1 Floating point arithmetic

Given that any computer has limited storage capacity, we can represent only finitely many numbers on a computer. Thus, inevitably, we have to deal with approximations of the real number system using finite computer representations. To arrive at a consistent representation of floating point numbers across different computer architecture, the most widely used and accepted standard is the Institute of Electrical and Electronics Engineers (IEEE-754) standard for representing real numbers. First note that all numbers are stored in binary (base 2) format. Any *normal* number on the machine is represented as

$$x = \pm 1.d_1 d_2 \dots d_s \times 2^e$$

where $1.d_1 d_2 \dots d_s$ is the significand and e is the exponent (both represented using 0's and 1's). For example, consider the number 77 in decimal. We have

$$77_{10} = 2^6 + 2^3 + 2^2 + 2^0 = 1001101_2 = 1.001101_2 \times 2^6 = 1.001101_2 \times 2^{110_2}$$

As inidicated in Figure 1, there is 1 sign bit, e bits for the exponent and s bits for the significand. We will now list out the general

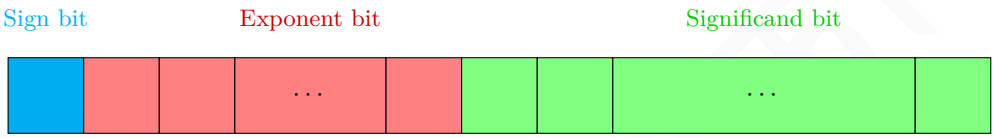

Figure 1: Bits to represent floating point numbers

conventions based on IEEE-754 standard.

- **Sign bit**: 0 indicates +, and 1 indicates −.
- **Exponent bit**: Since there are e bits for the exponent, there are a total of 2^e values the exponent can take. To represent negative exponents as well, a bias of $2^{e-1} - 1$ is introduced, i.e., 0 exponent is represented as $0\overbrace{111\dots1}^{e-1}{}_2$.
- **Significand bit**: Stores the leading bits in the mantissa apart from the leading 1.

Table 1: Floating point representations

Exponent \ Significand	All zeros	Non-zero
All zeros	± 0	Sub-normal numbers
All ones	$\pm\infty$	Not A Number
Else	Normal floating point numbers	

1.1 Normal floating point number

These are represented as

$$\pm 1.d_1 d_2 \dots d_s \times 2^E$$

where $2 - 2^{e-1} \leq E \leq 2^{e-1} - 1$ (after bias). Note that since normal floating numbers begin with 1, it suffices to store the s bits after the leading 1.

1.2 Sub-normal floating point number

These are represented as

$$\pm 0.d_1 d_2 \dots d_s \times 2^{2-2^{e-1}}$$

The exponent bits of sub-normal floating point numbers are all zero. The significand stores the s bits after the leading 0.

Table 2: Positive normal floating point number

	Mantissa	Exponent (without bias)	Number
Smallest	$d_i = 0$ for all $i \leq s$	$00\ldots01$	$2^{2-2^{e-1}}$
Largest	$d_i = 1$ for all $i \leq s$	$11\ldots10$	$\left(2-2^{-s}\right) \times 2^{2^{e-1}-1}$

Table 3: Positive sub-normal floating point number

	Representation	Number
Smallest	$d_i = 0$ for all $i < s$ and $d_s = 1$	$2^{2-s-2^{e-1}}$
Largest	$d_i = 1$ for all $i \leq s$	$\left(1-2^{-s}\right) \times 2^{2-2^{e-1}}$

1.3 Machine precision

This is defined as the difference between the smallest number exceeding 1 that can be represented on the machine and 1. Note that the smallest number exceeding 1 that can be represented on the machine is $1.00\ldots01 = 1 + 2^{-s}$.

$$\boxed{\text{Hence, machine precision is } \epsilon = 2^{-s}}$$

Note that if x is any real number and $\text{fl}(x)$ is the largest number not exceeding x representable on the machine (i.e., after appropriate chopping x will be represented as $\text{fl}(x)$ on the machine), we then have

$$\frac{x - \text{fl}(x)}{x} \leq \epsilon = 2^{-s} \text{ or equivalently } \text{fl}(x) = x(1-\delta)$$

where $0 \leq \delta < \epsilon$. From the above, observe that floating point representation introduces relative errors and not absolute errors.

- **Single precision** Of the total of 32 bits, 1 is alloted for sign, 8 for exponent and 23 for significand.
- **Double precision** Of the total of 64 bits, 1 is alloted for sign, 11 for exponent and 52 for significand.

Table 4: Floating point numbers on single and double precision

		32 bit machine	64 bit machine
Sub-normal	Smallest positive	$2^{-149} \approx 1.4 \times 10^{-45}$	$2^{-1074} \approx 4.94 \times 10^{-324}$
	Largest positive	$\left(1-2^{-23}\right) \times 2^{-126} \approx 1.18 \times 10^{-38}$	$\left(1-2^{-52}\right) \times 2^{-1022} \approx 2.23 \times 10^{-308}$
Normal	Smallest positive	$2^{-126} \approx 1.18 \times 10^{-38}$	$2^{-1022} \approx 2.23 \times 10^{-308}$
	Largest positive	$\left(2-2^{-23}\right) \times 2^{127} \approx 3.4 \times 10^{38}$	$\left(2-2^{-52}\right) \times 2^{1023} \approx 1.8 \times 10^{308}$
Machine precision		$2^{-23} \approx 1.2 \times 10^{-7}$	$2^{-52} \approx 2.2 \times 10^{-16}$