

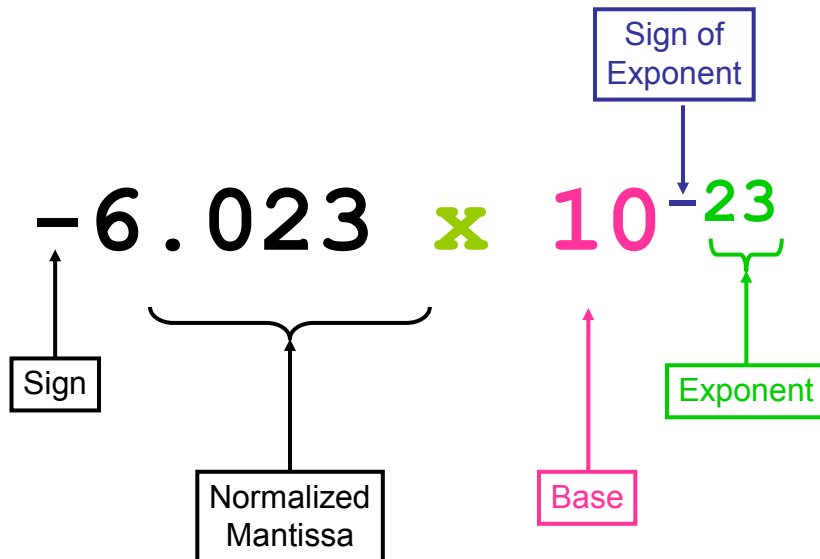
***Computational Methods***  
CMSC/AMSC/MAPL 460  
Representing numbers in floating point

Ramani Duraiswami,  
Dept. of Computer Science

## Floating point

- Attempt to
  - Handle decimal numbers
  - increase the range of numbers that can be represented
  - Provide a standard by which exceptions are consistently handled
- Use Scientific Notation as a guide
- Represent overflow (infinity), not-a-number, zero, underflow
- Handle things in small amount of space
  - Representing numbers in 32 bits
  - Representing numbers in 64 bits

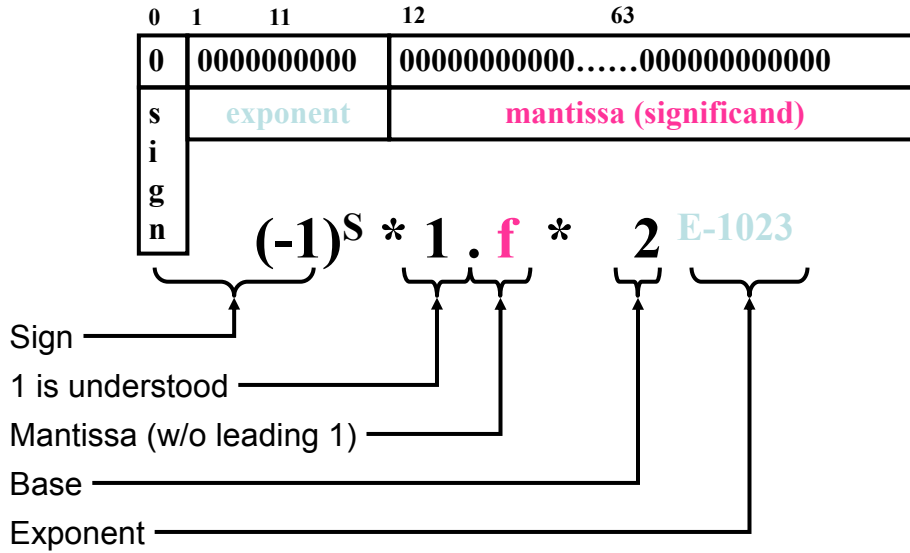
## Scientific Notation



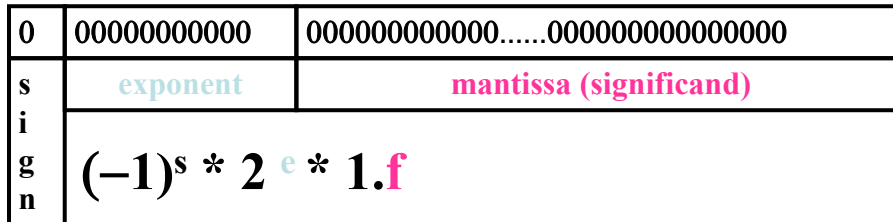
## Floating point on a computer

- Store each number as two numbers and two signs
  - Mantissa and exponent
- Mantissa is “normalized”
- If we have infinite spaces to store these numbers, we can represent arbitrarily large numbers
- With a fixed number of spaces for the two numbers (mantissa and exponent) the number representation is more limited

# IEEE-754 (double precision)

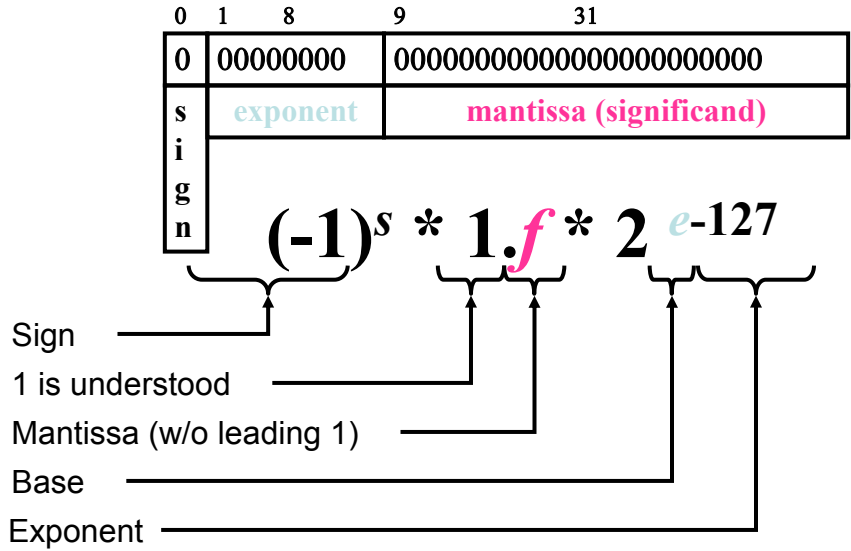


Can be written...

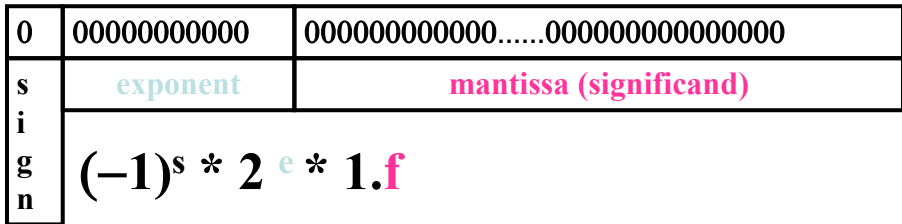


	e+1023 == 0	0 < e+1023 < 2047	e+1023 == 2047
f==0	0	Powers of Two	∞
f~0	Non-normalized typically underflow	Floating point Numbers	Not A Number

# IEEE-754 (single precision)



Can be written...



	e+127 == 0	0 < e+127 < 255	E+127 == 255
f==0	0	Powers of Two	∞
f~0	Non-normalized typically underflow	Floating point Numbers	Not A Number

## IEEE - 754

- Numbers are represented as  $x = \pm(1 + f) \cdot 2^e$   $0 \leq f < 1$
- The number  $f$  satisfies  $0 \leq 2^{52} f < 2^{52}$
- In single precision it satisfies  $0 \leq 2^{23} f < 2^{23} f$
- The number  $e$  satisfies  $-1022 \leq e \leq 1023$
- In single precision it satisfies  $-126 \leq e \leq 127$
- Finiteness of  $f$ : limitation in precision
- Finiteness of  $e$ : limitation in range
- Floating point numbers are not uniform
- They have a discrete maximum and minimum

## Some numbers cannot be exactly represented

- Would imagine decimal numbers could be easily represented
- But in fact 0.1 cannot be exactly represented

$$\frac{1}{10} = \frac{1}{2^4} + \frac{1}{2^5} + \frac{0}{2^6} + \frac{0}{2^7} + \frac{1}{2^8} + \frac{1}{2^9} + \frac{0}{2^{10}} + \frac{0}{2^{11}} + \frac{1}{2^{12}} + \dots$$

$$t = \left(1 + \frac{9}{16} + \frac{9}{16^2} + \frac{9}{16^3} + \dots + \frac{9}{16^{12}} + \frac{10}{16^{13}}\right) \cdot 2^{-4}$$

- $e=-4$   $f=(0.1001)$  repeating

## Special numbers

- *eps is the distance from 1 to the next larger floating-point number.*
- $\text{eps} = 2^{-52}$
- In Matlab

	Binary	Decimal
eps	$2^{-52}$	2.2204e-16
realmin	$2^{-1022}$	2.2251e-308
realmax	$(2-\text{eps}) \cdot 2^{1023}$	1.7977e+308

## Rounding vs. Chopping

- **Chopping:** Store  $x$  as  $c$ , where  $|c| < |x|$  and no machine number lies between  $c$  and  $x$ .
- **Rounding:** Store  $x$  as  $r$ , where  $r$  is the machine number closest to  $x$ .
- **IEEE standard arithmetic uses rounding.**

## Machine Epsilon

- **Machine epsilon** is defined to be the smallest positive number which, when added to 1, gives a number different from 1.
  - Alternate definition (1/2 this number)
- **Note:** Machine epsilon depends on  $d$  and on whether rounding or chopping is done, but does not depend on  $m$  or  $M$ !

## Examples from the book

- floatgui
  - In the software that is available for download from the book site



- Consequences of the representation
- How many counts do the following loops execute for?
 

```
x = 1; while 1+x > 1, x = x/2, pause(.02), end
```

```
x = 1; while x+x > x, x = 2*x, pause(.02), end
```

```
x = 1; while x+x > x, x = x/2, pause(.02), end
```