NC A

Point I

The answer by a computer is approximate.

Point II

$$x = \pm \left(d_1 + \frac{d_2}{\beta^2} + \dots + \frac{d_N}{\beta^{N-1}} \right) \cdot \beta^e$$

$$= \pm \sum_{k=1}^{N} \frac{d_k}{\beta^{k-1}} \cdot \beta^e ;$$

Toy

$\beta = 2$

$N = 3$

$L = -1$

$U = 1$

Double precision

IEEE

$\beta = 2$

$N = 53$

$L = -1022$

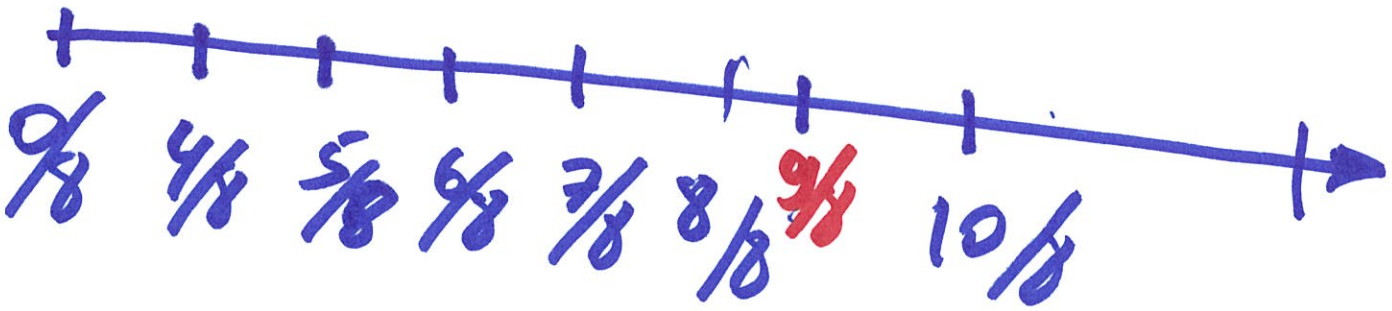
$U = 1023$

Toy System

4/8, 5/8 6/8 7/8

8/8 10/8 12/8 14/8

16/8 20/8 24/8 28/8



Finite # of #

- Almost every # can not be represented

x - number

Fl(x) - floating point presentation

$\varepsilon = \frac{1}{8}$

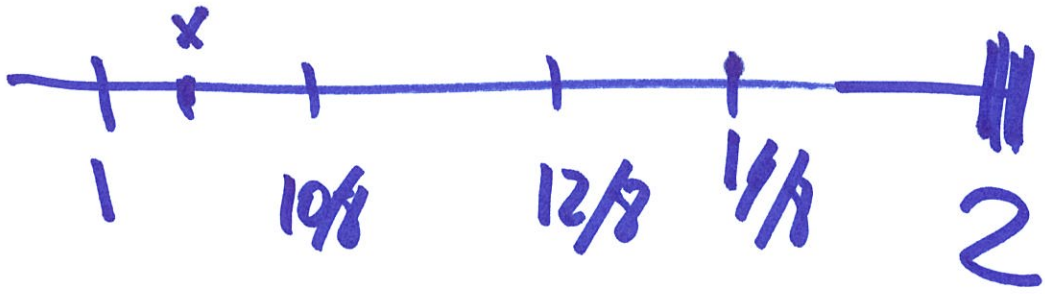
D

1.00 1/8

1.01 1/8

1.10 1/8

1.11 1/8



$1 \leq x < 2$

$|x - fl(x)| \leq \frac{1}{8}$

Round x to nearest

$fl(x)$

Chopping

$d_1 . d_2 d_3 d_4 \ldots d_N \rightarrow$

$d_1 . d_2 d_3 \ldots d_N$ ~~$d_{N+1} \ldots d_{N+2} \ldots$~~

Rounding to nearest

If there is a tie,
last digit is
even (zero = $\beta = 2$)

Chopping $|x - Fl(x)| < 2/8$

Rounding $|x - Fl(x)| < 1/8$?

1.00
1.01
] machine epsilon
epsilon machine
unit round off

~~ϵ_{mach}~~

1.01

$d_1 . d_2 d_3 \ldots d_N$

$\frac{1}{2}$ $\frac{1}{4}$ $\frac{d_N}{\beta^{N-1}}$

β^{1-N}

$\epsilon_{mach} = \beta^{1-N}$

$1 \leq x < 2$

chopping $|x - Fl(x)| \leq \epsilon_{mach} = \beta^{1-N}$

Rounding to nearest:

$|x - Fl(x)| \leq \frac{\epsilon_{mach}}{2} = \beta^{-N}$

Rounding to nearest $1 \le x < 2$

$$\left|\frac{x - fl(x)}{x}\right| < |x - fl(x)| < \frac{\epsilon_{mach}}{2}$$

$$\left|\frac{x - fl(x)}{x}\right| \le \frac{\epsilon_{mach}}{2}$$

Relative error in representing $1 \le x < 2$ is less than

$$\frac{\epsilon_{mach}}{2}$$

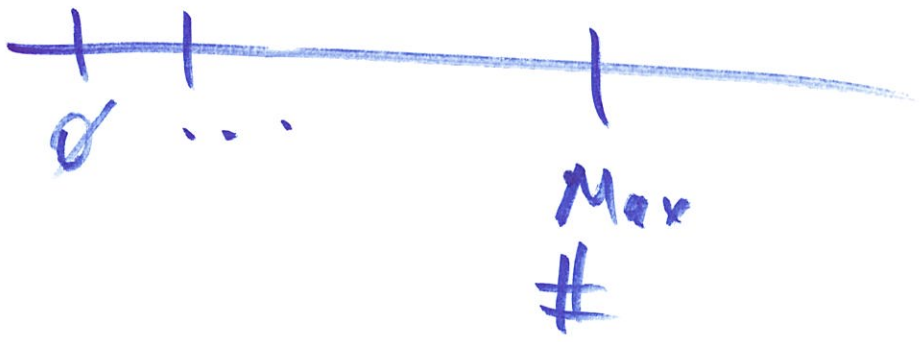
~~x = ma~~ $x \to m$ - mantissa

$$x = m \cdot \beta^{e}$$

Maximum relative error in representing "any" number is less than $\epsilon_{mach}/2 \equiv \beta^{-N}$

$$\left| \frac{x - fl(x)}{x} \right| \leq \frac{\epsilon_{mach}}{2} = \cancel{\beta^{-N}}$$

$$= \frac{\beta^{1-N}}{2}$$



Largest # representable in a system:

$1.11 \cdot 10^1$ - Toy

Double presion

1.111

$\cdot 10^{21}$

53 om's

Largest number $= \beta^{e+1}(1-\beta^{-n})$

Special numbers

$\emptyset$ - zero

Inf $\equiv$ Infinity

If computer obtains number > "largest #" The number is replaced by Inf.

overflow

$\frac{0}{0}$, Inf · 0, Inf/Inf

NaN - not a number

$$S_1 = \sum_{k=-1000}^{1000} k \cdot e^{k^2}$$ - overflow

k

$$= \left(\sum_{k=-1000}^{-1} + (k=0) + \sum_{k=1}^{1000} \right) k e^{k^2}$$

$k = -m$

$$\sum_{m=1}^{1000}$$

$$= \emptyset$$

$$S_2 = \sum_{k=1}^{1000} \frac{e^k + 2}{e^k + 3}$$

$$= \sum_{k=1}^{1000} \frac{(e^k + 2)e^{-k}}{(e^k + 3)e^{-k}} =$$

$$= \sum_{k=1}^{1000} \frac{1 + 2e^{-k}}{1 + 3e^{-k}}$$

Largest # is called OFL

realmax ≡ matlab
linderflow

Smallest # $= \beta^L$

smallest normalized #

$\beta = 2; L = -1022$

any number smaller than Smallest # is replaced by zero.

LINDERFLOW

Smallest normalized number

realmin = 2^{-53}

$= 2^{-1023}$

+ 0

$x = m_1 \cdot \beta^{e_1}$

$y = m_2 \cdot \beta^{e_2}$

$x + y = ?$

To sum, shift mantissas to match the exponents:

$\beta = 10;\ n = 2$

$1 + 10 = 1 + 1.0 \cdot 10^1 =$

$= 1.0 + 10 = 11$

$1.2 + 1.0 \cdot 10^1 =$

$= 1.2 + 10.0 = 11.2 \cong 11.0$

$1.0 \cdot 10^1 = 10.0 \cdot 10^0$

$1.2 \times 10^{-1} + 1.3 \cdot 10^1 =$

$= 0.012 \cdot 10^1 + 1.3 \cdot 10^1 =$

$= 1.312 \cdot 10^1 \approx 1.3 \cdot 10^1$

$x - \Delta x < x < x + \Delta x$

$y - \Delta y < y < y + \Delta y$

$(x+y) - (\Delta x + \Delta y) < (x+y) < (x+y) + (\Delta x + \Delta y)$

$x + y - (\Delta x + \Delta y) <$

Multiplications

x with d digits
y with d digits

$x \cdot y$ may $2d$ digits

Divide $\frac{x}{y}$ may have ∞ many digits

Q

$x = 1.3$

$y = 1.2$ > 2 significant digits

$x - y = 0.1$ ← one digit

$$x = \left(1 + \frac{\epsilon_{mach}}{2}\right)$$

$$y = \left(1 - \frac{\epsilon_{mach}}{2}\right)$$

$$x - y = \left(1 + \frac{\epsilon_{mach}}{2}\right) - \left(1 - \frac{\epsilon_{mach}}{2}\right) =$$

$$= \epsilon_{mach} \Rightarrow$$ computer will give 0

Submormals:

IF $E = L$ then

leading digit may be $\not{0}$.

"Submormals" ? 1986

"Toy" system

$0.01 \cdot 10^{-1}$

$0.10 \cdot 10^{-1}$

$0.11 \cdot 10^{-1}$

gradual underflow

$$X = 3(4/3 - 1) - 1$$

S

$\beta = 10, N = 2$

$\emptyset$, NaN, Inf

$4/3 = 1.3$

$4/3 - 1 = 0.3$

$3(4/3 - 1) = 0.9$

$3(4/3 - 1) - 1 = -0.1$

$$\left| \frac{X - Fl(x)}{X} \right| < \frac{\epsilon_{mach}}{2}$$

$$f(x) = \frac{1 - (1-x)}{x}$$

$x > 0$, $1 - x < 1$,

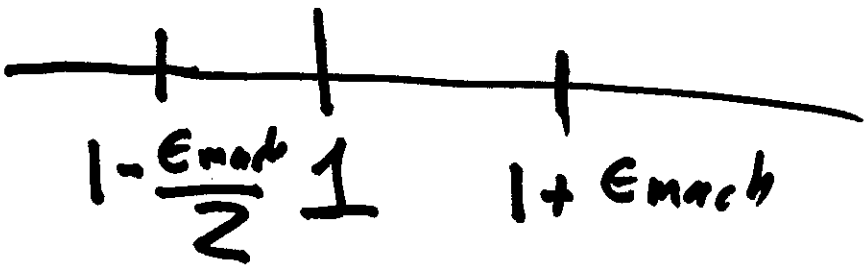
$\frac{\epsilon_{mach}}{2}$ apart

$x < 0$, $1 - x > 1$

ϵ_{mach} apart

$$f(x) = \frac{1-(1-x)}{x}$$

$$x = \frac{\epsilon_{mach}}{4}$$



$$\left(1 - \frac{\epsilon_{mach}}{4}\right) \approx 1 - \frac{\epsilon_{mach}}{2}$$

$$1-(1-x) = 1-\left(1-\frac{\epsilon_{mach}}{2}\right)$$

$$= \frac{\epsilon_{mach}}{2}$$

$$\frac{1-(1-x)}{x} \approx 2$$

$\epsilon_{mach} = 2.2E-16$

$\epsilon_{ps} = 2.2E-16$

$X = -\epsilon_{ps}/2 - \underset{\text{left}}{\downarrow}$

$1-(1-X) = 1-(1+\frac{\epsilon_{ps}}{2}+\text{left}) = -\epsilon_{ps}$; $\frac{1-(1-x)}{x} = 2$

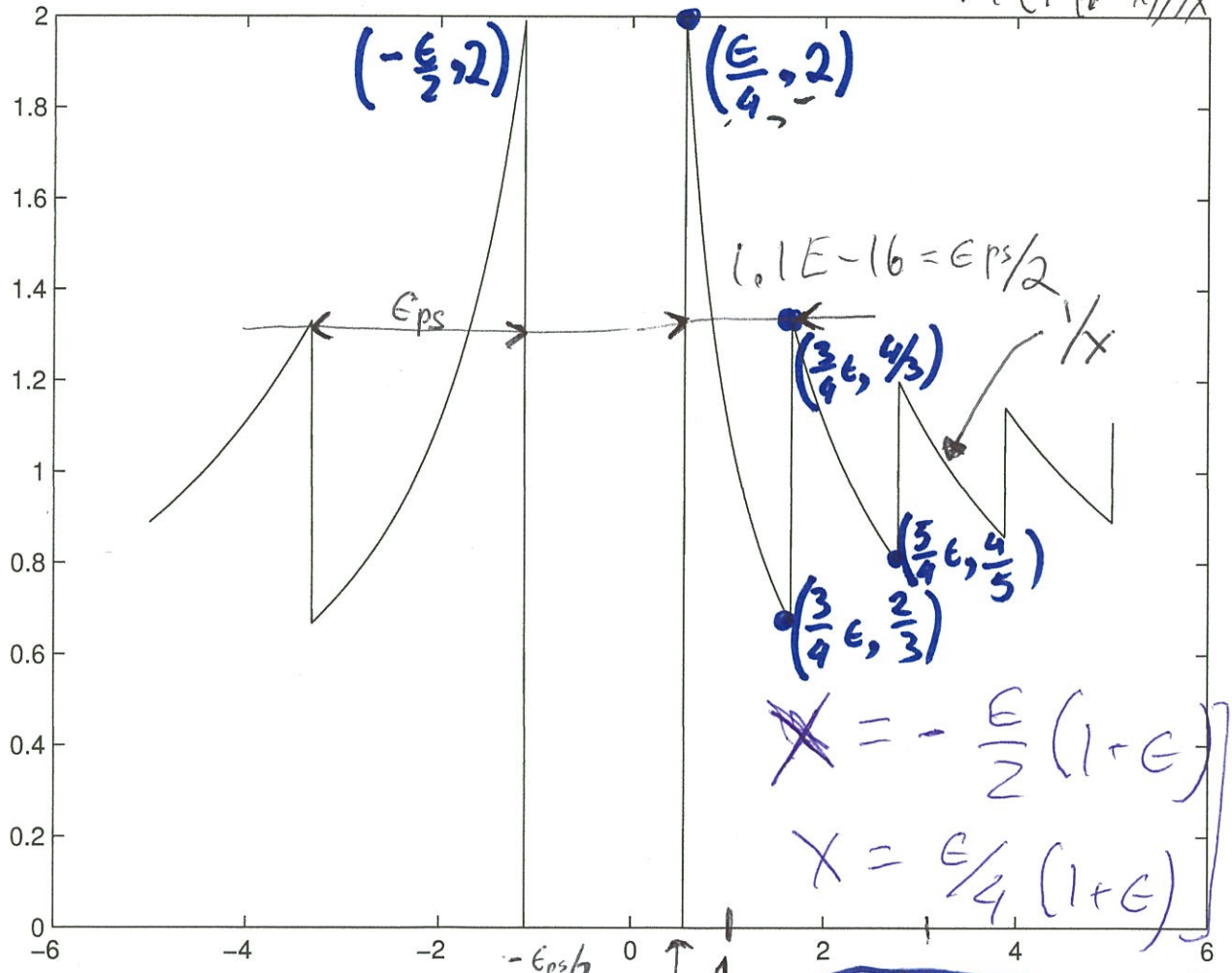
$\frac{1-(1-x)}{x}$

$X = -\frac{eps}{2} - 10^{-30}$

$(1-(1-x))/x = 1.99981$

$X = -\frac{\epsilon}{2}(1+\epsilon)$

$(1-(1-x))/x = 2$



$X = -\frac{\epsilon}{2}(1+\epsilon)$ $f(x) = 2$

$X = \epsilon/4\,(1+\epsilon)$ $f(x) = 2$

$X = \left(\frac{\epsilon}{4} + \frac{\epsilon}{2}\right)$; $f(x) = \frac{1-(1-\epsilon)}{3/4\,\epsilon} = \frac{4}{3}$

$0.55E(-16) = \epsilon_{ps}/4$; $X = (\frac{\epsilon}{4} + \frac{\epsilon}{2})(1-\epsilon)$

$f(x) = 2/3$

$1-\epsilon \quad 1-\frac{\epsilon}{2} \quad 1 \quad 1+\epsilon \quad 1+2\epsilon$

$X = \epsilon_{ps}/4$

$1 - X = 1 - \frac{\epsilon_{ps}}{4} = 1.000\ldots$ (n) $\quad 1 = 1 - \frac{\epsilon_{ps}}{2}$

$0.000\ldots 1111 \cdot 10^{-1}$

$\frac{1-(1-x)}{x} = 2$

2 digits, $X = 0.05$

$\frac{1-(1-0.05)}{1-(1-0.1)} = ?$

Matlab command used:

```
< M A T L A B (R) >
                Copyright 1984-2023 The MathWorks, Inc.
                R2023b (23.2.0.2365128) 64-bit (glnxa64)
                        August 23, 2023


To get started, type doc.
For product information, visit www.mathworks.com.

>> Inf/Inf
Inf/Inf

ans =

   NaN

>> -Inf+Inf
-Inf+Inf

ans =

   NaN

>> k = [ 1:1000];
k = [ 1:1000];

>> y = (exp(k)+2)./(exp(k)+3);
y = (exp(k)+2)./(exp(k)+3);
>> S1 = sum(y)
S1 = sum(y)

S1 =

   NaN

>> Inf+Inf
Inf+Inf

ans =

   Inf

>> Inf/Inf
Inf/Inf

ans =
```

```
   NaN

>> y = (2*exp(-k)+1)./(3*exp(-k)+1);
y = (2*exp(-k)+1)./(3*exp(-k)+1);
>> sum(y)
sum(y)

ans =

  999.6577

>> Maximum = 2^(1023+1)*( 1-2^(-53))
Maximu

>> m = 2^(1023+1)*( 1-2^(-53))

Maximum =

   Inf

>> Maximum = 2^(1023)*( 1-2^(-53))*2
m = 2^(1023+1)*( 1-2^(-53))

m =

   Inf

 >> Maximum = 2^(1023)*( 1-2^(-53))*2
Maximum = 2^(1023)*( 1-2^(-53))*2

Maximum =

  1.7977e+308


  >> realmax
realmax

ans =

  1.7977e+308

>>

>> 2^(-1022)
2^(-1022)

ans =
```

```
  2.2251e-308

>> realmin
realmin

ans =

  2.2251e-308

>> 2 * realmax
2 * realmax

ans =

   Inf

>> realmin/2
realmin/2

ans =

  1.1125e-308

>> 2^(-52)
2^(-52)

ans =

   2.2204e-16

>> eps
eps

ans =

   2.2204e-16

f(3)

y =

     1

>>

>>


>> f(eps*223/888)
```

```
y =

    1.9910

    >> f(-eps*449/888)

y =

    1.9777

>> f(eps*667/888)

y =

    1.3313

>> f(eps*664/888)

y =

    0.6687


ans =

    0.6687

    >> f(eps*667/888)

y =

    1.3313


    >> f(1.25*eps)

y =

    0.8000

 f(1.25*eps)

y =

    0.8000

f(-1.49*eps)
```

```
y =

    0.6711


ans =

    0.6711

>> f(-1.51*eps)

y =

    1.3245


ans =

    1.3245
```

==========

Here is how the file f.m looks like:

```
>> !more f.m
function y=f(x)
  y = (1-(1-x))./x
  end
```