# Digits and Rounding

Establish a relationship between '*accurate digits*' and rounding error.

$\pi = 3.1415\ldots$

$\tilde{\pi} \approx \underbrace{3.142}_{n=4}$

relative error $= \dfrac{\pi - \tilde{\pi}}{\pi} \leq 5 \cdot 10^{-4}$

$\leq 10^{-3}$

# Condition Numbers

Methods $f$ take input $x$ and produce output $y = f(x)$.
Input has (relative) error $|\Delta x| / |x|$.
Output has (relative) error $|\Delta y| / |y|$.
**Q:** Did the method make the relative error bigger? If so, by how much?

$$\kappa = \max_x \left( \frac{\text{relative change in } f(x)}{\text{rel. perturbation to } x} \right)$$

$$= \max_x \left( \frac{|f(x+\Delta x) - f(x)|}{|f(x)|} \Big/ \frac{|\Delta x|}{|x|} \right)$$

## absolute condition number

$$\kappa_{abs} = \max_{x} \left( \frac{|f(x+\Delta x) - f(x)|}{|\Delta x|} \right)$$

$$= \max_{\text{input}} \left( \frac{\text{absolute change of output}}{\text{size of perturbation to input}} \right)$$

$$\kappa(x) = \frac{|x|\,|f'(x)|}{|f(x)|}$$

$$f(x) = a + bx + cx^2 + \dots + y \cdot x^n$$

as $x \to \infty$, $f(x) \approx y x^n$

$$\kappa(x) \approx \frac{|x|\,|n y x^{n-1}|}{|y x^n|} = n$$

# $n$th-Order Accuracy

Often, *truncation error* is controlled by a parameter $h$.

Examples:

- distance from expansion center in Taylor expansions
- length of the interval in interpolation

A numerical method is called '$n$th-order accurate' if its truncation error $E(h)$ obeys

$$E(h) = O(h^n).$$

# Outline

# Wanted: Real Numbers... in a computer

Computers can represent *integers*, using bits:

$$23 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = (10111)_2$$

How would we represent fractions, e.g. 23.625?

$$23.625 = 10111.\underbrace{101}_{1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}}$$

10111.101 fixed-point

$$\underbrace{\qquad\qquad}_{32} . \underbrace{\qquad\qquad}_{32}$$

# Fixed-Point Numbers

Suppose we use units of 64 bits, with 32 bits for exponents $\geqslant 0$ and 32 bits for exponents $< 0$. What numbers can we represent?

$2^{31} + 2^{30} + \dots 2^{-32} \approx 2^{32} - 2^{-32} \approx 10^9$

$2^{-32}$

How many 'digits' of relative accuracy (think relative rounding error) are available for the smallest vs. the largest number?

9 decimal digits

$\underbrace{2^{31} \quad 2^{30} \quad \dots \quad 2^{-32}}$

1 accurate digit for $2^{-32}$

# Floating Point numbers

Convert $13 = (1101)_2$ into floating point representation.

$1.2345\ldots \times 10^6$

$1.101 \times 2^3$

$\underbrace{1101}_{\text{significand}}$ $\underbrace{11}_{\text{exponent}}$

What pieces do you need to store an FP number?

significand - binary digits

$s_0 \ldots s_n$

exponent - magnitude

$e_1 \ldots e_k$

$s_0 . s_1 s_2 s_3 \ldots s_n \cdot 2^{e_1 \ldots e_k}$

$n+k \approx$ bits

$s_0 = 1 \qquad s_1 \dots s_n = 0$

$e_1, e_2 \dots e_k = 1$

$$1.0 \cdot 2^{\underbrace{11111}} = 2^{2^k - 1}$$

$$k = 10 \qquad 2^{1024-1} = 2^{1023}$$

$$1.\underbrace{01011}_{\text{significand}}$$

$$1.s_0 s_1 \dots s_n \cdot 2^{e_1 \dots e_k}$$

$$= 0.1 s_0 \dots s_n \cdot 2^{e_1 \dots e_k + 1}$$

$$1011101.11001$$

$$1.01110111001 \times 2^{6}$$

$$.0001101$$

$$1.101 \cdot 2^{-4}$$

64-bit

1 | 13 | 50
--- | --- | ---
sign bit | exponent | significand

if exponent = 000000

$\hookrightarrow 2^{-2^{13}}$

roundoff error based machine epsilon

$\varepsilon$ is smallest such that $fp(1+\varepsilon) \neq fp(1)$

$exp = 000 \Rightarrow 2^{-4}$

$100 \Rightarrow 2^{0}$

$111 \Rightarrow 2^{3}$

$fp(1) = \underbrace{0}_{\text{sign}} \; \underbrace{100}_{\text{exp}} \; \underbrace{0 \ldots 0}_{\text{sig.}}$

$exp = 100 \Rightarrow \cancel{2^{0}} \; 2^{-4}$

$5 = 1.01 \cdot 2^2$

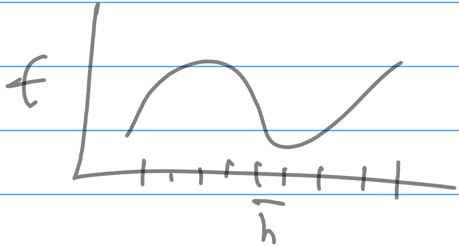
$s_1 \; s_2$

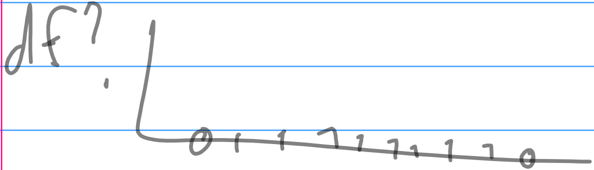
$6 = 1.10 \cdot 2^2$

$s_1$

$7 = 1.11$

$8 = 1.00 \cdot 2^3$ $\quad 9$


f
h


df?

$$fp(1) = \underbrace{0}_{\text{sign}} \ \underbrace{000}_{\text{exponent}} \ \underbrace{00000}_{\text{significand}}$$

$$fp(1+\varepsilon) = 0 \ \ 000 \ \ 00001$$

$$fp(exp) = 0|00 \Rightarrow 2^{-4} \Rightarrow 2^{-111}$$
$$= 1|00 \Rightarrow 2^{0}$$
$$= 1|11 \Rightarrow 2^{3}$$

## **In-class activity:** Floating Point