MCEN 3030

25 Jan 2024

HW #1 Released Tom/Sat

* See formatting guidelines

Due Sunday 11:59PM

Last time:
- Nested functions
- Error

Today:
- Number Storage + operations
- Round-off / Truncation Error

Q: How does a computer calculate something like $\sin(4.5)$?

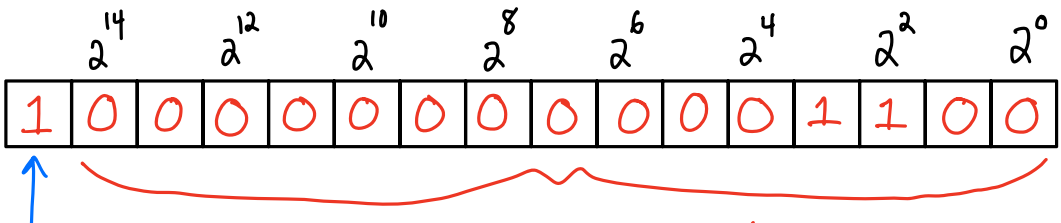
Computer Representation of Numbers

An important issue in many/most languages:
Declaring the number "type"

MATLAB is sloppy

- "Boolean": True/False → 1 or 0
  stored in one "bit"
- "Integer": $-3, -1, 0, 99, 1015799$

Typical: signed magnitude



this is 16 bits

sign
0: +
1: −

Binary Representation of a number

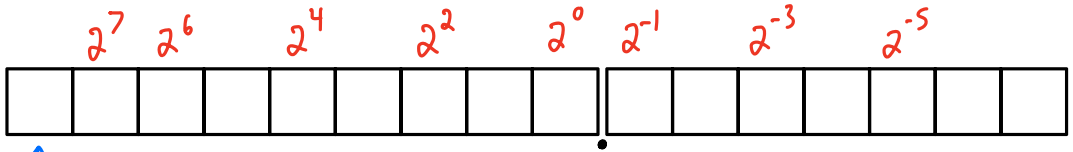
Base 10: $12 = 1 \times 10^1 + 2 \times 10^0$

Base 2: $1100 =$

$= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$

$= 8 + 4 + 0 + 0$

- "Fixed Point" representation of a non-integer

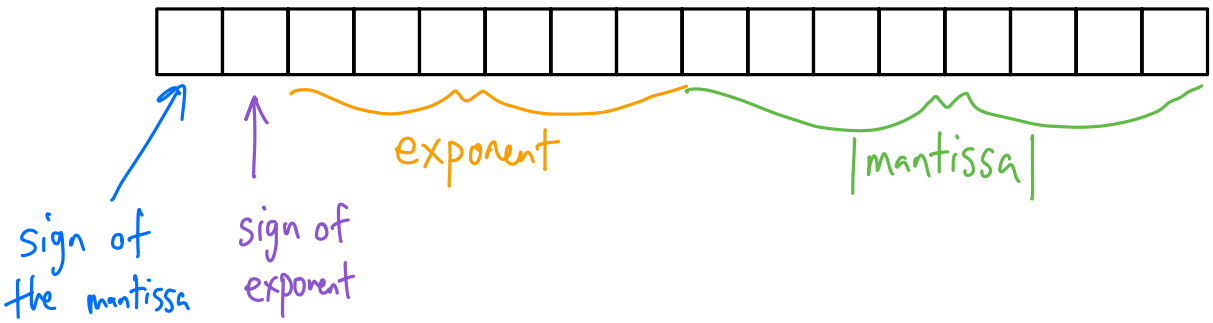


sign
0: +
1: −

  - "knows" where the decimal point is based on the declared data type
  - Obvious use (in base 10): money
  - If a certain application has a typical number size, this storage strategy can be beneficial
  - But generally: trouble if you want to store $\sim 10^{25}$ and 10^{-25}.

- "Float": Represent with $m \cdot b^n$ (n: exponent; m: mantissa; b: base = 2)

$\frac{1}{b} \leq m < 1$
to avoid leading 0



This is what MATLAB uses — actually a "double float" or just "double": 32 bits

Arithmetic with floats

$b = 10$

Let's use smaller (storage size) numbers and base 10 and demonstrate by example

$$153.1 + 1.471 = 0.1531 \times 10^3$$
$$+ \; 0.001471 \times 10^3$$
$$0.154571 \times 10^3$$

gets lost

$$\Longrightarrow \quad 0.1545 \times 10^3 = 1545$$

$\rightarrow$ We have "round-off" error because the numbers here have storage size "4"

So, back to the question: Calculate $\sin(4.5)$

Done via Taylor Series:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

$$= \sum_{n=1}^{\infty} \frac{(-1)^{n-1} x^{2n-1}}{(2n-1)!}$$

- forgot the $(-1)^{n-1}$ in lecture
- Other ways to write this generalization too

Involves operations:

- Factorial → multiplication
- Raising to powers → multiplication
- Division → multiplication w/ a negative power
- Addition/subtraction

Obviously we can't take infinitely many terms (would take infinite time to stop)

→ Truncation error

$$\epsilon_{relative} \equiv \frac{(\text{current value}) - (\text{last value})}{\text{current value}}$$

→ Stop when $\epsilon_{relative}$ reaches an acceptable value.

→ The difference between this "acceptable value" and the value for ∞ terms is the truncation error