University of Idaho

Finite Precision Arithmetic

Number systems

$$x = \pm (d_n d_{n-1} \dots d_1 d_0 . d_{-1} d_{-2} \dots)_\beta =$$

$$= \pm d_n \cdot \beta^n + d_{n-1} \cdot \beta^{n-1} + \dots + d_1 \cdot \beta^1 + d_0 \cdot \beta^0 +$$

$$+ d_{-1} \cdot \beta^{-1} + d_{-2} \cdot \beta^{-2} + \dots , \qquad d_n \neq 0$$

β: base, d_i: digits, $0 \le d_i \le \beta - 1$

Ex $\beta = 10$: decimal system

$$(2015)_{10} = 2 \cdot 10^3 + 0 \cdot 10^2 + 1 \cdot 10^1 + 5 \cdot 10^0$$

University of Idaho

$\beta = 2$: binary system

$$(10101.01)_2 = 1\cdot 2^4 + 0\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 +$$
$$+ 0\cdot 2^{-1} + 1\cdot 2^{-2} = 16 + 4 + 1 + \frac{1}{4} = (21.25)_{10}$$

Q: how are real numbers represented in a computer?

Answer: floating point representation

$$x = \pm (0.d_1 d_2 \ldots d_n)_\beta \cdot \beta^e$$: x has n significant digits

$d_1 \neq 0$

$0.d_1 d_2 \ldots d_n$: mantissa

e : exponent $\quad -M \leq e \leq M$

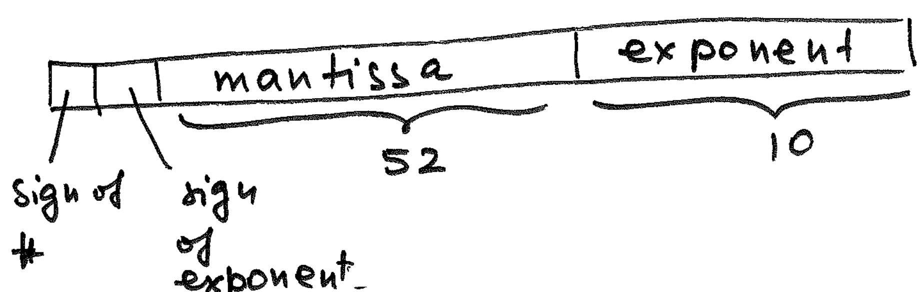
<u>Ex</u> Consider a computer with $\beta = 2$, $n = 4$, $M = 3$

$$x_{max} = (0.1111)_2 \cdot 2^3 = (111.1)_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} =$$
$$= (4 + 2 + 1 + 0.5)_{10} = (7.5)_{10}$$

$$x_{min} = (0.1000)_2 \cdot 2^{-3} = (0.0001)_2 = 1 \cdot 2^{-4} = (0.0625)_{10}$$

<u>Note</u>

1. In IEEE double precision arithmetic, each number is stored in memory as a string of 64 bits



The first two bits are used to store signs of a number and of exponent, 52 bits for mantissa, and remaining 10 for exponent.

Hence, we have $\beta = 2$, $n = 52$

$$M = (1111111111)_2 = 2^{10} - 1 = 1023$$

Aside

$$11_2 = 1 \cdot 2^1 + 1 \cdot 2^0 = 3 = 2^2 - 1$$

2. If x is a real number and $fl(x)$ is its floating point representation, then $x - fl(x)$ is called the <u>roundoff error</u>.

Number x can be chopped or rounded.

$$fl_{chop}(x) = \pm (0.d_1 d_2 \ldots d_n)_\beta \cdot \beta^e$$

or number x can be rounded

$$fl_{round}(x) = \begin{cases} \pm (0.d_1 d_2 \ldots d_n)_\beta \cdot \beta^e & \text{if } d_{n+1} < \frac{\beta}{2} \\ \pm \left[(0.d_1 d_2 \ldots d_n)_\beta + \beta^{-n} \right] \cdot \beta^e & \text{if } d_{n+1} \geq \frac{\beta}{2} \end{cases}$$

x : exact

$\tilde{x}$: approximation

$x - \tilde{x}$: <u>error</u>

$|x - \tilde{x}|$: <u>absolute error</u>

$\dfrac{|x-\tilde{x}|}{|x|}$: <u>relative error</u>

Note: it can be shown that when a number is rounded, the bounds on both absolute and relative errors due to roundoff are one-half the bounds when a number is chopped.

Ex $\pi = (3.14159265358979\ldots)_{10} = \underline{1}\cdot 2^1 + \underline{1}\cdot 2^0 + 0\cdot 2^{-1} +$

$+ 0\cdot 2^{-2} + 1\cdot \underset{=\frac{1}{8}}{2^{-3}} + 1\cdot \underset{=\frac{1}{64}}{2^{-6}} + 1\cdot \underbrace{2^{-12}}_{\frac{1}{4096}} + \ldots =$

$2^{-1} = \frac{1}{2} = 0.5$

$2^{-2} = \frac{1}{4} = 0.25$

$$= (11.001001000010\ldots)_2 =$$

$$= (0.\underbrace{1100}1001000010)_2 \cdot 2^2$$

with rounding

$$n = 4 \Rightarrow fl(\pi) = (0.1101)_2 \cdot 2^2 = 3.25 : \text{ closest 4-bit representation of } \pi$$

In reality, $n = 52$, the roundoff error in $fl(x)$ is $2^{-52} \approx 10^{-15}$