Floating point arithmetic. (Demmel).

Note Title

The fact that infinitely many real nos. have to be stored in a finite amount of space gives rise to 2 limitations:
1) cannot represent arbitrary large or small numbers.
2) there will have to be gaps between the numbers.

So any real no. has to be rounded off to the closest representative — this introduces rounding error.

Several different representations have been proposed but by far the most widely used is the floating point repr.

The floating point number system $\mathbb{F}$ (IEEE standard) is the system that is accepted & used in all computing systems now.

$\mathbb{F} \subseteq \mathbb{R}$ determined by a base β & a precision p

$\mathbb{F} = \{0\} \cup \{\text{floating pt. numbers}\}$. ($\beta$ is an integer ≥ 2) (p is an integer ≥ 1)

Elements of $\mathbb{F}$ are called floating point numbers & are represented as follows:

any real number can be considered as

$$\pm \left[d_0 + d_1 \beta^{-1} + d_2 \beta^{-2} + \cdots + d_{p-1} \beta^{-(p-1)} \right] \times \beta^{e}$$

(exponent: e)

where each $0 \leq d_i < \beta$

& is stored as: $\pm \; d_0 \cdot d_1 d_2 \cdots d_{p-1} \times \beta^{e}$ → exponent.

($\pm$: sign; $d_0 \cdot d_1 d_2 \cdots d_{p-1}$: significand ($p$ digits))

eg: ① $\beta = 10$, $p = 3$: 0.1 is represented as 1.00×10^{-1} (normalized)

$0 \leq d_i \leq 9$.

0.10×10^{0}, 0.01×10^{1} (unnormalized)

In binary.

② $\beta = 2$, $p = 3$: 0.1 is represented as $0.000110011\ldots$

$$0.1 = \sum_{i=0}^{N} d_i \left(\frac{1}{2^i}\right) = 0\left(\frac{1}{2}\right) + 0\left(\frac{1}{4}\right) + 0\left(\frac{1}{8}\right) + 1\left(\frac{1}{16}\right) + 1\left(\frac{1}{32}\right) + \cdots$$

OR $0.1 = \frac{1}{10} = \frac{1}{1010_{(2)}}$, use long division →

Normalised repr. with $p = 3$: 1.10×2^{-4}.

Ex. Let $\beta = 2$, $p = 3$, $e_{min} = -1$, $e_{max} = 2$. (normalised).

$d_0 . d_1 d_2 \quad 2^{-1}$

$0 \le d_i < \beta \quad 2^{-2}$

List all floating point numbers for F with these parameters

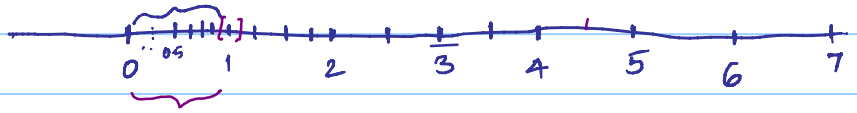
Floating pt. number	Corr. real number.
1.00×2^{-1}	$1.00 \times 2^{-1} = 1 \times \frac{1}{2} = 0.5$
1.01×2^{-1}	0.625
1.10×2^{-1}	0.75
1.11×2^{-1}	0.875
1.00×2^{0}	1
1.01×2^{0}	1.25
1.10×2^{0}	1.5
1.11×2^{0}	1.75
1.00×2^{1}	2
1.01×2^{1}	2.5
1.10×2^{1}	3
1.11×2^{1}	3.5
1.00×2^{2}	4
1.01×2^{2}	5
1.10×2^{2}	6
1.11×2^{2}	7

$0 \le d_i \le 1$

0, 1

Notice that the floating point numbers are not equally spaced.

The numbers represented are:



and their negatives.

Remark: In addition to the parameters β & p, the bounds on the exponent e_{min} & e_{max} are also defined for FF.

Convention: 0 is represented by $1.0 \times \beta^{e_{min}-1}$.

• Relative error and ulps.

The error in floating point computations is measured in 2 ways-

1) Units in last place (ulps).

If the floating point number $d.d\ldots d \times \beta^e$ is used to represent a real number z, it is said to be in error by $\left| d.d\ldots d - \frac{z}{\beta^e} \right| \times \beta^{p-1}$ ulps.

eg: (i) if 3.12×10^{-2} is used to represent $z = 0.0314$

$$\left| d.d\ldots d - \frac{z}{\beta^e} \right| \times \beta^{p-1} = |3.12 - 3.14| \times 10^2$$

$$= 0.02 \times 10^2 = 2 \text{ ulps.}$$

(ii) if $z = 0.0314159$ is represented by 3.14×10^{-2} then the error is -

$$\left| d.d\ldots d - \frac{z}{\beta^e} \right| \times \beta^{p-1} = |3.14 - 3.14159| \times 10^2$$

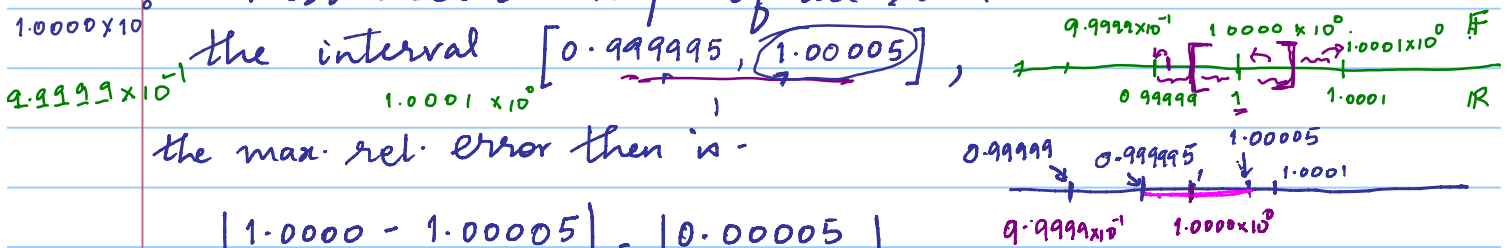
$$= 0.159 \text{ ulps.}$$

2) Relative error : If x is a real number & x' is its fl. pt. repr. then the relative error in representation is $\left|\frac{x - x'}{x}\right|$.

eg: $x = 3.14159$, $x' = 3.14 \times 10^0$, then the relative error is $\left|\frac{0.00159}{3.14159}\right| = 0.0005$.

Question : What is the maximum error that can occur in a floating point repr. with base β & precision p ?

If $\beta = 10$ & $p = 5$, then the number 1.0000×10^0 is the most accurate repr. of all real numbers in the interval $[0.999995, 1.00005]$,



the max. rel. error then is -

$$\left|\frac{1.0000 - 1.00005}{1.00005}\right| = \left|\frac{0.00005}{1.00005}\right|$$

$$= 0.5 \times 10^{-4}.$$

R.	F
1	1.0000×10^0
just before 1 : 0.99999	9.9999×10^{-1}
just after 1 : 1.0001	1.0001×10^0

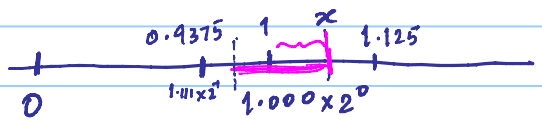
Note that the max. rel. repr. error occurs at 1.0000×10^0 because for $x > 1$, the denominators get larger, while for $0 < x < 1$, there are smaller gaps between the floating pt. numbers.

eg: $p=4$, $\beta=2$ consider

(say, $e_{min}=-1$)

$$\left.\begin{matrix}1.000\\1.001\\1.010\\\vdots\\1.111\end{matrix}\right\} \times 2^{-1} \quad \& \quad 1.000\times 2^{0}$$

$\mathbb{F}$ $\qquad \varepsilon = 0.5\times\beta^{1-p}$



The largest error of repr. occurs when x is repr. by 1.000×2^{0}.

$$1.111\times 2^{-1} = \left(1+1\times 2^{-1}+1\times 2^{-2}+1\times 2^{-3}\right)\times 2^{-1}$$

$$= \cdots = 0.9375.$$

$$1.001\times 2^{0} = \cdots = 1.125.$$

In general, the max. rel. repr. error for a fl. pt. number system with base β & precision p is $0.5\times\beta^{1-p}$.

This is exactly half the distance between 1 & the next (larger) floating point number $1+\beta^{1-p}$.

The IEEE standard stipulates that this error be taken as "machine epsilon",

i.e. $\varepsilon_{mach} := 0.5\times\beta^{1-p}$ i.e. $\frac{1}{2}\beta^{1-p}$.

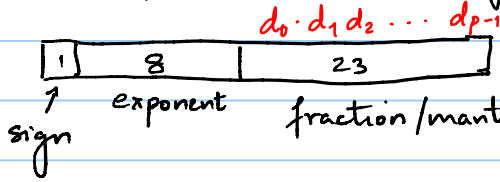
By defn., machine epsilon satisfies the foll. ppty:

$\forall x \in \mathbb{R} \quad \exists x' \in \mathbb{F}$ such that $\frac{|x - x'|}{|x|} \leq \varepsilon_{mach}$

i.e. $|x - x'| \leq \varepsilon_{mach} |x|$.

IEEE single precision: $\beta = 2$, $p = 23$, length = 32 bits, broken up as-

$0 \leq d_i \leq 1$



$e_{min} = -126$

$e_{max} = 127$

$\varepsilon_{mach} = 2^{-24} \approx 6 \times 10^{-8} = 0.00000006$

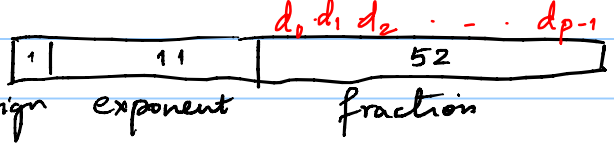
Range of positive normalized numbers is from

2^{-126} to 2^{127}

i.e. approx. 10^{-38} to 10^{38}.

(2^{-126} : underflow threshold
2^{128} : overflow threshold.)

IEEE double precision : $\beta = 2$, $p = 52$, 64 bits long



$e_{min} = -1022$

$e_{max} = 1023$.

$\varepsilon_{mach} = 2^{-53} \approx 10^{-16}$.

Range of normalized numbers is from

2^{-1022} to 2^{1023}

$\approx 10^{-308}$ to 10^{308}.

Fundamental properties of Floating pt. arithmetic.

Suppose $fl : \mathbb{R} \to \mathbb{F}$, $x \mapsto fl(x)$ denotes the floating pt. repr. of a real number x.

(i) Fund. ppty of fl pt. repr.

$\forall x \in \mathbb{R}$, $\exists\, \varepsilon$ with $|\varepsilon| \le \varepsilon_{mach}$ such that

(I) $$fl(x) = x(1+\varepsilon).$$

$\{ fl(x) = x + x\varepsilon$ i.e. $\left| \frac{fl(x) - x}{x} \right| = |\varepsilon| \le \varepsilon_{mach}.$

↑ rel. error of repr. $\}$

(ii) Let $x \circledast y$ denote the result (i.e. computed answer) of the arithmetic operation $x * y$. ($*$: addn / subt. / multi. / div.).

Fund. ppty of fl. pt. arithmetic -

$\forall x, y \in \mathbb{R}$, $\exists\, \varepsilon$ with $|\varepsilon| \le \varepsilon_{mach}$ such that

$$x \circledast y = (x * y)(1+\varepsilon).$$

(II) $$\left\{ \left| \frac{x \circledast y - x * y}{x * y} \right| = |\varepsilon| \le \varepsilon_{mach} \right\}.$$

↑ rel. error in fl. pt. arithmetic

Underflow, overflow; guard digits; NAN