UCLA Math151A Fall 2021
Lecture 3
20210929
Root Finding with Bisection

Continue last time…

Big O notation

"on the order of",

$a(t) = O(b(t))$ with $b(t) > 0$ means

$$\exists\ C > 0 \text{ s.t. } |a(t)| \leq Cb(t) \text{ for } t \to 0 \text{ or } \infty$$

We mostly deal with $t \to 0$ case, and in such cases we often have $b(t) \to 0$.

$a(t) = O(b(t))$ is equivalent to saying

$$\lim_{t \to 0} \frac{|a(t)|}{b(t)} \quad \text{is bounded by a positive number.}$$

$$\boxed{\lim_{t \to 0} \frac{|a(t)|}{b(t)} \quad \text{is bounded by a positive number.}}$$

Example

The taylor theorem for $\cos(h)$ about 0 is

$$\cos(h) = 1 - \frac{1}{2}h^2 + \frac{1}{24}h^4 \cos(\xi(h)) \qquad \text{with some } 0 < \xi(h) < h.$$

Denote $f(h) = \cos(h) + \frac{1}{2}h^2 - 1 = \frac{1}{24}h^4 \cos(\xi(h))$.

$$\lim_{h \to 0} \frac{|f(h)|}{h^4} = \lim_{h \to 0} \frac{1}{24} |\cos(\xi(h))| \leq \frac{1}{24}$$

Thus by definition of big O notation,

$$f(h) = O(h^4).$$

The major topic for today…

Root Finding with Bisection

Optional reading: book 2.1

The **goal is to** find a root, or a zero, of a function f.

I.e., find p s.t. $f(p) = 0$.

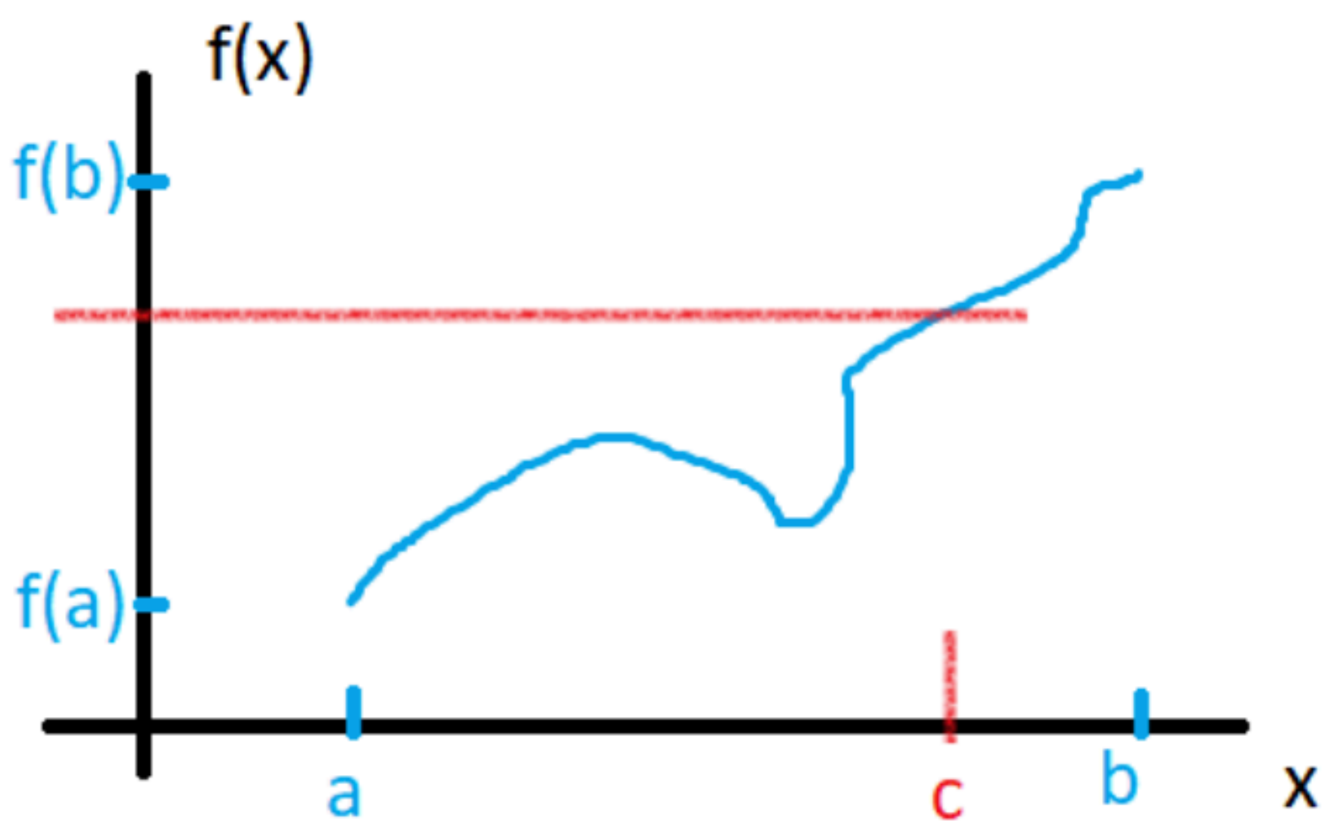
The **goal is to** find a root, or a zero, of a function f.

I.e., find p s.t. $f(p) = 0$.

We will assume (1) $f \in C([a,b])$,

(2) $f(a)f(b) < 0$

then $\exists p$ s.t. $f(p) = 0$. Why? I.V.P theorem tells so!



The goal of **numerical root finding** methods is

$$\boxed{\text{to find } p \text{ s.t. } f(p) = 0}$$

(or, that $f(p)$ is close to 0).
Due to numerical errors we typically cannot get it exact.

$$\boxed{\text{In contrast, I.V.P. only tells us that } p \text{ exists.}}$$

Example

$$f(x) = \sqrt(x) - \cos(x). \quad [a, b] = [0, 1].$$

$$f(0) = -1 \qquad f(1) = 1 - \cos(1) \approx 0.4597 > 0$$

Therefore by IVP, $\exists p \in (0, 1)$ s.t. $\sqrt{p} - \cos(p) = 0$.

$$\Leftrightarrow \sqrt{p} = \cos(p)$$

Bisection Method (B.M.)

B.M. is an algorithm to approximate p s.t. $f(p) = 0$, look for p on $[a, b]$.

Algorithm 1: Bisection Method (given $f(x) \in C([a,b])$, with $f(a)f(b) < 0$)

set $a_1 = a$, $b_1 = b$;

set $p_1 = \frac{a_1+b_1}{2}$;

if $f(p_1) == 0$ **then**

  We are done;

else if $f(p_1)$ *has same sign as* $f(a_1)$ **then**

  $p \in (p_1, b_1)$;

  set $a_2 = p_1$, $b_2 = b_1$

else if $f(p_1)$ *has same sign as* $f(b_1)$ **then**

  $p \in (a_1, p_1)$;

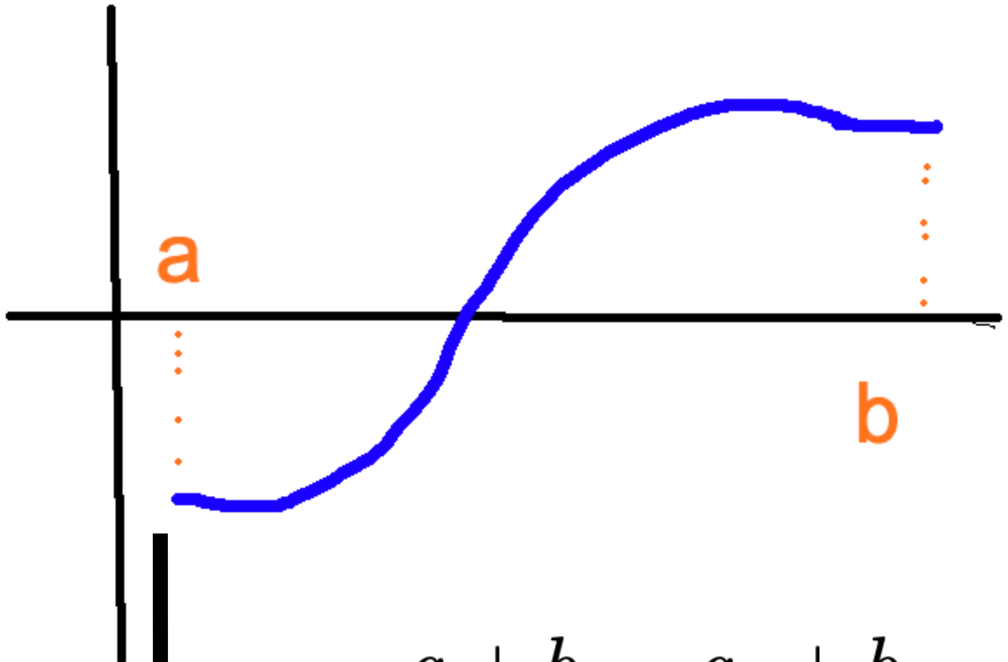
  set $a_2 = a_1$, $b_2 = p_1$.

end

set $p_2 = \frac{a_2+b_2}{2}$;

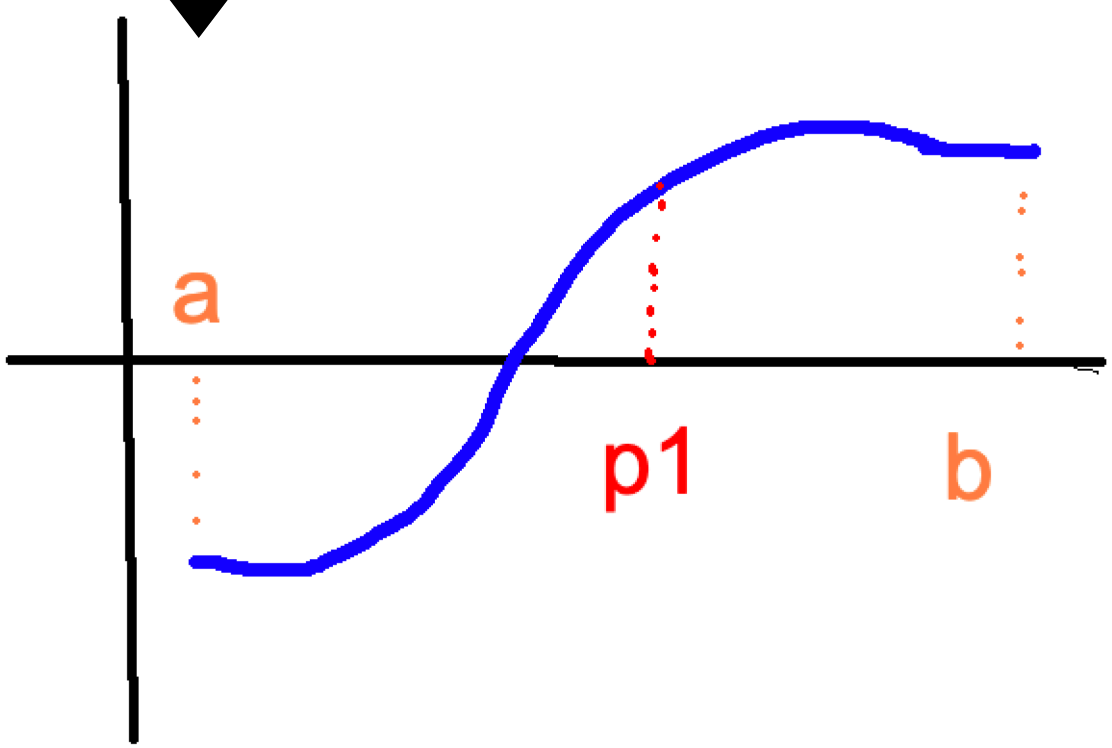
Repeat

Example



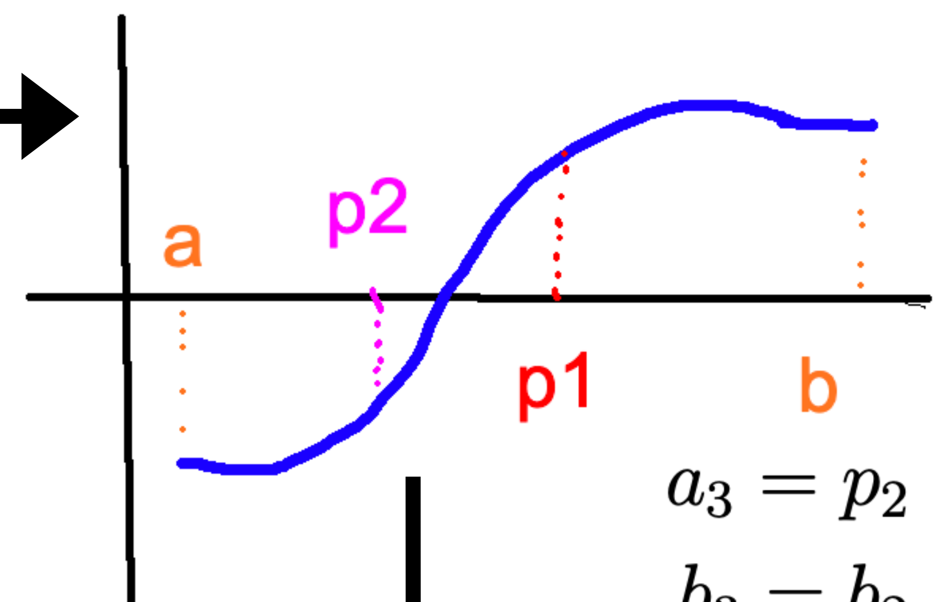
$$p_1 = \frac{a+b}{2} = \frac{a_1+b_1}{2}$$

set $a_1 = a$, $b_1 = b$;
set $p_1 = \frac{a_1+b_1}{2}$;
if $f(p_1) == 0$ **then**
  We are done;
else if $f(p_1)$ *has same sian as* $f(a_1)$ **then**
  $p \in (p_1, b_1)$;
  set $a_2 = p_1$, $b_2 = b_1$
else if $f(p_1)$ *has same sign as* $f(b_1)$ **then**
  $p \in (a_1, p_1)$;
  set $a_2 = a_1$, $b_2 = p_1$.
end
set $p_2 = \frac{a_2+b_2}{2}$;
Repeat



Since $f(p_1) > 0$,

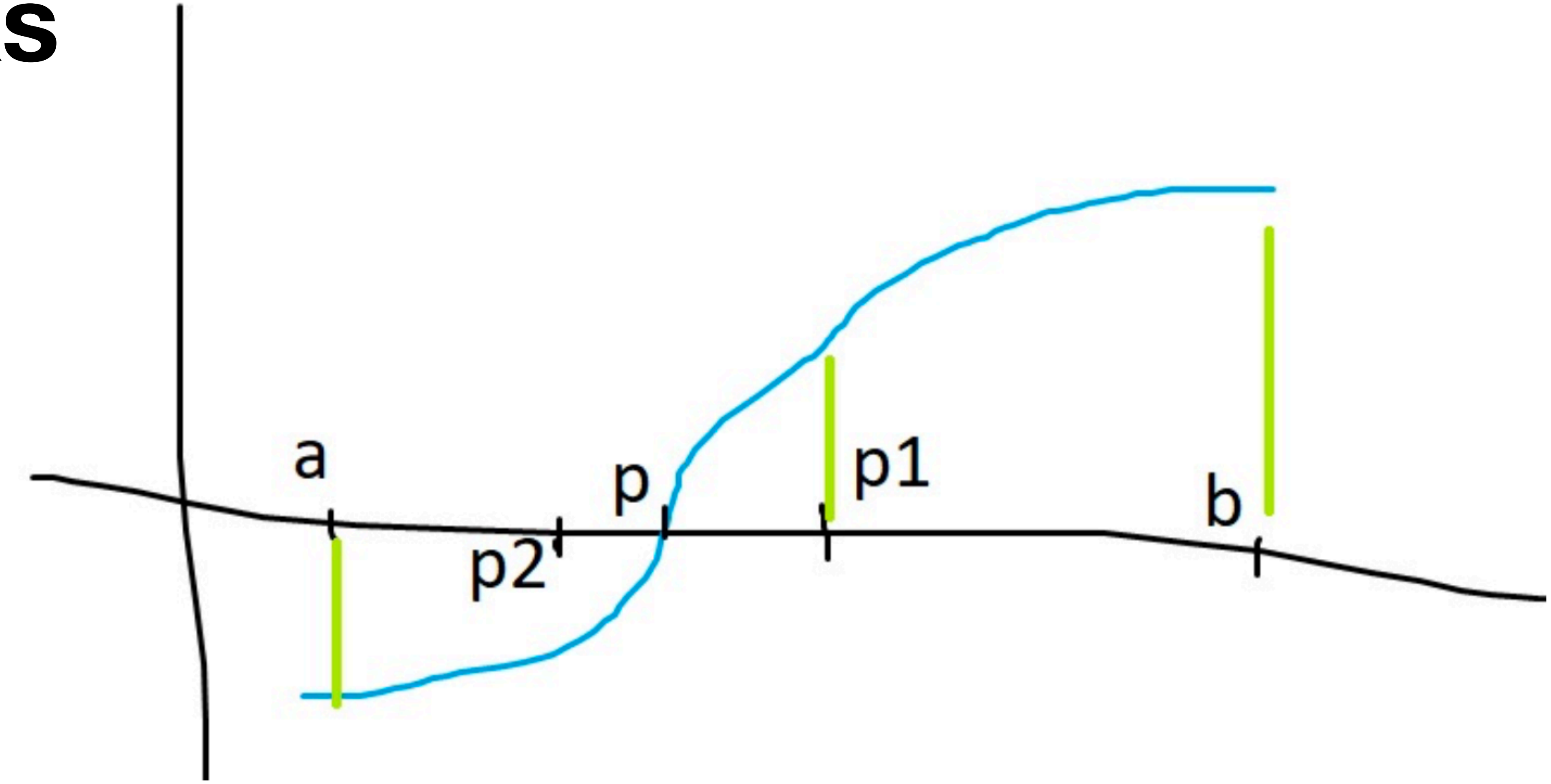
$$a_2 = a_1$$
$$b_2 = p_1$$
$$p_2 = \frac{a_2+b_2}{2}$$



Since $f(p_2) < 0$,

$$a_3 = p_2$$
$$b_3 = b_2$$
$$p_3 = \frac{a_3+b_3}{2}$$

Remarks



$\boxed{\text{Obviously, B. M. is similar to binary search in computer algorihtms.}}$

If $\exists$ multiple roots, for example, $\{p, q, r\} \in [a, b]$,
$f(p) = f(q) = f(r) = 0$,

then the B.M. is guaranteed to find exactly one root, not all of them.

But no guarantee exists for which one the method will find.

Stopping Criteria

We need a sequence $(p_1, p_2, \ldots)$ and need a sepcified tolerance ϵ.

choices for when to stop an algorithm:

- $|p_n - p_{n-1}| < \epsilon$
  like an aboslute differnce between successive elements of the sequence.

- $\frac{|p_n - p_{n-1}|}{|p_n|} < \epsilon$ (assumes $p_n \neq 0$)
  like a relative difference.

- $|f(p_n)| < \epsilon$
  sometimes called a **residual**: "how close are we to the answer".