Numerical Errors

Part 1: Introduction and Roundoff Errors

Outline (Part 1, Part 2)

- 1.1: Definitions
- 1.2: Roundoff Errors

This video

- 2.1: Truncation Errors
- 2.2: Total Numerical Error

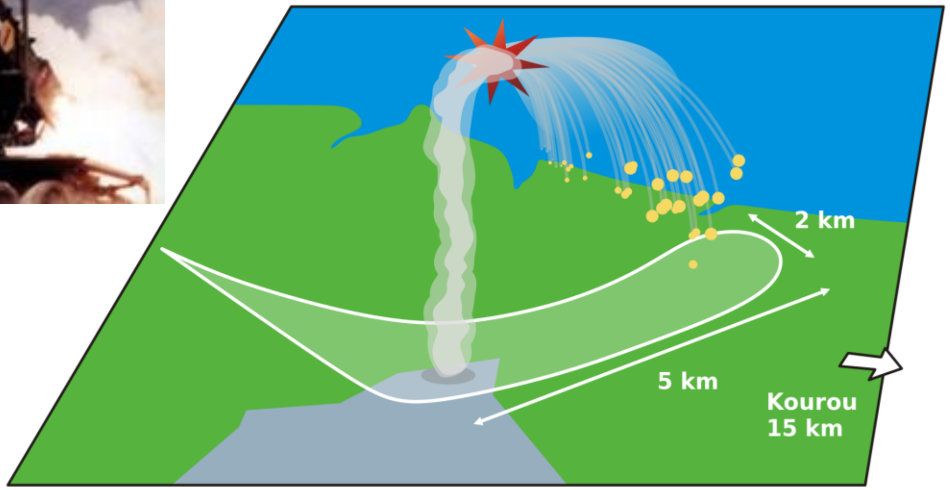
1.1: Definitions

Case Study: Vancouver Stock Exchange

- Vancouver Stock Exchange: index initialized at 1000 in January 1982
- Every time a stock price changed (~2,800 times/day), the index was recalculated to 4 decimals but only printed to 3
- Computer *truncated* 4th digit instead of *rounded*, causing the index to fall by ~1 point/day → ~20 points/month
- Error propagated over time:
  - 12 months: index = 725 instead of 960
  - 22 months: index corrected from 524.811 to 1098.892

Other Case Studies

- Patriot missile defense system failed to intercept a missile due to a number conversion error
- Ariane 5 rocket self-destructed 36 seconds after liftoff due to an overflow error
- German Parliament makeup changed due to improper rounding



Error Definitions

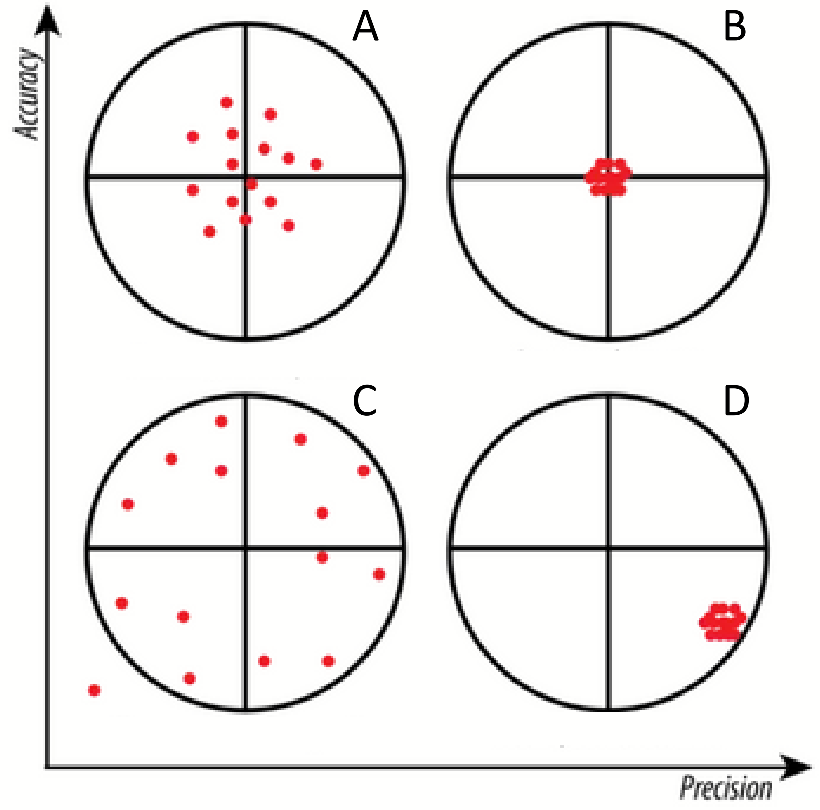
- Accuracy: how closely the computed value agrees *with the true value*
- Precision: how closely individually computed values agree *with each other*

A: accurate, imprecise

B: accurate, precise

C: inaccurate, imprecise

D: inaccurate, precise



Error Definitions

- Absolute error:

$$|\boldsymbol{E_t}| = |\boldsymbol{true} - \boldsymbol{approx}| \tag{3.1}$$

  - Shortcoming: doesn't account for order of magnitude

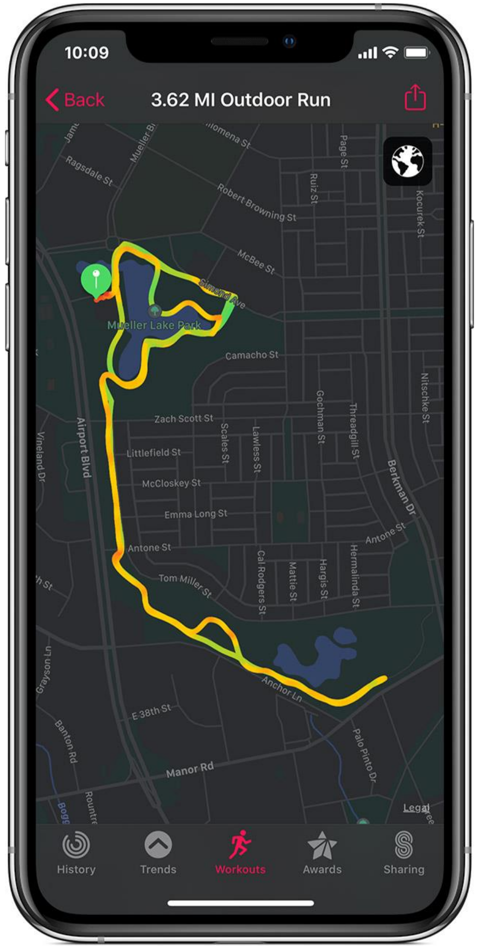
★ • True *relative* error/percent error:

$$|\mathbf{e}_{\boldsymbol{t}}| = \mathbf{100\%} * \frac{|\boldsymbol{true} - \boldsymbol{approx}|}{\boldsymbol{true}} \tag{3.2}$$

Error Definitions

- *Example*: You walk 100 ft but your phone's GPS measures 99 ft. The next day, you walk 10 ft but the GPS registers 9 ft. Compute and compare E_t and e_t.

	$\mathbf{100\ ft}$	$\mathbf{10\ ft}$
$\boldsymbol{E_t}$	$\lvert 100 - 99 \rvert = 1\ ft$	$\lvert 10 - 9 \rvert = 1\ ft$
$\boldsymbol{e_t}$	$100\% * \left\lvert \frac{100 - 99}{100} \right\rvert = 1\%$	$100\% * \left\lvert \frac{10 - 9}{10} \right\rvert = 10\%$



Error Definitions

- E_t and e_t require knowledge of the true value, but this is unknown in most practical applications
- Introduce percent relative error:

$$|e_a| = 100\% * \left|\frac{present - previous}{present}\right| \qquad (3.3)$$

- Iterate to a prespecified tolerance (stopping criterion) e_s:

$$|e_a| \leq e_s \qquad (3.4)$$

Error Definitions

- Percent relative error needed to achieve n significant digits of accuracy:

$$\boldsymbol{e_s = \left(0.5 * 10^{(2-n)}\right)\%} \qquad (3.5)$$

- *Example*: What e_s is required to achieve 4 digits of accuracy?

$$e_s = (0.5 * 10^{-2})\% = 0.005\%$$

1.2: Roundoff Errors

Roundoff Errors

- Roundoff errors arise because digital computers cannot represent some quantities exactly

- 2 major facets of roundoff errors:
  - Digital computers have *magnitude* and *precision* limits on their ability to represent numbers
  - Certain computations are highly sensitive to roundoff errors

Roundoff Errors

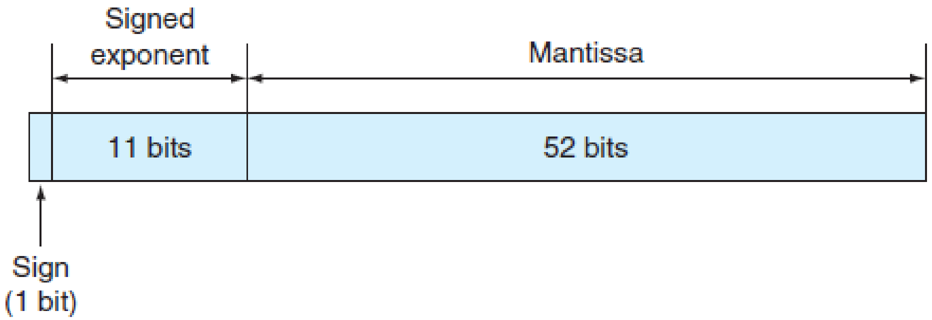
- Bit: *binary digit*
- Numbers are represented using *positional notation*

$(523)_{10}$		
100	10	1
5	2	3

$(10110)_2 = (22)_{10}$				
16	8	4	2	1
1	0	1	1	0

Roundoff Errors

- MATLAB expresses numbers using 64 bits:



$$n = \pm(1 + f) \times \ 2^{e}$$

- f: mantissa; $0 \leq f < 1$
- e: *signed* exponent; $-1022 \leq e \leq 1023$
  - Signed exponent: 0 = positive, 1 = negative

$(10110)_2$, signed exponent using 11 bits										
0	0	0	0	0	0	1	0	1	1	0

Roundoff Errors

- Largest (binary) positive number = $+1.111\ldots111 \times 2^{+1023}$
  - `realmax` = $1.7977 * 10^{308}$ (base 10)
- Smallest (binary) positive number = $+1.000\ldots000 \times 2^{-1022}$
  - `realmin` = $2.2251 * 10^{-308}$ (base 10)
- Largest (binary) negative number = $-1.111\ldots111 \times 2^{+1023}$
- Smallest (binary) negative number = $-1.000\ldots000 \times 2^{-1022}$

Roundoff Errors

$n = \pm(1+f) \times 2^{e}$

- Finiteness of f is a limitation on *precision*
  - 52 bits → ~15-16 decimals
- Finiteness of e is a limitation on *range*
  - 11 bits → -1022 to +1023
- Any numbers that don't meet these limitations must be approximated by ones that do
- Roundoff error ∝ magnitude
- Machine epsilon: maximum relative error
  - `eps`$= 2^{-52} = 2.2204 * 10^{-16}$

Roundoff Errors

- Some circumstances invoking roundoff errors:
  - Subtractive cancellation: subtracting 2 nearly equal numbers
  - Cumulative errors: a small roundoff error can cascade in subsequent computations
  - Adding a large and small number: digits are lost when a small number's mantissa is shifted to the right to be the same scale as the large number