lec 25

outline:

- FP
- quiz discussion
- non lin equations

Q's: don't know errors - how do we compute them?
(not the point!)
but: important question: when do we stop?
(when the guess stops changing)

10

## Floating Point Arithmetic

Want: Something like the real numbers... in a computer

$32 = 2^5$
$16 = 2^4$
$8 = 2^3$
$4 = 2^2$
$2 = 2^1$
$1 = 2^0$

Have: Integers, made of bits

$23 = 16 + 0 + 4 + 2 + 1$

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

How should we even represent fractions?

Idea: Keep going down past exponent zero

$23.625 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$
$+ 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \rightarrow (10111.101)_2$

So: Could store
- a fixed number of bits with exponents >= zero
- a fixed number of bits with exponents < zero

Suppose we use a 64-bit integer, with 32 bits >= 1 and 32 bits < 1.

What is the smallest number we can represent?

$(2^{31} \text{ —— } 32 \text{ —— } 2^0)(2^{-1} \text{ —— } 32 \text{ —— } 2^{-32})$

$2^{-32} = 10^{-10}$

What is the biggest number we can represent?

$2^{31} + 2^{30} + \ldots 2^0 + 2^{-1} + \ldots + 2^{-32} = 4 \cdot 10^9$

What's our range then?

$10^{-10} \ldots 10^9$

What happens if we multiply the largest number by 2?

Error

What happens if we divide the smallest number by 2?

0

How many accurate decimal digits do we have in a number near $10^9$ ?

$\sim 19$

How many accurate digits do we have in a number near $10^{-9}$ ?

$\sim 1?$

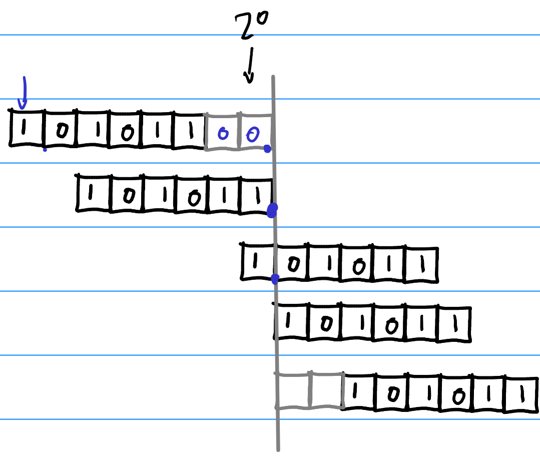
This is called fixed-point arithmetic, and it's pretty bad.

Should be able to do better.

Idea: Set a few bits aside to store the largest exponent. How?

$$\underbrace{1 \cdot 2^{213} + 0 \cdot 2^{212} + 1 \cdot 2^{211} +}_{\text{fixed \# digits}}$$

$$1 \cdot 2^{-213} + 0 \cdot 2^{-214} + \cdots$$




What is the:

| exponent? | "significand"? | value? |
|---|---|---|
| 7 | $(101011)_2 = 43$ | $1.34375 \cdot 2^7$ |
| 5 | — " — | $1.34375 \cdot 2^5$ |
| 0 | $(1.01011) = 1.34375$ | $1.34375 \cdot 2^0$ |
| -1 | 1.34375 | $1.34375 \cdot 2^{-1}$ |
| -3 | | $1.34375 \cdot 2^{-3}$ |

In our 64-bit example:

- 1 bit for sign (+/-)
- 11 bits for largest exponent
- 52 bits for "bits"

Exponent ranges from -1022 to 1023

This is called "double precision".

What is (very roughly) the smallest positive number we can represent?

$1 \cdot 2^{-1022}$ ← smallest exponent

$(1.0000\ldots0)_2 \cdot 2^{-1022} = (0.0\ldots01) \cdot 2^{-1022}$

What is (very roughly) the largest number we can represent?

$1 \cdot 2^{1023}$

How many accurate decimal digits do we have in the largest representable number?

largest: $2^{1023} \approx 10^{307}$

last bit of sig. $2^{1023-51} \approx 10^{292}$

$\Big\}$ ~15 digits

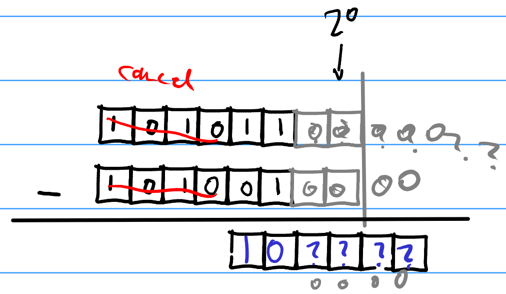
How many accurate decimal digits do we have in the smallest (positive) representable number?

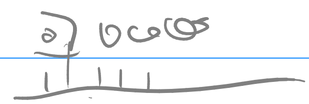
~ smallest: $2^{-1022} \approx 10^{-308}$

smallest number in significand: $10^{-323}$ } 15 digits

Same relative accuracy for numbers of every magnitude: Yay!

So what could possibly go wrong?



How many accurate (binary) digits are there in the above result?

2 out of 6

"catastrophic cancellation"

$e_{k+1} \sim C \cdot e_k^2$ ← quadratically conv