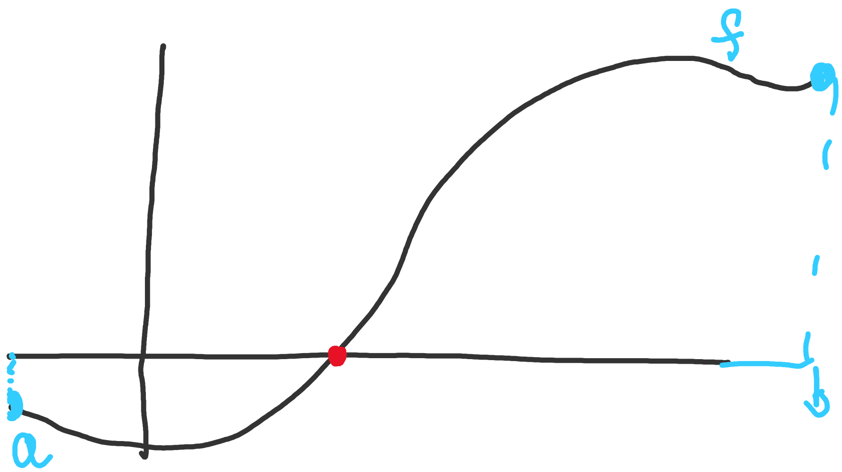
Section 2.1: Bisection Method

Breakout Room: Cutting in half helped us find te "number"

This is essentially te Bisection Method (similar to binary search).

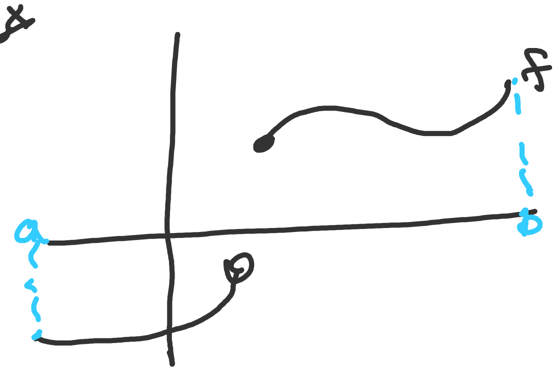


What guarantees that there will be a 0 if f(a) & f(b) are opposite signs?

IVT (intermediate value Thm)

Why must f be continuous?

ex



How does bisection method work?

* if f(a) & f(b) have different signs

& f is cont on $[a,b]$

- choose midpoint $p_1 = \frac{a+b}{2}$

  If $f(p_1) = 0$, then we're done.

  Otherwise $f(p_1)$ has the same sign as $f(a)$ or $f(b)$

  - set new interval (p_1, b) if sign $f(a) =$ sign $f(p_1)$
  - set new interval (a, p_1) if sign $f(b) =$ sign $f(p_1)$

& repeat all steps.

How do we know when to stop?

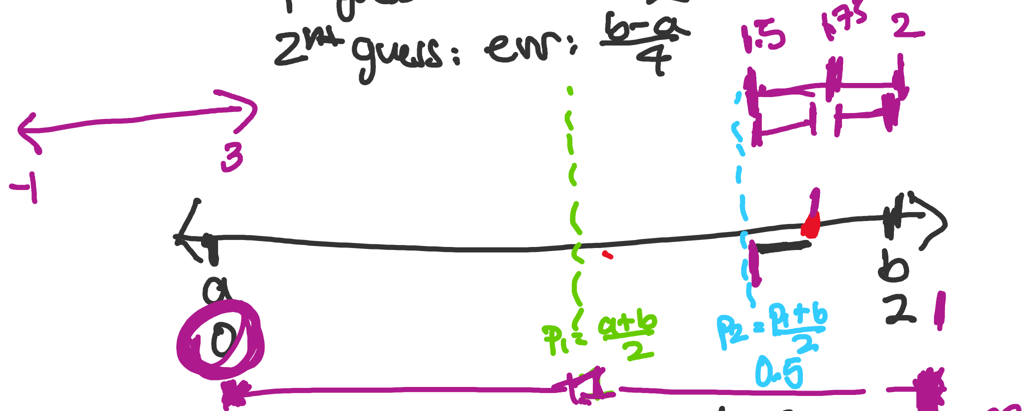
- if the error is sufficiently small
- too many iterations.

What is the max error at any guess?

$\boxed{\frac{1}{2} \text{ the current interval.}}$

1st guess: err $= \frac{b-a}{2}$ *

2nd guess: err: $\frac{b-a}{4}$



After n guess: err: $\frac{b-a}{2^n}$

OR we can calculate actual error (not theoretical)

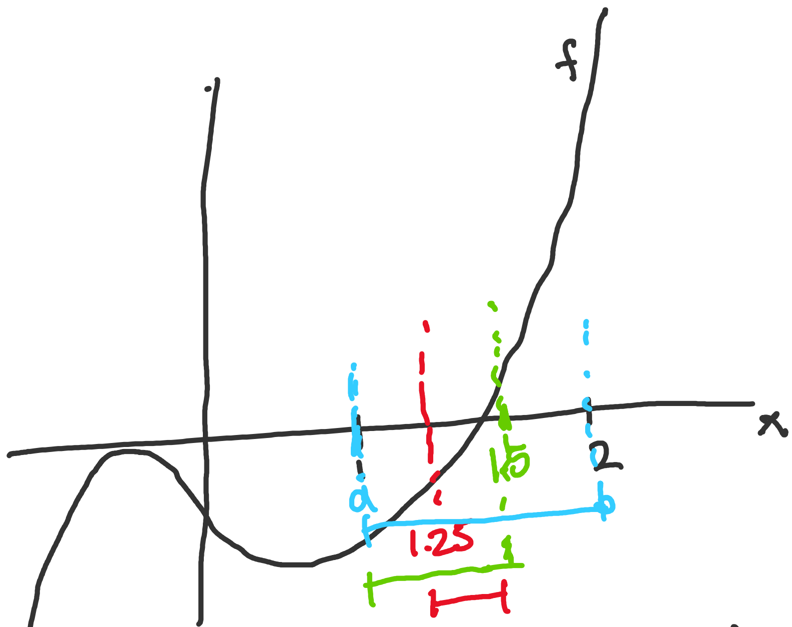
$|f(p_1) - 0|$

approx / guess   actual zero.

... $f(p_1)$ and 0.

abs. error between $f(p)$ and ...

ex Show that $x^3 + 4x^2 - 10 = 0$ has a root in $[1,2]$ and use Bisection method to approximate the root up to P_3. What is theoretical & actual error?



pf

$f(1) = 1^3 + 4(1)^2 - 10 = -5 < 0$

$f(2) = 2^3 + 4(2)^2 - 10 = 14 > 0$

$\Rightarrow$ by IVT $\exists$ a root in $[1,2]$ "there exists"

Let's do bisection method!

① Choose midpoint between 1, 2, $P_1 = 1.5$

evaluate $f(1.5) = 2.375 > 0$

new interval $[1, 1.5]$

② Choose midpoint of 1, 1.5, $P_2 = 1.25$

Evaluate $f(1.25) = -1.796875 < 0$

new interval $[1.25, 1.5]$

③ Choose midpoint of 1.25, 1.5, $P_3 = 1.375$

$f(1.375) > 0$

new interval $[1.25, 1.375]$

error is at most 0.0625 $\left(\frac{b-a}{2^n} : \text{error}\right)$

actual error ≈ 0.0098 (a lot less)

What is the rate of convergence for bisection method?

$$\underbrace{|p_n - p|}_{\text{absolute error}} \le \lambda \theta(b_n)$$

$$\le \frac{b-a}{2^n}$$

$$\le \underbrace{(b-a)}_{\lambda} \cdot \frac{1}{2^n}$$

The rate of convergence for bisection method is $\theta(\frac{1}{2})^n$.

Pros of bisection method:

- it always converges given enough guesses

Cons of bisection method

- slow! ($\theta(\frac{1}{2^n})$ rate of convergence)

ex/ find $e^x = x$, we can rewrite as $e^x - x = 0$. and use bisection method.

Pseudo code:

Inputs: f, a, b, N, tol

- f – function
- a, b – endpoints
- N – max # its
- tol – error tolerance

assume b>a

outputs: errbound, num_its, p.

- errbound – theoretical (or actual) error.
- num_its – number of iterations to get final approx
- p – final approx. to zero

initialize err, num_its

while err > tol or num_its < N

$p = \frac{a+b}{2}$ calc midpoint

err= abs $\frac{(b-a)}{2}$ cal error

num_its=num_its + 1 inc loop counter

if sign f(p) = sign f(b)

b=p

else

a=p

end % if

end% while