относительную [точность](https://ru.wikipedia.org/wiki/Точность) и изменяющуюся абсолютную. Используемое наиболее часто представление утверждено в стандарте [IEEE 754](https://ru.wikipedia.org/wiki/IEEE_754). Реализация математических операций с числами с плавающей запятой в вычислительных системах может быть как аппаратная, так и программная.

Так как в некоторых, преимущественно [англоязычных](https://ru.wikipedia.org/wiki/Английский_язык) и [англофицированных](https://ru.wikipedia.org/wiki/Англизация" \t "Англизация) странах при записи чисел целая часть отделяется от дробной точкой, то в терминологии этих стран фигурирует название «плавающая точка» ([англ.](https://ru.wikipedia.org/wiki/Английский_язык) floating point). Так как в [России](https://ru.wikipedia.org/wiki/Россия) целая часть числа от дробной традиционно отделяется запятой, то для обозначения того же понятия исторически используется термин «плавающая запятая», однако в настоящее время в русскоязычной литературе и технической документации можно встретить оба варианта.

Числа с плавающей запятой повсеместно используются в научных исследованиях, в математике, поэтому нужно уметь вычислять приближенное значение функций с помощью компьютера.

# **IEEE 754**

IEEE 754 — широко используемый стандарт [IEEE](https://ru.wikipedia.org/wiki/Институт_инженеров_электротехники_и_электроники), описывающий формат представления чисел с плавающей точкой. Используется в программных ([компиляторы](https://ru.wikipedia.org/wiki/Компилятор) с разных [языков программирования](https://ru.wikipedia.org/wiki/Язык_программирования)) и аппаратных ([CPU](https://ru.wikipedia.org/wiki/CPU) и [FPU](https://ru.wikipedia.org/wiki/FPU)) реализациях арифметических действий (математических операций).

**Стандарт описывает**:

* формат [чисел с плавающей точкой](https://ru.wikipedia.org/wiki/Число_с_плавающей_запятой): [мантисса](https://ru.wikipedia.org/wiki/Экспоненциальная_запись), экспонента (показатель), знак числа;
* представление [положительного и отрицательного нуля](https://ru.wikipedia.org/wiki/Отрицательный_и_положительный_ноль), положительной и отрицательной [бесконечностей](https://ru.wikipedia.org/wiki/Бесконечность), а также неЧисла (Not-a-Number, [NaN](https://ru.wikipedia.org/wiki/NaN));
* методы, используемые для преобразования числа при выполнении математических операций;
* исключительные ситуации: [деление на ноль](https://ru.wikipedia.org/wiki/Деление_на_ноль), [переполнение](https://ru.wikipedia.org/wiki/Арифметическое_переполнение), [потеря значимости](https://ru.wikipedia.org/wiki/Антипереполнение), работа с [денормализованными числами](https://ru.wikipedia.org/wiki/Денормализованные_числа) и другие;
* операции: арифметические и другие.
* Стандарт 2008 года заменяет [IEEE 754-1985](https://ru.wikipedia.org/wiki/IEEE_754-1985). В новый стандарт включены двоичные форматы из предыдущего стандарта и три новых формата. В соответствии с действующим стандартом, реализация должна поддерживать по крайней мере один из основных форматов, как и формат арифметики и формат обмена.

## **Разработка стандарта**

Текущая версия IEEE 754—2008 была опубликована в 2008 году. Она дополняет и заменяет предыдущую версию IEEE 754-1985, созданную Dan Zuras и отредактированную [Майком Коулишоу](https://en.wikipedia.org/wiki/Mike_Cowlishaw)

Бинарные форматы в первоначальном стандарте включены в новый стандарт наряду с тремя новыми основными форматами (одним бинарным и двумя десятичными). Для того, чтобы соответствовать текущему стандарту, реализация должна реализовать по крайней мере один из основных форматов.

По состоянию на сентябрь 2015 года, стандарт пересматривается с целью включения уточнений.

## **Формат**

Формат IEEE 754 представляет собой «совокупность представлений числовых значений и символов». Формат может также включать в себя способ кодирования.

Формат включает:

* Числа, которые могут рассматриваться в двоичной или десятичной системе счисления. Вещественное число представляется тремя целыми числами, где S=знак (0 для положительного и 1 для отрицательного), С=[мантисса](https://ru.wikipedia.org/wiki/Мантисса_(математика)) (коэффициент), Q=[экспонента](https://ru.wikipedia.org/wiki/Экспонента). Для заданных целых чисел S, C, Q значением соответствующего вещественного числа является: (−1)S × C× BQ , где B является основанием (2 или 10). Например, число с основанием 10, битом знака 1 (число отрицательное), мантиссой 12345 и экспонентой −3 определяют число (−1)1 × 12345 × 10−3 = −12.345.
* Положительный нуль +0 и [отрицательный нуль −0](https://ru.wikipedia.org/wiki/−0_(программирование)).
* Две бесконечности: +∞ и −∞.
* Два вида [NaN](https://ru.wikipedia.org/wiki/NaN): тихий NaN (qNaN) и сигнализационный NaN (sNaN). NaN может нести полезную нагрузку, предназначенный для диагностической информации, указывающей источник, вызвавший NaN. Знак NaN не имеет никакого значения, но может быть предсказуемым в некоторых случаях.
* Возможные конечные значения, которые могут быть представлены в формате, определяются основанием b, количеством знаков в мантиссе (с точностью р) и максимальным значением emax:
* c должен быть целым числом в диапазоне от нуля до bp−1 (если b=10 и p=7 тогда с может быть от 0 до 9999999)
* q должно быть целым числом, чтобы 1−emax ≤ q+p−1 ≤ emax (если p=7 и emax=96 тогда q может быть от −101 до 90).
* Следовательно (для предыдущего примера) наименьшее отличное от нуля положительное число, которое может быть представлено 1 × 10−101 а самым большим 9999999 × 1090 (9,999999 × 1096), а также полный спектр чисел от −9,999999 × 1096 до 9,999999 × 1096. Числа -b1-emax и b1-emax (−1 × 10−95 и 1 × 10−95) являются самыми маленькими (по абсолютной величине) нормальными числами; ненулевые числа между этими наименьшими числами называются [субнормальные](https://ru.wikipedia.org/wiki/Субнормальный_ряд).

**Представление и кодирование в памяти**

Некоторые числа могут иметь несколько представлений в формате, в котором они были только что описаны. Например, если b = 10 и р = 7, то число −12,345 может быть представлено как: −12345 × 10−3, −123450 × 10−4 или −1234500 × 10−5.

Для десятичных форматов любое представление справедливо, и совокупность этих представлений называется когортами. Когда результат может иметь несколько представлений, стандарт определяет, какой выбран членом когорты.

Для бинарных форматов представление делается уникальным путём выбора наименьшего представляемого показателя. Для чисел с показателем в нормальном диапазоне (не все из них или все нули), ведущий бит мантиссы всегда будет равен 1. Следовательно, ведущий 1 бит может подразумеваться, а не сохраняться явно в памяти. Это правило называется ведущей битной конвенцией или скрытой битной конвенцией. Правило позволяет сберечь 1 бит памяти, чтобы иметь ещё один бит точности. Ведущий бит конвенции не используется для субнормальных чисел; их показатель находится за пределами нормального диапазона значений.

**Форматы обмена**

Форматы обмена предназначены для обмена данными с плавающей запятой с использованием битовой строки фиксированной длины.

Для обмена двоичными числами с плавающей запятой определены форматы обмена длиной 16 бит, 32 бита, 64 бита и любое кратное 32 битам ≥128. 16-разрядный формат предназначен для обмена или хранения небольших чисел (например, для графики или нейросетевых вычислений).

Схема кодирования этих двоичных форматов обмена такая же, как и для IEEE 754-1985: знаковый бит, за которым следуют индексы, которые описывают смещение экспоненты, и p-1 биты, которые описывают значение. Ширина поля экспоненты для k-битового формата вычисляется как w = round(4 log2(k))−13. Существующие 64- и 128-битные форматы следуют этому правилу, но 16- и 32-битные форматы имеют больше битов степени (5 и 8 битов соответственно), чем даёт эта формула (3 и 7 битов соответственно).

Как и в IEEE 754—1985, существует некоторая гибкость в кодировании NaN.

Для обмена десятичными числами с плавающей запятой определены форматы обмена для любого кратного 32 бита.

## **Предикат общего порядка**

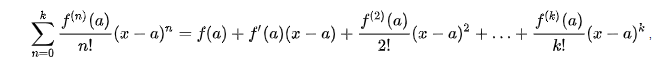
Стандарт предоставляет предикат totalOrder, который определяет общий порядок для всех чисел с плавающей точкой для каждого формата. Предикат согласуется с обычными операциями сравнения. Однако обычные операции сравнения обрабатывают NaN как неупорядоченные и сравнивают -0 и +0 как равные. Предикат totalOrder будет упорядочивать эти случаи, и также различать различные представления NaN для одного и того же числа с плавающей запятой, закодированное различными способами

# **Ряд Тейлора**

Ряд Тейлора — разложение [функции](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) в бесконечную сумму [степенных функций](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B5%D0%BF%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F).

Ряд Тейлора был известен задолго до публикаций Тейлора — его использовали ещё в XIV веке в Индии, а также в XVII веке [Грегори](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B5%D0%B3%D0%BE%D1%80%D0%B8,_%D0%94%D0%B6%D0%B5%D0%B9%D0%BC%D1%81) и [Ньютон](https://ru.wikipedia.org/wiki/%D0%9D%D1%8C%D1%8E%D1%82%D0%BE%D0%BD,_%D0%98%D1%81%D0%B0%D0%B0%D0%BA).

Ряды Тейлора применяются при аппроксимации ( или приближение — [научный](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%83%D0%BA%D0%B0) [метод](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4), состоящий в замене одних объектов другими, в каком-то смысле близкими к исходным, но более простыми.) функции многочленами. В частности, линеаризация уравнений происходит путём разложения в ряд Тейлора и отсечения всех членов выше первого порядка.

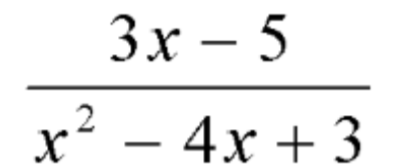


# **Описание программы**

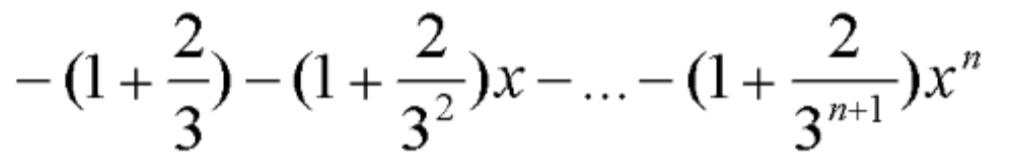
* Вычисление машинного эпсилон;
* Ввод коэффициента k (EPS \* k) и количество частей отрезка [ 0; 0.5 ];
* Выбор точки отрезка по формуле , где i - номер точки (i=0,…,n),

a- левая точка отрезка, b-правая точка отрезка.

* Вычисление по стандартной функции



* Вычисление по ряду Тейлора



Вычисляем следующий член суммы и добавляем его к сумме пока разница между предыдущей суммой и новой будет меньше дельты.

* Считаем погрешность между результатом вычислений по ряду Тейлора и стандартной функцией;
* Выводим строку таблицы:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Значение точки | Результат стандартной функции | Результат вычислений по ряду Тейлора | Количество итераций | Погрешность |

# **Заключение**

Так как погрешность небольшая, то вычисления в стандартной функции происходят аналогичным способом. Благодаря этому заданию, я научился вычислять значения не целых результатов функций и находить приближенное значение сложных функций.

# **Список Литературы**

* <https://en.wikipedia.org/wiki/IEEE_754-2008_revision>
* *Емелин Александр* Разложение функций в степенные ряды. Ряд Тейлора. Ряд Маклорена. Примеры решений
* *Ильин В. А., Садовничий В. А., Сендов Б. Х.* Математический анализ, ч. 1, изд. 3, ред. А. Н. Тихонов. М.: Проспект, 2004.
* *Камынин Л. И.* Математический анализ. Т. 1, 2. — 2001.