MACHINE EPSILON: eps

$$\text{eps}_{SP} = 2^{-24} \sim 6 \cdot 10^{-8}$$

$$\text{eps}_{DP} = 2^{-53} \sim 10^{-16}$$

$fl: \mathbb{R} \to \mathbb{F}$ Maps real #'s onto the floating point set

countably infinite set

For all $x \in \mathbb{R}$ $\exists$ $x' \in \mathbb{F}$ st

$$\frac{|x - x'|}{|x|} \leq \text{eps}$$

$\therefore$ For all $x \in \mathbb{R}$ $\exists$ ε st $|\varepsilon| \leq \text{eps}$

$$fl(x) = x(1+\varepsilon).$$

The difference between a floating point number and a real number closest to it is always smaller than eps.

FLOATING POINT ARITHMETIC

The operations $+, -, \times, \div$ have

compute analogues: $\oplus, \ominus, \otimes, \oslash \equiv \circledast$

$$x \circledast y = fl(x * y)$$

for all $x', y' \in \mathbb{F}$ $\exists$ ε $|\varepsilon| < \text{eps}$ st

$$x' \circledast y' = (x * y)(1+\varepsilon)$$

where $x, y \in \mathbb{R}$

$\therefore$ every fl arithmetic operation is exact up to a relative error of size, at most, eps.

Rmk: Complex floating point arithmetic is done via software. //

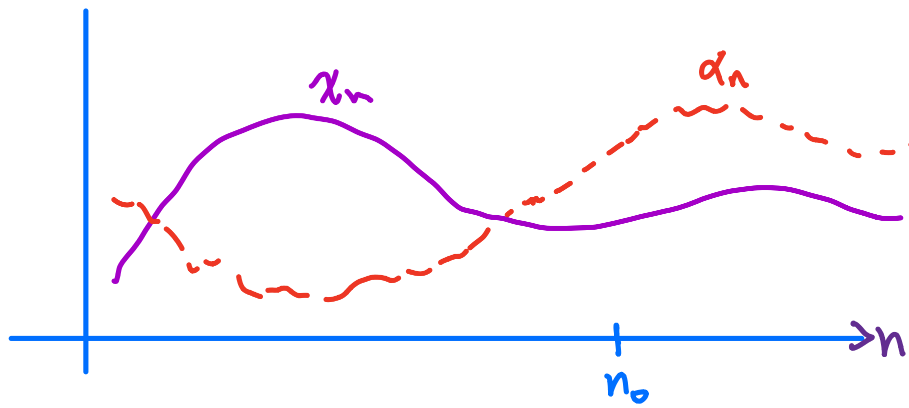
LECTURE 14 Stability

Big "oh" $\mathcal{O}$ & Little o Notation:

S'pse $\{x_n\}$, $\{\alpha_n\}$ 2 sequences

if $x_n = \mathcal{O}(\alpha_n)$

Big "Oh"

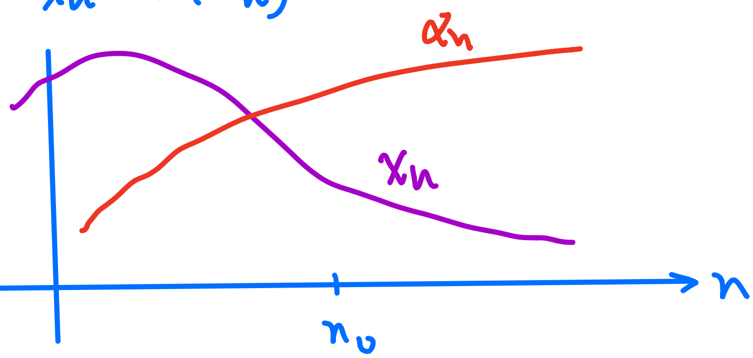


$\exists$ K, constant and n_0 constant s.t.

$|x_n| \leq K |\alpha_n|$ for $n \geq n_0$

$$\lim_{n\to\infty} \frac{|x_n|}{|\alpha_n|} \leq K$$

If $x_n = o(\alpha_n)$ "little oh"



if $\lim_{n\to\infty} \frac{|x_n|}{|\alpha_n|} = 0$

Every o is O but not the other way around.

ex) $h(n) = 1 + 4e^{2n}$
$g(n) = e^{2n}$

$$\lim_{n\to\infty} \frac{h(n)}{g(n)} = \lim_{n\to\infty} \frac{1+4e^{2n}}{e^{2n}} = 4$$

$$h(n) = O(g(n))$$

ex) $f(n) = e^{n}$ $\quad g(n) = e^{2n}$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{e^{n}}{e^{2n}} = 0$$

$$\therefore f(n) = o(g(n))$$

//

We might be looking at what happens when $\varepsilon \to 0$

say $g(\varepsilon) = o(f(\varepsilon))$

$$\lim_{\varepsilon\to 0} \frac{g(\varepsilon)}{f(\varepsilon)} = 0$$

e.g. $\displaystyle \lim_{\varepsilon\to 0} \frac{\sqrt{\varepsilon} + 3\varepsilon^3}{1+4\varepsilon^2} = 0$

//

ACCURACY = FIDELITY

Let $\bar{f}$ be an algorithm that computes f

Definition:

Abs err (absolute error) $\equiv \|\bar{f}(x) - f(x)\|$

Rel err (relative error) $\equiv \dfrac{\|\bar{f}(x) - f(x)\|}{\|f(x)\|}$

for some x.

Relative Accuracy $\equiv \dfrac{\|\bar{f}(\bar{x}) - f(x)\|}{\|f(x)\|}$

want this to be $O(\text{eps})$ //

STABILITY $\bar{f}$ is relatively STABLE if

$$\frac{\|\bar{f}(x) - f(\bar{x})\|}{\|f(\bar{x})\|} = O(\text{eps})$$

for some $\bar{x}$ $\dfrac{\|\bar{x} - x\|}{\|x\|} = O(\text{eps})$ //

We note that we compute $\bar{f}(\bar{x}) \neq \bar{f}(x) \neq f(\bar{x})$ generally.

BACKWARD STABILITY

The algorithm $\tilde{f}$ is backward stable if

$$\tilde{f}(x) = f(\tilde{x}) \text{ for some } x \text{ with } \frac{\|\tilde{x}-x\|}{\|x\|} = O(\text{eps})$$

Gives the right answer using a nearby input.

Rmk: Since all norms are "equivalent" on finite dimensional problems, an algorithm will be stable (or otherwise) regardless of the (unweighted) norm used.

LECTURE 15

Example of backward stability:

Subtraction

$$f = x_1 - x_2$$

$$\tilde{f}(\tilde{x}) = fl(x_1) \ominus fl(x_2)$$

$$fl(x_1) = x_1(1+\varepsilon_1) \qquad |\varepsilon_1| < \text{eps}$$

$$fl(x_2) = x_2(1+\varepsilon_2) \qquad |\varepsilon_2| < \text{eps}$$

$$\bar{f}(\bar{x}) = \left[x_1(1+\varepsilon_1) - x_2(1+\varepsilon_2)\right](1+\varepsilon_3)$$

$|\varepsilon_3| < \text{eps}$

$$\therefore \ fl(x_1) \ominus fl(x_2) = x_1(1+\varepsilon_1)(1+\varepsilon_3) - x_2(1+\varepsilon_2)(1+\varepsilon_3)$$

$$= x_1(1+\varepsilon_1+\varepsilon_3+\varepsilon_1\varepsilon_3) - x_2(1+\varepsilon_2+\varepsilon_3+\varepsilon_2\varepsilon_3)$$

$\varepsilon_1\varepsilon_3$ & $\varepsilon_2\varepsilon_3 = O(\text{eps}^2)$

we ignore

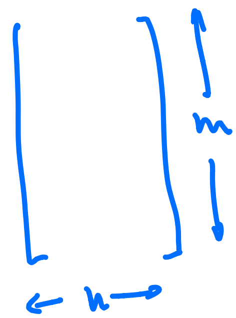
$$\therefore \ fl(x_1) \ominus fl(x_2) = x_1(1+\varepsilon_4) - x_2(1+\varepsilon_5)$$

$\therefore \ \bar{f}(x) = fl(x_1) \ominus fl(x_2)$ is equal to the difference of $\bar{x}_1$ and $\bar{x}_2$ where

$$\frac{|\bar{x}_1 - x_1|}{|x_1|} \leq \text{eps} \qquad \frac{|\bar{x}_2 - x_2|}{|x_2|} \leq \text{eps}$$

$\therefore$ Backward stable operation.

ex) $A = xy^*$ $\quad x \in \mathbb{C}^m \quad y \in \mathbb{C}^n$

$A =$ 

$\bar{A}(x,y) = fl(x) \otimes fl(\bar{y})^*$

$fl(x) = x + \delta x$

$fl(y) = y + \delta y$

$$fl(x) \otimes fl(\bar{y})^*$$

$$= (x+\delta x) \otimes (y+\delta y)^*$$

$$= (x+\delta x)\,(y+\delta y)^*(I+\delta)$$

so let $\bar{x} = x+\delta x \quad \bar{y} = y+\delta y$

$\bar{x}\,\bar{y}^*(I+\delta)$ could find neighboring $\bar{x}$ & $\bar{y}$

but generally this δ destroys the rank 1 nature of the exact product.

AXIOMS OF FLOATING POINT ARITHMETIC

(I) $\forall x \in \mathbb{R}$ $\exists$ ε $|\varepsilon| < eps$

st $fl(x) = x(1+\varepsilon)$

(II) For all $x, y \in \mathbb{R}$ $\exists$ ε $|\varepsilon| < eps$ st

$x \circledast y = x * y(1+\varepsilon)$

ACCURACY OF BACKWARD STABLE ALGORITHMS

Thm: Spse BW stable algorithm. Apply it to solve a problem $f : X \to Y$ w/ condition number κ on a computer satisfying Floating Point axioms:

$$\Rightarrow \text{Rel err } \frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(\kappa(x)\, eps)$$

This means that algorithmic choices that improve the conditioning, by making it smaller, can have a significant impact on accuracy, as well as on stability.