Computer precision and type of numbers

On computers we have only finite precision (number of digit bits)

Def: Normalized floating point representation with respect to base b, stores a number x as $x = 0.a_i \cdots a_k\, b^n$ with:

$a_i \in \{0, \cdots, b-1\}$ are digits, k is called precision, n is called exponent, $a_i \cdots a_k$is called a mantissa, $a_1 \neq 0$. The fact that $a_1 \neq 0$ is called a normalization, it makes representation unique.

Ex.:

base $b = 10$: 32.213 $0.32213\ \cdot 10^2$

base $b = 2$: $x = \pm 0.b_1 b_2 \cdots b_k 2^n$, where $b_1 = 1$

The single precision, $4\ bytes = 32\ bits$

1 bit for the sign mantissa

1 bit ^{n} (exponent)

7 bits for the exponent integer (0-127)

23 bits for mantissa

$$2^{127} \approx 10^{38}$$

So, the range of representable numbers by single precision is:

$$10^{-38} < |x| < 10^{38}$$

Example

Since $2^{-24} \approx 10^{-7}$, we can represent 7 significant digits

We can obtain better representation via double precision (8 bytes)