

# Лабораторная работа №2. Исследование особенностей чисел с плавающей точкой

Кузин Максим Витальевич

15 декабря 2025 г.

## Содержание

1 Двоичное представление unsigned int	1
2 Двоичное представление float	2
3 Ограничения мантиссы	2
4 Неожиданное поведение цикла	3
5 Численное интегрирование	4
6 Заключение	6

В данной работе исследуются способы представления числовых типов в языке программирования C++, а именно беззнаковых целых чисел и чисел с плавающей точкой. Особое внимание уделяется ограничениям формата хранения, возникающим ошибкам округления и нетривиальным эффектам, проявляющимся при вычислениях.

Целью лабораторной работы является анализ внутреннего устройства чисел с плавающей точкой, выявление границ их корректного применения и демонстрация ситуаций, в которых вычисления могут приводить к неожиданным результатам.

## 1 Двоичное представление unsigned int

Для начала рассмотрим, как в памяти хранится беззнаковое целое число. С помощью отладочного вывода отобразим его битовую структуру.

```
Enter number: 127
0000000000000000000000000000001111111
```

Полученный результат демонстрирует, что значение unsigned int представлено в виде фиксированного набора бит, напрямую соответствующих двоичному представлению числа.

## 2 Двоичное представление float

Аналогичным образом выведем на экран внутреннюю структуру числа с плавающей точкой типа float.

```
Enter float: 5.5  
01000000101100000000000000000000
```

Рассмотрим полученную запись подробнее. Первый бит равен нулю, что указывает на положительный знак числа. Далее следуют 8 бит экспоненты, имеющие значение 10000001. Оставшиеся биты образуют мантиссу: 011000000000000000000000.

В результате число представляется в нормализованном виде 101.1. При переводе в десятичную систему целая часть действительно равна 5, а дробная часть — 0.5, что подтверждает корректность интерпретации формата IEEE 754.

## 3 Ограничения мантиссы

Для исследования точности типа float был написан код, последовательно сохраняющий числа вида  $10^n$  при увеличении показателя степени  $n$ .

```
Curr: 1000000000.000000  
Bin: 010011100110111001101100101000  
  
Curr: 1000000000.000000  
Bin: 01010000001010100000101111001  
  
Curr: 99999997952.000000  
Bin: 0101000110111010010001110110111
```

Результаты выполнения программы показывают, что начиная с некоторого значения  $n$  (при мерно с  $n = 11$ ), сохранённое число перестаёт точно соответствовать степени десяти. Хотя

полученные значения остаются близкими к ожидаемым, погрешность увеличивается с ростом степени.

Это объясняется конечной разрядностью мантиссы и невозможностью точного представления больших десятичных чисел в формате float.

## 4 Неожиданное поведение цикла

Следующий эксперимент демонстрирует влияние ограниченной точности на работу циклов.

```
Curr: 16777216.000000
```

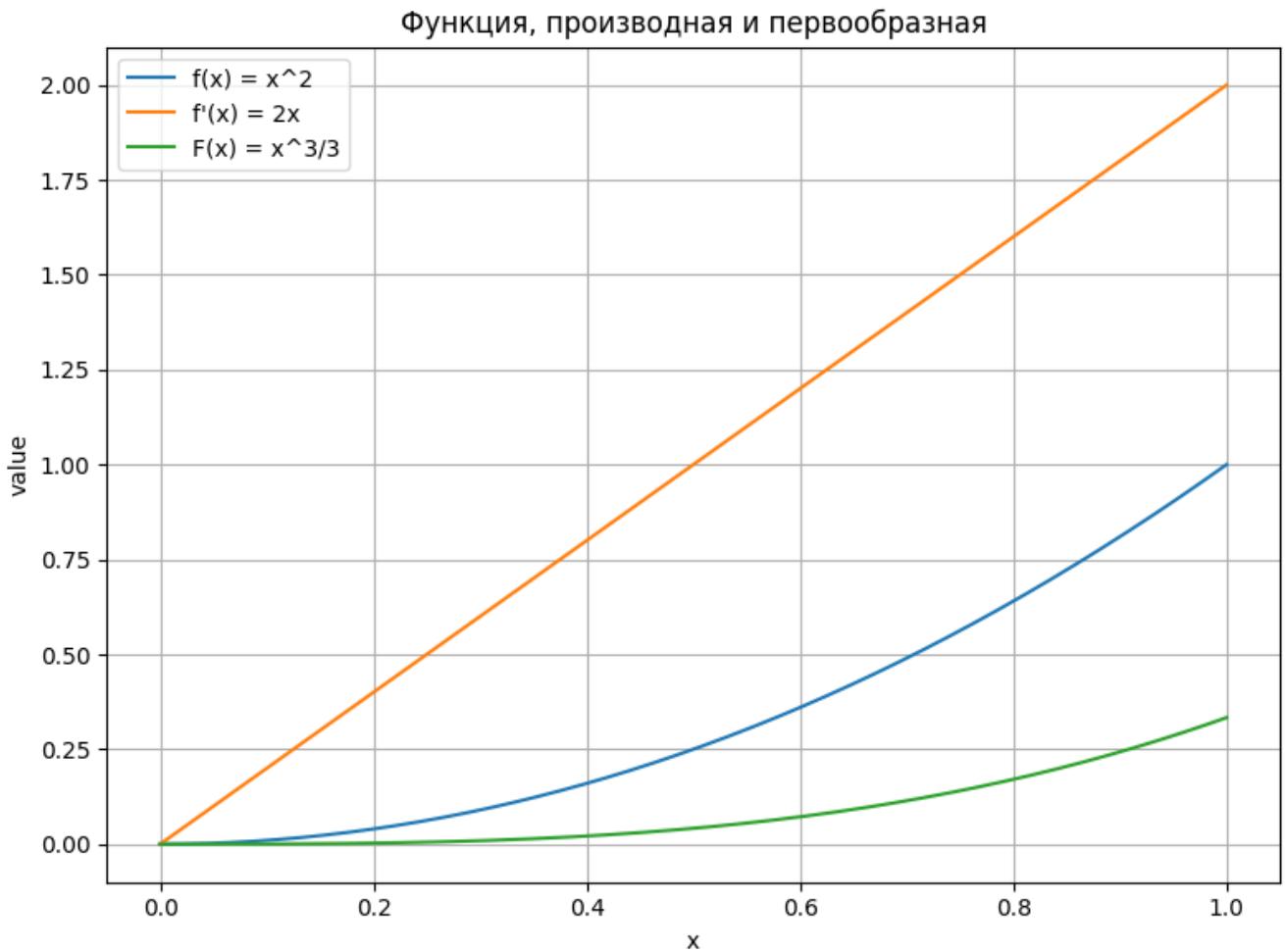
```
Int: 16777216
```

В ходе выполнения программы обнаруживается, что при достижении значения 16777216 переменная цикла типа float перестаёт увеличиваться. В результате условие выхода из цикла никогда не выполняется, и программа зацикливается.

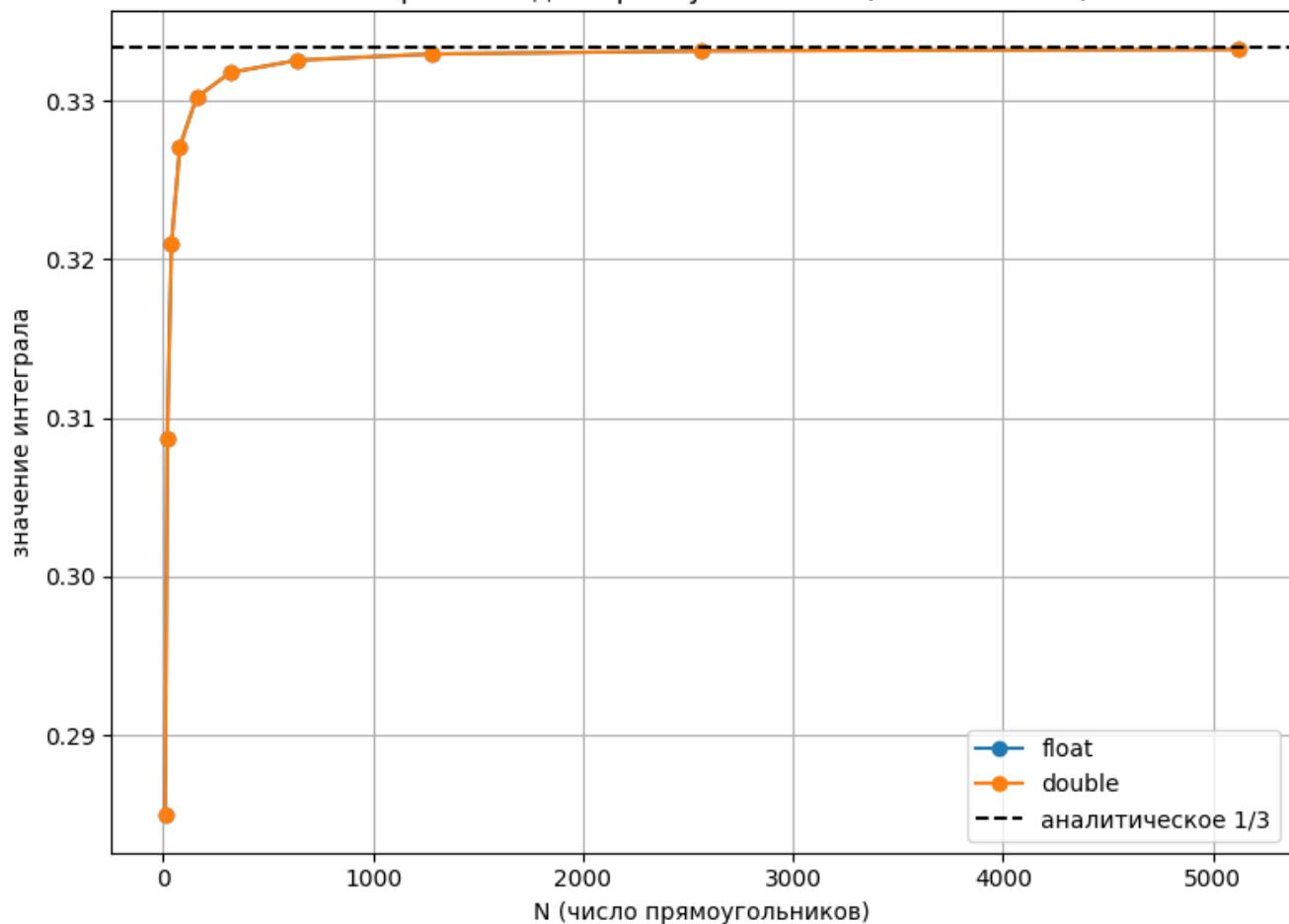
Данный эффект связан с тем, что начиная с этого значения, шаг изменения переменной становится меньше минимально различимого при данной точности числа.

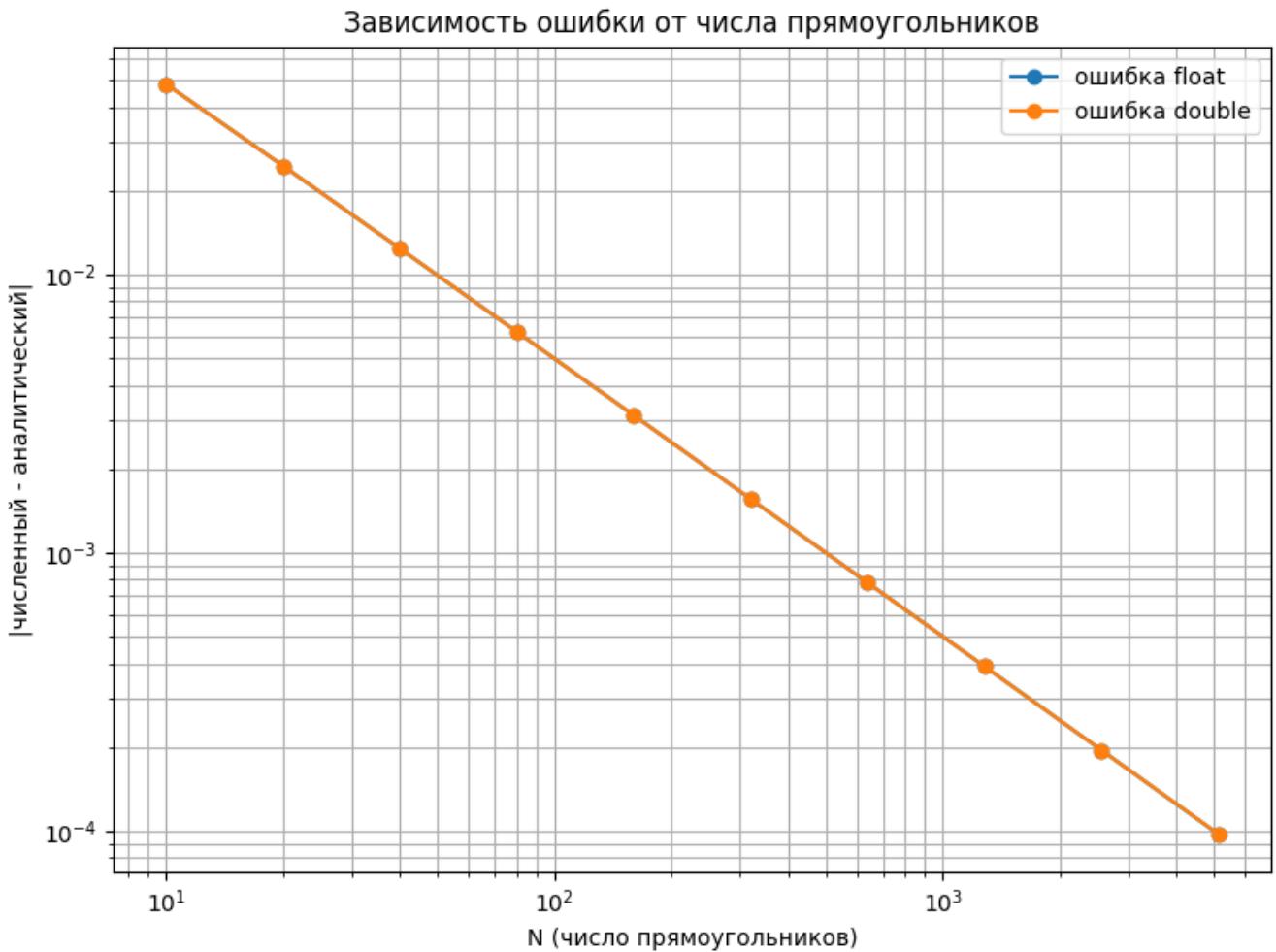
## 5 Численное интегрирование

В рамках работы также было проведено численное интегрирование функции с известным аналитическим результатом. Интеграл вычислялся методом прямоугольников при различном числе разбиений.



### Интеграл методом прямоугольников (float vs double)





Аналитическое значение интеграла равно:

$$\int_0^1 x^2 dx = 0.3333333333333333$$

Результаты численных экспериментов:

#### FLOAT:

- Оптимальное число разбиений:  $N = 5120$
- Полученное значение: 0.333235830068588
- Абсолютная ошибка:  $9.75 \cdot 10^{-5}$

#### DOUBLE:

- Оптимальное число разбиений:  $N = 5120$
- Полученное значение: 0.333235683441162
- Абсолютная ошибка:  $9.76 \cdot 10^{-5}$

## 6 Заключение

В ходе выполнения лабораторной работы были изучены особенности представления чисел с плавающей точкой в памяти компьютера. Были выявлены ограничения точности формата

`float`, продемонстрированы примеры накопления ошибок и показано, как они могут приводить к некорректному поведению программ.

Дополнительно были проведены эксперименты по численному интегрированию, наглядно иллюстрирующие влияние разрядности числового типа на точность вычислений.