

Wrap up on Algorithms / Rootfinding and the bisection method

Friday, September 6, 2024 1:17 PM

Class 06: September 9, 2024

Recall: After wrapping up our discussion on stability and condition number, we covered important notation we will use to talk about the convergence and approximation properties of an algorithm (as a given parameter h goes to zero or number of iterations n goes to infinity): the big O and small o notation. We finally discussed how big O can also be used to talk about the growth of computational costs (time, floating point operations (flops), memory, energy, emissions) as the number of degrees of freedom (variables) N goes to infinity.

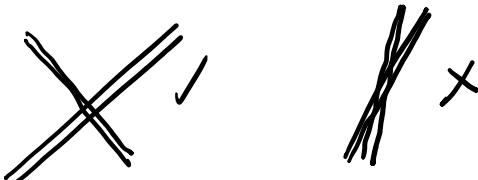
- We said $f(x)$ is "big O of x^p " as $x \rightarrow 0$ if, for small enough $|x|$, $f(x)$ is "sandwiched" between $+Mx^p$ and $-Mx^p$. That is, $|f(x)/x^p| < M$ for $|x| < \delta$. This is true, in particular, if the limit of $|f(x)/x^p|$ exists. In words, " $f(x)$ goes to zero at least as fast as x^p does".
- We then said $f(x)$ is "small o of x^p " as $x \rightarrow 0$ if the limit of $|f(x)/x^p|$ goes to zero. In words, " $f(x)$ goes to zero faster than x^p does".
- We then touched on the use of big O and little o when we have a sequence of iterates x_n that converge to a desired result x . We will similarly use big O to formalize the idea that $|x_n - x|$ goes to zero at least as fast as some other sequence (say, $1/n^3$), and little o to formalize the idea that it goes to zero faster.
- Finally, for quantities $C(N)$ which go to infinity as N goes to infinity, we will similarly use bounds and/or limits to formalize saying " $C(N)$ goes to infinity at least as fast as N^q ".
- We saw some examples of each of these, taking advantage of **Taylor expansions** and other calculus as tools to mathematically justify statements of this form.

Today, we will wrap up this section and begin our first module tackling solvers for a mathematical problem of general interest: **solution of non-linear equations and the equivalent problem of rootfinding**.

FIRST: A bit more on condition # for linear systems of eqs.

- PROBLEM: $A\vec{x} = \vec{b}$ for $n \times n$ invertible A .
INPUTS $\rightarrow A, \vec{b}$, OUTPUT: \vec{x}

INTUITION



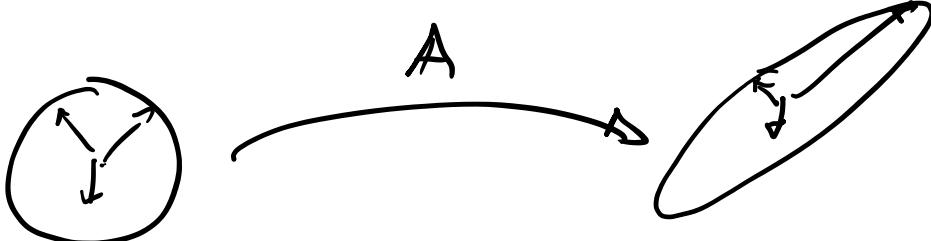
FOR SIMPLICITY, WE ASSUME $\tilde{b} = \vec{b} + \Delta \vec{b}$, but A is exact. What is the error in \tilde{x} ?

$$\text{REL ERROR IN INPUTS: } \frac{\|\vec{b} - \tilde{b}\|_2}{\|\vec{b}\|_2} = \frac{\|\Delta \vec{b}\|_2}{\|\vec{b}\|_2}.$$

REL ERROR IN OUTPUTS:

$$\begin{aligned} \frac{\|\vec{x} - \tilde{x}\|}{\|\vec{x}\|} &= \frac{\|A^{-1}\vec{b} - A^{-1}\tilde{b}\|}{\|\vec{x}\|} = \frac{\|A^{-1}(\vec{b} - \tilde{b})\|}{\|\vec{x}\|} \\ &= \frac{\|A^{-1}\Delta \vec{b}\|}{\|A^{-1}\vec{b}\|} = \frac{\|A^{-2}\Delta \vec{b}\|}{\|\vec{x}\|}. \end{aligned}$$

$$\begin{aligned}
 K_A(b) &= \lim_{\Delta b \rightarrow 0} \frac{\|A^{-1}\Delta b\| / \|A^{-1}b\|}{\|\Delta b\| / \|b\|} = \frac{\|A^{-1}\Delta b\|}{\|x\|} \\
 &= \lim_{\Delta b \rightarrow 0} \frac{\left(\frac{\|A^T \Delta b\|}{\|\Delta b\|} \right) \cancel{\left(\frac{\|A_x\|}{\|x\|} \right)}}{\cancel{\left(\frac{\|A^T \Delta b\|}{\|\Delta b\|} \right)} \cancel{\left(\frac{\|x\|}{\|x\|} \right)}} \leq \underbrace{\|A^T\|_2 \|A\|_2}_{\text{cond}(A)}
 \end{aligned}$$



$$\boxed{\|M\|_2} = \max_{\vec{x} \neq 0} \frac{\|Mx\|_2}{\|x\|_2} = \max_{\|x\|_2=1} \|Mx\|_2$$

$$\|Mx\| \leq \|M\|_2 \|x\|_2 \text{ for all } x.$$

$$A = U \Sigma U^*$$

↗ diag $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$
↘ unitary.

$$\boxed{K_A = \frac{\max |\lambda_i|}{\min |\lambda_i|}}.$$

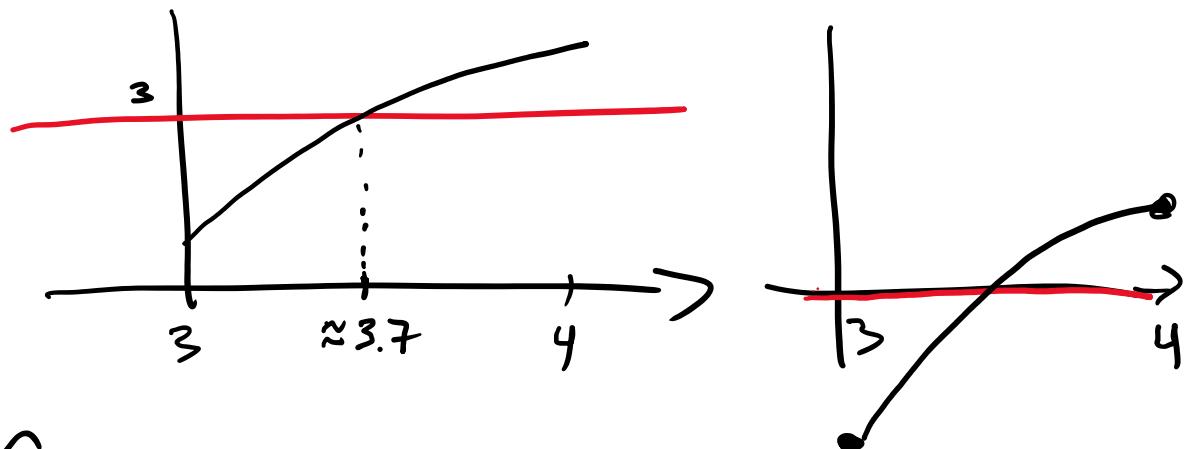
In general, you need $A = U \Sigma V^*$
where U, V are unitary, $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots)$

$$\boxed{K_A = \frac{\sigma_{\max}}{\sigma_{\min}}} \rightarrow \sigma_1 / \sigma_n.$$

NON-LINEAR EQUATIONS & ROOTFINDING.

"Find x such that $f(x) = d$."

$$\underline{x + \cos(x) = 3} \quad (x \in [3, 4])$$

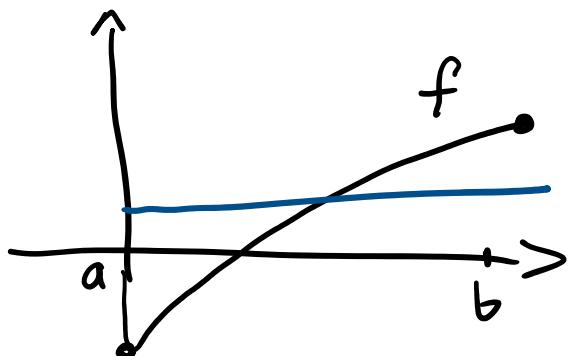


Problem

$$\underline{x + \cos x - 3 = 0}$$

x such that $f(x) = 0 \rightarrow \underline{\text{ROOTS}}$
↳ ROOTFINDING PROBLEM

BISECTION METHOD



• f is continuous on $[a, b]$.

Intermediate Value Thm

$y_a = f(a) \neq y_b = f(b)$

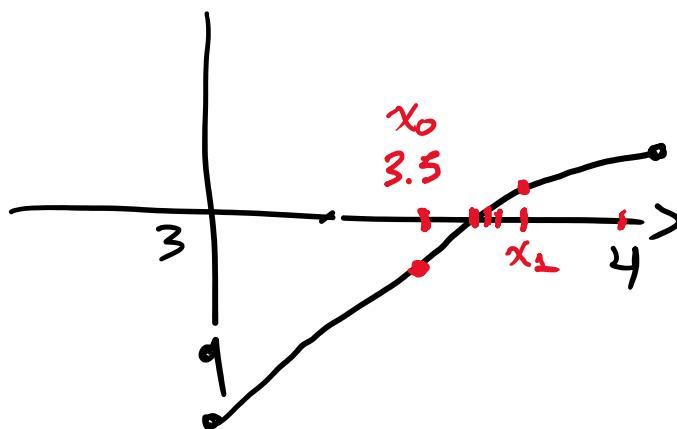
"Given any $y \in (y_a, y_b)$,
there is $c \in (a, b)$ s.t.

✓

»

Given any $y \in (y_a, y_b)$,
there is $c \in (a, b)$ s.t.
 $f(c) = y$.

- $f(a)f(b) < 0 \rightarrow$ change of sign.



• GUESS $x_0 = 3.5$.

• EVAL $f(3.5)$

□ $f(3.5) > 0$

Change of sign $[3, 3.5]$
OR

$f(3.5) < 0$

Change of sign $[3.5, 4]$

PSEUDO CODE:

INPUTS : $a, b, f(x), n_{\max}, TOL$.

OUTPUTS : r

① Initialize: $n=0, x_0 = (a+b)/2, a_0=a, b_0=b$.

② WHILE $(n \leq n_{\max} \text{ AND } \underline{(b_n - a_n)} \geq 2TOL)$

③ IF $f(x_n) == 0$

 return $r = x_n$;

ELSE IF $f(x_n)f(a_n) < 0$

$a_{n+1} = c_n; b_{n+1} = x_n;$

ELSE

$a_{n+1} = x_n; b_{n+1} = b_n;$

$x_{n+1} = (a_{n+1} + b_{n+1})/2;$

$n = n + 1;$

$n = n + 1;$

END

$r = x_n; \text{ return } r;$