Sep 26, 2025

When we design a numerical algorithm, we would like it to be

$\begin{cases} \text{Accurate} \\ \text{Fast} \end{cases}$ (in Finite Precision Arithmetic (FPA), aka. floating-point arithmetic)

How do we measure speed? We count floating point operations (FLOPS)

$+, -, \times, /$

Fact. Computers stores only a finite number of decimals

Ex. $\pi = 3.141592653 5 \ldots$

$\pi \approx 3.14$ FPA with 3 digits

$\pi \approx 3.1415$ FPA with 5 digits

Consequence. loss of accuracy

error accumulates with more floating point operations

Conclusion: Need to structure computation to minize errors.

Ex. FPA with 8 digits of accuracy

$10^7 = 10000000$

$10^{-7} = 0.0000001$

$$x = 10^7 (\underbrace{10 + 10^{-7}} - 10)$$

$10 + 10^{-7} = 10.0000001$

$\overset{FPA}{\approx} 10.000000 = 10$

$\Rightarrow x = 10^7 (10 - 10) = 0$

Better Strategy: Change the order of $+/-$:

$$x = 10^7(10-10+10^{-7}) = 10^7 \cdot 10^{-7} = 1$$

Count FLOPS $(+, -, \times, /)$ in an algorithm

Ex. Inner product between $v, w \in \mathbb{R}^n$

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} . \quad w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

$$\langle v, w \rangle = v^T \cdot w = [v_1 \ v_2 \ \cdots \ v_n] \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \sum_{i=1}^{n} v_i w_i$$

FLOP count? n multiplications, $n-1$ additions $\Rightarrow$ $2n-1$ FLOPS

Core idea.

```
p = 0;
for i = 1:n
    p = p + v(i)*w(i)
end
```

$\left.\begin{array}{l} \text{for } i=1:n \\ \quad p = p + v(i)*w(i) \\ \text{end} \qquad\quad 2 \end{array}\right\}$ $2n$ FLOPS.