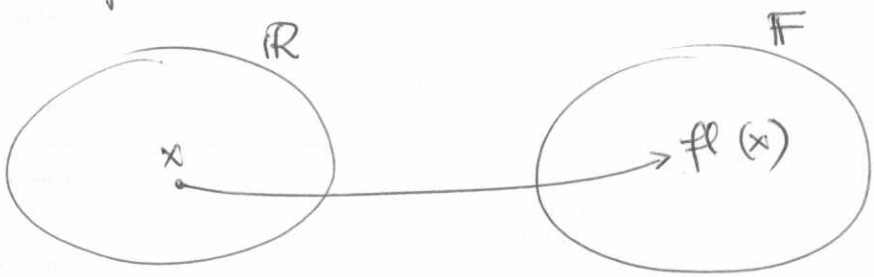
THE FLOATING POINT SYSTEM

$\mathbb{R}$ = set of real numbers $\quad \mathbb{R} = (-\infty, +\infty)$

$\mathbb{F}$ = set of machine numbers $\quad \mathbb{F} \subset \mathbb{R}$

$\mathbb{F}$ is a finite (i.e. its cardinality is finite) and bounded set

Not any $x \in \mathbb{R}$ can be exactly represented on the computer, thus we denote by $fl(x)$ the representative number of x in the set $\mathbb{F}$.



In order to describe $\mathbb{F}$ we start by the scientific (or exponential) form of a real number.

let us consider $x \in \mathbb{R}$ and we write it as

$$x = (-1)^s 0.a_1 a_2 \cdots a_t \cdots \beta^e = (-1)^s \underbrace{a_1 a_2 \cdots a_t \cdots}_{m} \beta^{e-t}$$

where $s = 0$ or 1

β = basis

e = exponent of the basis.

a_i are integer numbers : $0 \leq a_i \leq \beta - 1$, $i \geq 1$

with $a_1 \neq 0$

$a_1 a_2 \cdots a_t \cdots$ = m is said mantissa

We cannot store infinite digits $a_1 a_2 \cdots a_t \cdots a_\infty$ but only a finite number, say t

$$\Rightarrow fl(x) = (-1)^s a_1 a_2 \cdots \tilde{a}_t \beta^{e-t} = (-1)^s 0.a_1 a_2 \cdots \tilde{a}_t \beta^e$$

The last digit $\tilde{a}_t = \begin{cases} a_t & \text{if } a_{t+1} < \beta/2 \\ a_t + 1 & \text{if } a_{t+1} \geq \beta/2 \end{cases}$

ex1: $x = 123.45678 = (-1)^0 0.12345678 \cdot 10^3$

$t = 6$ $\qquad fl(x) = (-1)^0 0.123457 \cdot 10^3 \neq x$

ex2: $x = 123.45678$

$fl(x) = (-1)^0 0.12345678 \cdot 10^3 = x$

$t = 8$

Thus $fl(x)$ can be stored in a finite number of Bytes. Why?

Single precision format = 4 Bytes = 32 bits

$-126 \leq e \leq 127$

23 bits = 23 digits for $\beta = 2$

$\cong$ 6,9 digits for $\beta = 10$

$\left(2^{23} = 10^x \Rightarrow x = \log_{10} 2^{23}\right)$

Double precision format = 8 Bytes = 64 bits

$-1022 \leq e \leq 1023$

+1 digit if $\beta = 2$

53 bit = 53 digits in $\beta = 2$

$\sim$ 16 digits in $\beta = 10$

We denote by: t = n° of digits with respect to β

L, U = lower and upper bounds for e $(L<0<U)$

$\Rightarrow$ the set of numbers the machine can represent is denoted by $\mathbb{F} = \mathbb{F}(\beta, t, L, U)$:

$$\mathbb{F}(\beta,t,L,U) = \left\{ x = (-1)^s 0.a_1 \cdots a_t \cdot \beta^e, \text{ with } s=0,1 \right.$$
$$\left. 0 \le a_i \le \beta-1,\ a_1 \neq 0,\ L \le e \le U \right\} \cup \{0\}$$

Remark: zero cannot be represented by the floating point form, since it has all $a_i = 0$.

$\mathbb{F}$ is said <u>floating-point</u> system

"floating point" means that the point is not fixed, but it moves to match the form $x = (-1)^s 0.a_1 \cdots a_t \beta^e$

<u>Properties</u> of $\mathbb{F}$:

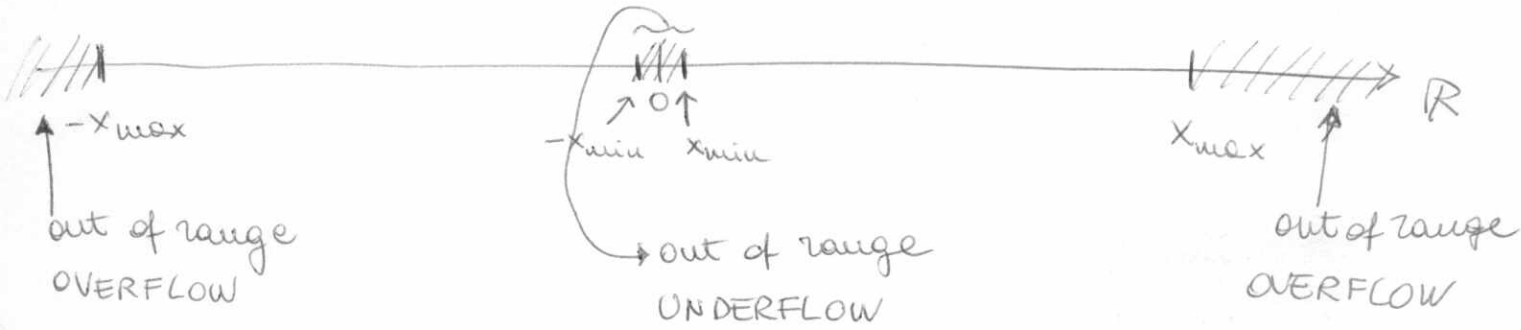
1. $\text{card}(\mathbb{F}) = 2(\beta-1)\beta^{t-1}(U-L+1)+1 < +\infty$
   ($\mathbb{F}$ is finite)
2. the smallest positive number is $x_{min} = \beta^{L-1}$
3. the largest positive number is $x_{max} = (1-\beta^{-t})\beta^U$
4. $\mathbb{F}$ is symmetric w.r.t. zero
5. floating point numbers are more dense close to 0 and less dense far from zero

Example $\mathbb{F}(2,3,-2,3)$
$\beta \; t \; L \; U$

$x_{min} = 0.100 \cdot \beta^{-2} = \left[1\cdot\beta^{-1} + 0\cdot\beta^{-2} + 0\cdot\beta^{-3}\right]\cdot\beta^{-2} = \beta^{-3} = 2^{-3} = \frac{1}{8}$

read these numbers following the positional notation

$x_{max} = 0.111 \cdot \beta^{+3} = \left[1\cdot\beta^{-1} + 1\cdot\beta^{-2} + 1\cdot\beta^{-3}\right]\beta^{3} = \left[\frac{1}{2}+\frac{1}{4}+\frac{1}{8}\right]\cdot 8 = 7$



$\mathbb{F}$ is bounded

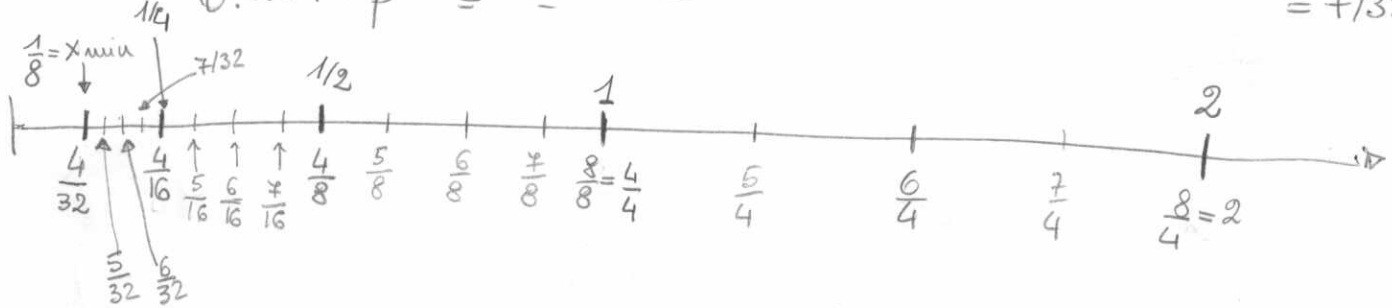
Now we compute the f.p. numbers between x_{min} and x_{max}

$x_{min} = 0.100 \cdot \beta^{-2} = \left[1\cdot\beta^{-1} + 0\cdot\beta^{-2} + 0\cdot\beta^{-3}\right]\beta^{-2} = \frac{1}{8} = \frac{4}{32}$

$0.101 \cdot \beta^{-2} = \left[1\cdot\beta^{-1} + 0\cdot\beta^{-2} + 1\cdot\beta^{-3}\right]\beta^{-2} = \left(\frac{1}{2}+\frac{1}{8}\right)\cdot\frac{1}{4} = \frac{5}{8}\cdot\frac{1}{4} = \frac{5}{32}$

$0.110 \cdot \beta^{-2} = \quad = 6/32$

$0.111 \cdot \beta^{-2} = \quad = 7/32$



We wrote all fp numbers with $e = -2$, now we

fix $e = -1$

$0.100 \cdot \beta^{-1} = 1/4 = 4/16$
$0.101 \cdot \beta^{-1} = 5/16$
$0.110 \cdot \beta^{-1} = 6/16$
$0.111 \cdot \beta^{-1} = 7/16$

then with $e = 0$, $e = 1$, ...

$e = 0$
$0.100 \cdot \beta^{0} = 1/2 = 4/8$
$0.101 \cdot \beta^{0} = 5/8$
$0.110 \cdot \beta^{0} = 6/8$
$0.111 \cdot \beta^{0} = 7/8$

$e=1$: $0.100 \cdot \beta^1 = (1) = 4/4$

$0.101 \cdot \beta^1 = 5/4$

$0.110 \cdot \beta^1 = 6/4$

$0.111 \cdot \beta^1 = 7/4$

. . . —

In Matlab $\beta = 2$, $t = 53\ (= 52+1)$

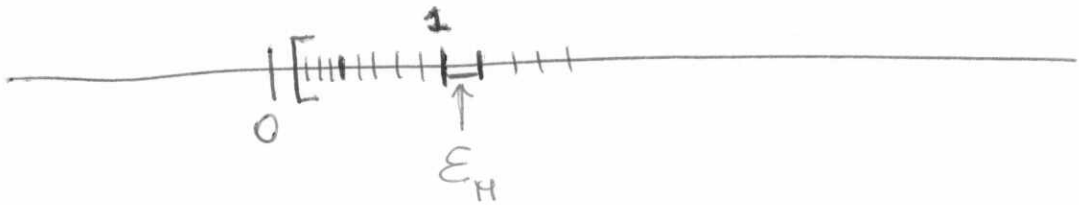
$U = -1021$, $L = 1024$

$x_{min} \simeq 2.2251 \cdot 10^{-308}$ >> realmin

$x_{max} \simeq 1.7977 \cdot 10^{+308}$ >> realmax

MACHINE PRECISION ε_M

It is the distance between 1 and the smallest floating point number greather than 1.



$$\varepsilon_M = 0.\underbrace{10\cdots01}_{t} \cdot \beta^1 - 0.\underbrace{100\cdots0}_{t} \cdot \beta^1 =$$

$$= 0.10\cdots01 \cdot \beta^1 - 0.10\cdots00 \cdot \beta^1 =$$

$$0.00\cdots\underset{\uparrow}{1}\cdot\beta^1 = \beta^{1-t}$$

t

$$\Rightarrow \boxed{\varepsilon_M = \beta^{1-t}}$$

Remark : ε_M is not the smallest positive f.p. but it is the distance between 1 and the smallest fp number greather than 1.

In $\mathbb{F}(2,3,-2,3)$ we have

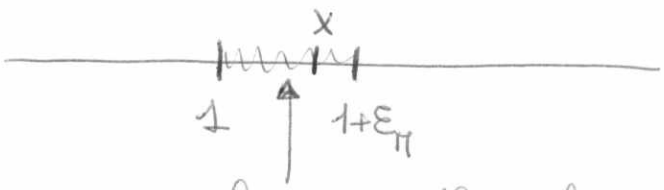
$$x_{min} = \frac{1}{8}$$

while $\varepsilon_M = \beta^{1-t} = 2^{1-3} = \frac{1}{4}$, In matlab $\varepsilon_M = 2^{1-53} = 2^{-52} \simeq 2.22 \cdot 10^{-16}$

Why is ε_M so important ?

Not every $x \in \mathbb{R} \cap [x_{min}, x_{max}]$ is a f.p. number

let us consider x between 1 and $1+\varepsilon_M$

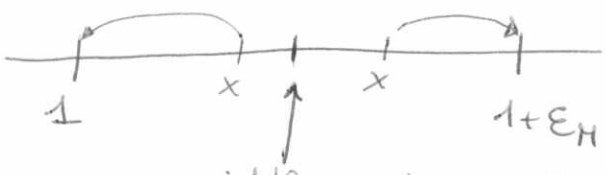


here no other f.p. number is present

This $x \notin \mathbb{F} \Rightarrow fl(x)$ is the floating point number closest to x and $fl(x) \neq x$.

$fl(x) = 1$ or $fl(x) = 1+\varepsilon_M$?

It depends if x is at the left or at the right of the middle point between 1 and $(1+\varepsilon_M)$



middle point $= \frac{1+(1+\varepsilon_M)}{2} = 1+\frac{\varepsilon_M}{2}$

if $x \in [1, 1+\frac{\varepsilon_M}{2}] \Rightarrow fl(x) = 1$

if $x \in (1+\frac{\varepsilon_M}{2}, 1+\varepsilon_M] \Rightarrow fl(x) = 1+\varepsilon_M$

We are interested in evaluating which error we generate ~~se~~ when we approximate x with $fl(x)$.

We have:

$$\boxed{\frac{|x - fl(x)|}{|x|} \leq \frac{1}{2}\varepsilon_M} \qquad \forall x \in [-x_{max}, -x_{min}] \cup [x_{min}, x_{max}]$$

$\frac{1}{2}\varepsilon_M$ is the semi amplitude of the interval $[1, 1+\varepsilon_M]$

and $\frac{|x - fl(x)|}{|x|}$ is the relative rounding error generated by the floating point system, or equivalently, by the machine arithmetics.

We measure relative (and not absolute) errors, since relative errors are more meaningfull and they are independent of the order of magnitude of the measure.

We set $\| u = \frac{1}{2}\varepsilon_M$ roundoff unit $\|$.

Obviously, if $x \in \mathbb{R} \cap \mathbb{F} \Rightarrow fl(x) = x$ and the error is null.

Remark $x = 0.1$ (in decimal expansion) $= 0.000\,11\,00\,11\,00\,11\,00\ldots$ ($\beta = 2$)
has infinite digits in binary expansion.

$\Rightarrow fl(0.1) \neq 0.1$ and $0 < \frac{|0.1 - fl(0.1)|}{|0.1|} \leq \frac{1}{2}\varepsilon_M = \frac{1}{2}\beta^{1-t}$

The larger "t" is and the smaller the rounding errors are.

PROPAGATION OF ROUNDING ERRORS

We evaluate how the errors on the data affects the result of a sum (a difference) and a product (division)

Given $x, y \in \mathbb{R}$, for simplicity we denote $fl(x)$ by $\bar{x}$ and $fl(y)$ by $\bar{y}$ $\quad (\bar{x} = fl(x),\ \bar{y} = fl(y))$

We want to compute $(x+y)$, the machine computes

$$fl(\bar{x}+\bar{y})$$

thus we want to measure the error in computing the sum:

$$e_+ = \frac{|fl(\bar{x}+\bar{y}) - (x+y)|}{|x+y|}$$

It holds: $e_+ = \dfrac{|fl(\bar{x}+\bar{y}) - (x+y)|}{|x+y|} \leq u(1+C_{xy}) + \underbrace{\mathcal{O}(u^2)}_{\sim u^2 << u}$

(u: roundoff unit)

where $C_{xy} = \dfrac{|x|+|y|}{|x+y|}$

Remark: if $\text{sign}(x) = \text{sign}(y) \Rightarrow C_{xy} = 1$

and the error e_+ is of the same order as u, the error on the result is comparable with the errors on the data.

if $\text{sign}(x) \neq \text{sign}(y)$ and $|x| \sim |y| \quad (x \sim -y)$

$\Rightarrow C_{xy} >> 1$ and $e_+ \leq u \cdot C_{xy}$

If, e.g. $C_{xy} \sim 10^8 \Rightarrow \quad \sim 10^{-16} \cdot 10^8 = 10^{-8}$

$$e_+ \leq 10^{-8}$$

which means that with one operation we loose 8 orders of magnitude on the accuracy of our results

Remark: Why did I write $fl(\bar{x}+\bar{y})$ if $\bar{x}, \bar{y} \in F$?

Because, even if $\bar{x}, \bar{y} \in F$, it is not guaranteed that $\bar{x}+\bar{y} \in F$, thus we have to approximate $(\bar{x}+\bar{y})$ by its float.

We want to compute $x \cdot y$, the machine computes $fl(\bar{x} \cdot \bar{y})$ and the error is

$$e_* = \frac{|fl(\bar{x} \cdot \bar{y}) - xy|}{|xy|} \leq 3u + O(u^2)$$

independently of x and y, thus when we multiply two fp numbers the error on the result is of the same order as the errors on the data.

We say that the sum is potentially UNSTABLE: even if the errors on data are small, we can have large errors on the solution.

The product is a STABLE operation: when the errors on the data are small, the error on the result is small, too.

Ex: $x = 0.1234 \quad y = -0.1231$

$x+y = 0.1234 - 0.1231 = 3 \cdot 10^{-4}$

$\beta = 10 \quad t = 3$

$\varepsilon_M = 10^{-2}$

$u = \frac{1}{2} \cdot 10^{-2}$

$\bar{x} = 0.123 \quad \bar{y} = -0.123$

$\Rightarrow \bar{x} + \bar{y} = 0 \qquad e_+ = \frac{|0 - 3 \cdot 10^{-4}|}{3 \cdot 10^{-4}} = 1 = 100\%$ di errore

$\rightarrow e_+ \leq C_{xy} \cdot u$

$C_{xy} = \frac{|x|+|y|}{|x+y|} = \frac{0.2465}{3 \cdot 10^{-4}} \simeq 821,\bar{6} \qquad u = \frac{1}{2}\beta^{1-t} = \frac{1}{2} \cdot 10^{-2}$; $C_{xy} u \simeq 4.108$

Associativity is violated whenever a situation of overflow or underflow occurs.

$a = 1.e+308$

$b = 1.1e+308$

$c = -1.001\,e+308$

$a + (b+c) = 1.0990\ e+308$

$(a+b)+c = Inf.$

Unicity of zero is violated

$1.e10 + 1.e-7 = 1.e10$

i.e. 1.e-7 is neutral in this addition

$x+y = x \quad \forall\, y < |x|u$ (u = roundoff unity)

Reference: Scientific computing (or Calcolo Scientifico) chapter 1.