Announcements

- My Office hours will be MW 10-11am
- Homework is updated with instructions

Recap on Taylor

$$f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(c)}{k!}(x-c)^k + E_{n+1}$$

$$x \to x + h$$

$$c \to x$$

$$f(x+h) = \sum_{k=0}^{n} \frac{f^{(k)}(x)}{k!}h^k + E_{n+1}$$

with

$$E_{n+1} = \frac{f^{(n+1)}(\xi)}{(n+1)!}h^{n+1}$$

- Error term converges to zero with the same rate as h^{n+1}.
- Introduce big O notation, $E_{n+1} = O(h^{n+1})$, which means

$$|E_{n+1}| \leq C|h|^{n+1}$$

Examples

- It holds for every n

$$f(x+h) = \sum_{k=0}^{n} \frac{f^{(k)}(x)}{k!} h^k + E_{n+1}$$

- Some commonly used ones:

$$\begin{aligned}
f(x+h) &= f(x) + f'(\xi_1)h \\
&= f(x) + \mathcal{O}(h) \\
f(x+h) &= f(x) + f'(x)h + \frac{1}{2!} f''(\xi_2)h^2 \\
&= f(x) + f'(x)h + \mathcal{O}(h^2) \\
f(x+h) &= f(x) + f'(x)h + \frac{1}{2!} f''(x)h^2 + \frac{1}{3!} f'''(\xi_3)h^3 \\
&= f(x) + f'(x)h + \frac{1}{2!} f''(x)h^2 + \mathcal{O}(h^3)
\end{aligned}$$

Today's Overview

- Computers usually do not use base-10 numbers.
- Numbers that have a finite expression in one number system may have an infinite one in another, e.g. $\frac{1}{10} = (0.1)_{10} = (0.0\,0011\,0011\,0011\,0011\,0011\,0011\,0011\,0011\,\ldots)_2$
- We will discuss
  - Floating-point number system
  - Roundoff errors

Floating-point (FP) Repre.

- Decimal form
  - Integer part
  - A decimal point
  - Fractional part

For example,
37.2345, 0.0003541, -3093453.32

- Normalized scientific notation: leading digit in the fraction is NOT zero.
  - e.g., $37.2345 = 0.372345 \times 10^2$
- (Standard) scientific notation:
  - e.g., 2.99×10^8 m/s

Normalized FP Repre.

Normalized floating-point representation:

$$x = \pm r \times 10^{n}$$

- A sign that is either + or –
- A number $r \in [\frac{1}{10}, 1)$ $\qquad x = \pm 0.d_1 d_2 d_3 \ldots \times 10^n$
  - called normalized mantissa
- An integer power of 10
  - n is called exponent

Binary system

- If $x \neq 0$, it can be written as

$$x = \pm q \times 2^m \ (\frac{1}{2} \leq q < 1)$$

- The mantissa would be expressed a sequence of binary values (0 or 1)

$$q = (0.b_1 b_2 b_3 \cdots)_2$$

- $b_1 \neq 0 \rightarrow b_1 = 1 \rightarrow q \geq \frac{1}{2}$.
- Next example: list all the numbers can be expressed as $x = \pm(0.b_1 b_2 b_3)_2 \times 2^{\pm k}$ and $k = 0$ or 1

Example 1

$$
\begin{array}{lll}
(0.000)_2 \times 2^{-1} = 0, & (0.000)_2 \times 2^{0} = 0, & (0.000)_2 \times 2^{1} = 0 \\
(0.001)_2 \times 2^{-1} = \frac{1}{16}, & (0.001)_2 \times 2^{0} = \frac{1}{8}, & (0.001)_2 \times 2^{1} = \frac{1}{4} \\
(0.010)_2 \times 2^{-1} = \frac{2}{16}, & (0.010)_2 \times 2^{0} = \frac{2}{8}, & (0.010)_2 \times 2^{1} = \frac{2}{4} \\
(0.011)_2 \times 2^{-1} = \frac{3}{16}, & (0.011)_2 \times 2^{0} = \frac{3}{8}, & (0.011)_2 \times 2^{1} = \frac{3}{4} \\
(0.100)_2 \times 2^{-1} = \frac{4}{16}, & (0.100)_2 \times 2^{0} = \frac{4}{8}, & (0.100)_2 \times 2^{1} = \frac{4}{4} \\
(0.101)_2 \times 2^{-1} = \frac{5}{16}, & (0.101)_2 \times 2^{0} = \frac{5}{8}, & (0.101)_2 \times 2^{1} = \frac{5}{4} \\
(0.110)_2 \times 2^{-1} = \frac{6}{16}, & (0.110)_2 \times 2^{0} = \frac{6}{8}, & (0.110)_2 \times 2^{1} = \frac{6}{4} \\
(0.111)_2 \times 2^{-1} = \frac{7}{16}, & (0.111)_2 \times 2^{1} = \frac{7}{4}, & (0.111)_2 \times 2^{0} = \frac{7}{8}
\end{array}
$$

Only 25 distinct numbers!!

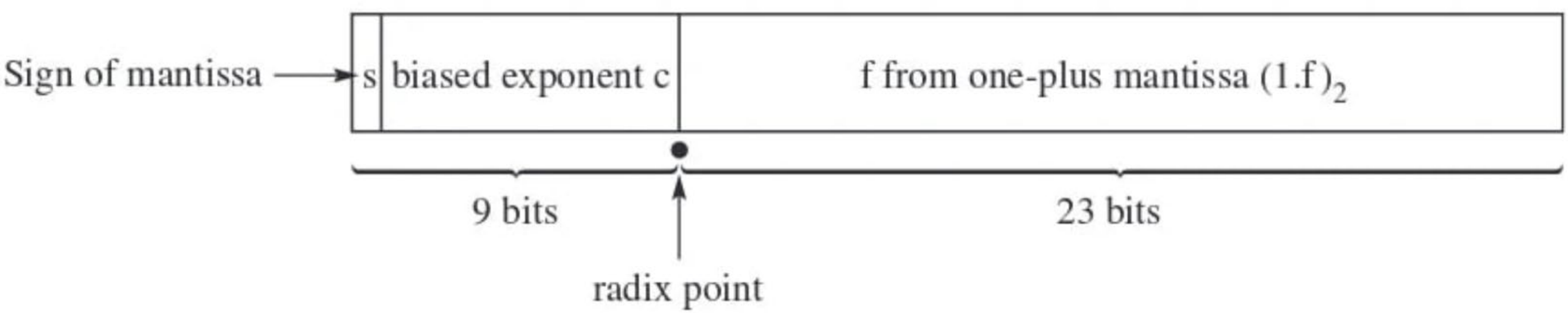
Computer number system

- Every computer can only represent a finite number of digits.
- The real numbers that are representable in a computer are called its machine number.
- Overflow/underflow describe something is too big/small.
- An overflow often results in a fatal error.
- An underflow is usually treated automatically by setting to zero with a warning message.

Common levels of precision

Precision	Bits	Sign	Exponent	Mantissa
Single	32	1	8	23
Double	64	1	11	52
Long Double	80	1	15	64

$$(-1)^s \times 2^{c-127} \times (1.f)_2$$



Single precision

- Recall $(-1)^s \times 2^{c-127} \times (1.f)_2$
- $0 < c < (11\,111\,111)_2 = 255$
  $\Rightarrow -127 < c - 127 < 128$
- $1 \leq (1.f)_2 = 2 - 2^{-23}$
- Largest machine number: 3.4×10^{38}
- Smallest machine number: 1.2×10^{-38}
- Machine epsilon: smallest number $1 + \epsilon \neq 1$
  - $\epsilon = 2^{-24} \approx 6 \times 10^{-8} \Rightarrow$ **7 significant decimal digits**

Example 2

Determine -52.234375 in single precision.

Double precision

- 11 bits for exponent and 52 for mantissa
- Largest machine number: 1.8×10^{308}
- Smallest machine number: 2.2×10^{-308}
- Machine epsilon: $2^{-53} \approx 1.11 \times 10^{-16}$
  - **15 significant decimal digits**

Computer errors

- The process of replacing a number by its nearest machine number is called correct rounding; the error involved is called roundoff error.
- If a number is overflow or underflow, roundoff error could be huge.

FP machine number

- Define fl(x) be the FL machine number that corresponds to x.
- The function fl depends on the computer.
- For a 32-bit word-length computer, we have

$$\frac{|x - \mathrm{fl}(x)|}{|x|} \leq u \qquad \left(u = 2^{-24}\right)$$

- The inequality can be expressed by

$$\mathrm{fl}(x) = x(1+\delta) \qquad \left(|\delta| \leq 2^{-24}\right)$$