NUMERICAL ANALYSIS

2021

HARRI HAKULA

FLOATING-POINT NUMBERS

- Note: Many calculators use decimal system

Representation:

$$x = \pm (d_0 . d_1 d_2 \;\; \dots \; d_p \;)_k \cdot k^e$$

Parameters: integers

p : precision
k : base or radix
e : exponent $\longrightarrow$ $m \leq e \leq M$

Set $k = 2$ $\longrightarrow$ binary numbers

Normalisation: $d_0 \neq 0$, i.e., if $k = 2$
$\Rightarrow d_0 = 1$

<u>EXAMPLE</u> Toy floating-point system

$1.b_1 b_2$ exponents $-1, 0, 1$

$1.00_2 = 1$
$1.01_2 = 5/4$

Consider : $k = 10$

$$1.01 = 1 \cdot 10^0 + 0 \cdot 10^{-1} + 1 \cdot 10^{-2}$$

$\uparrow\uparrow$
1 2

Now : $k = 2$

$$1.01_2 = 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 1 + \frac{1}{4} = \frac{5}{4}$$

So,

$1.00_2 = 1$; exponents $-1, 0, 1$
$1.01_2 = 5/4$
$1.10_2 = 3/2$ $\quad 2^0 = 1, \; 2^{-1} = \frac{1}{2},$
$1.11_2 = 7/4$ $\quad 2^1 = 2$

The whole set :

1	5/4	3/2	7/4
2	5/2	3	7/2
1/2	5/8	3/4	7/8

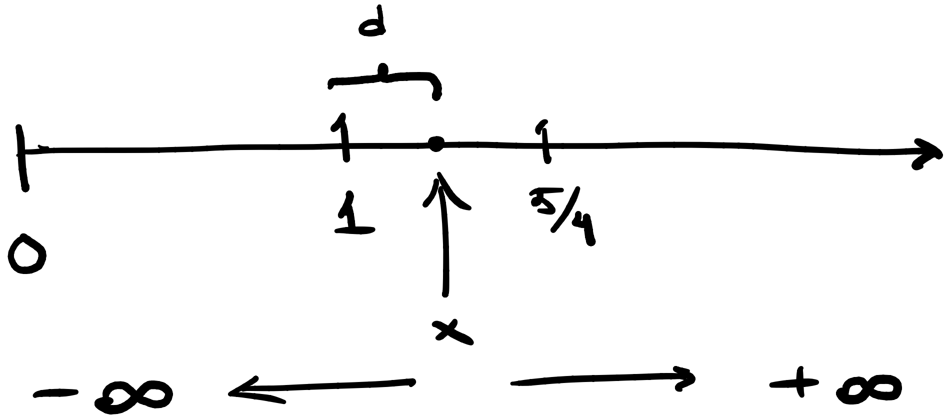
Important quantity :

Machine epsilon : $\frac{5}{4} - 1 = \frac{1}{4}$

Rounding

$x = RN(x) = \text{round}(x)$

RN rounding to nearest



Default: rounding to nearest

Here: $\text{round}(x) = 1$

$RU(x) = 5/4$ (rounding to $+\infty$)

It holds: $\text{round}(x) = x(1+\delta)$,

$|\delta| < \varepsilon/2$, where ε is the machine epsilon.

For instance:

$$a \oplus b = \text{round}(a+b) = (a+b)(1+\delta_1)$$
$$a \ominus b = \text{round}(a-b) = (a-b)(1+\delta_2)$$

$$\delta_1 \neq \delta_2$$

IEEE "Double Precision"

$k = 2$, 64 bits $\longrightarrow$ How are they used?

The sign : 1 bit
The exponent : 11 bits
The mantissa : 52 bits

The exponent field	Then number is	Type of number
$00\ldots0$	$\pm(0.b_1b_2\ldots b_{52})_2 \times 2^{-1022}$	0 or subnormal
$00\ldots01 = 1_{10}$	$\pm(1.b_1b_2\ldots b_{52}) \times 2^{-1022}$	
$00\ldots10 = 2_{10}$	$\pm(1.b_1b_2\ldots b_{52}) \times 2^{-1021}$	
$\ldots$		
$011\ldots11 = 1023_{10}$	$\pm(1.b_1b_2\ldots b_{52}) \times 2^{0}$	
$\ldots$		
$111\ldots10 = 2046_{10}$	$\pm(1.b_1\ldots b_{52}) \times 2^{1023}$	
$111\ldots11$	$\pm\infty$ if $b_1 = \ldots = b_{52} = 0$ otherwise NaN (not a number)	

Smallest positive normalised number :

$1.0_2 \times 2^{-1022} \approx 2.2 \times 10^{-308}$

Largest ... :

$1.1\ldots1 \times 2^{1023} \simeq 1.8 \times 10^{308}$