Stability of Algorithms:-

1. **Desirable** :- Exact solutions to numerical problems

2. **Reality** :- Problems are continuous while computer arithmetic is discrete

3. Stability tells us what it means to get the best answer even if this is not the exact answer!

Algorithm:-

An abstract way to think about solving a problem is evaluating function

$$f : X \rightarrow Y$$

X : vector space of data

Y : vector space of solutions

$y = f(x)$ where $x \in X$, $y \in Y$

An algorithm can be viewed as a different function $\tilde{f}$ that usually takes the same data $x \in X$ and maps it to a result which is a collection of floating point numbers that belongs to Y.

<u>Accuracy</u>:- A good algo should have an $\tilde{f}$ that closely approximates the underlying problem f.

→ <u>Absolute</u> <u>error</u> of <u>computation</u>:-

$$\|\tilde{f}(x) - f(x)\|$$

→ <u>Relative</u> <u>error</u> of <u>computation</u>:

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|}$$

We say that $\tilde{f}$ is an accurate algorithm for f if for all relevant input data x,

<u>Forward</u> <u>relative</u> <u>error</u>

$$\boxed{\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(\varepsilon_M)}$$

"On the order of machine epsilon"

If f is ill-conditioned

i.e $\max_{\delta x} \dfrac{\frac{\|\delta f\|}{\|f\|}}{\frac{\|\delta x\|}{\|x\|}} = \hat{K}$ is very large.

$\dfrac{\|\delta x\|}{\|x\|} \lesssim O(\varepsilon_M)$

$\dfrac{\|\delta f\|}{\|-\|} \lesssim K\, O(\varepsilon_M)$

Very ambitious to esign an algo $\tilde{f}$ such that relative error in computation is $O(\varepsilon_M)$

Instead of aiming for accuracy of an algo, the most we can aim is for stability!

We can say that an algorithm $\tilde{f}$ for solving a problem f is stable if for all (relevant) input data x

$$\frac{\|\tilde{f}(x) - f(\tilde{x})\|}{\|f(\tilde{x})\|} = O(\varepsilon_M)$$

for some $\tilde{x}$ satisfying $\dfrac{\|\tilde{x} - x\|}{\|x\|} = O(\varepsilon_M) \leq C\,\varepsilon_M$

i.e A stable algorithm gives nearly right answer to nearly right question.

Backward stability :-

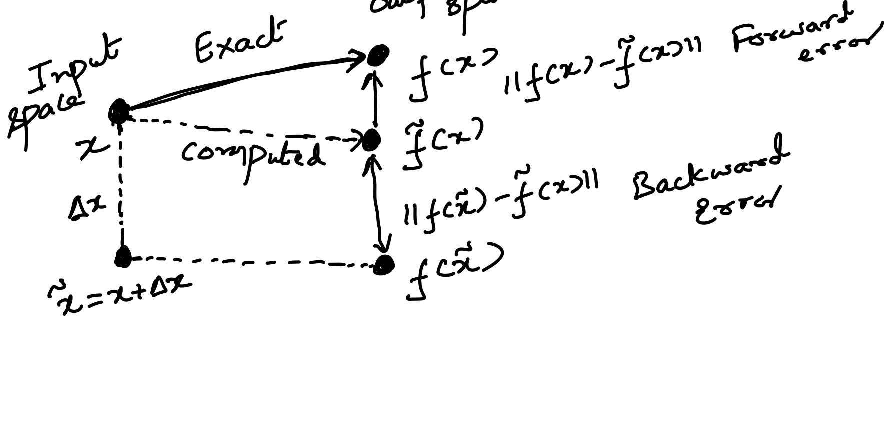
Backward error :- $\|f(\tilde{x}) - \tilde{f}(x)\|$

What is the input $\tilde{x}$ for which my algo with input x has exactly computed a solution for

$$f(\tilde{x}) = \tilde{f}(x)$$

Then such an algo $\tilde{f}$ for problem f is backward stable. (i.e Backward error is zero)

i.e Exactly right answer to a nearly right question!



Stability of Floating point arithmetic operations!

$+, -, \times, /$ (Classical arithmetic operations)

Floating point analogues $\oplus, \ominus, \otimes, \oslash$

Recall Floating point axioms :-

$$fl(x) = x(1+\varepsilon) \text{ where } |\varepsilon| < \varepsilon_M$$

$$x \circledast y = x * y(1+\varepsilon) \text{ where } |\varepsilon| < \varepsilon_M$$

Example 1:-

Floating point arithmetic for $\ominus$

$\underset{\sim}{x} \in \mathbb{R}^2$ $\quad \underset{\sim}{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$f(\underset{\sim}{x}) = x_1 - x_2$

$f(x_1, x_2) = x_1 - x_2$

$x_1 \to fl(x_1)$; $x_2 \to fl(x_2)$

$fl(x_1) = x_1(1+\varepsilon_1)$; $fl(x_2) = x_2(1+\varepsilon_2)$

$\varepsilon_1, \varepsilon_2$ satisfy $|\varepsilon_1|, |\varepsilon_2| = O(\varepsilon_M)$

Algo :- $fl(x_1) \ominus fl(x_2)$

$\Rightarrow [x_1(1+\varepsilon_1)] \ominus [x_2(1+\varepsilon_2)]$

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

where $|\varepsilon_3| = O(\varepsilon_M)$

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

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

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

where $|\varepsilon_4|, |\varepsilon_5| \leq 2\varepsilon_M + O(\varepsilon_M^2)$

$= O(\varepsilon_M)$

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

$$= \tilde{x}_1 - \tilde{x}_2 = f(\tilde{x}_1, \tilde{x}_2)$$

Algo is backward stable!

Example 2:-

Outer product between two vectors :-

$\underset{\sim}{x} \in \mathbb{R}^m, \ \underset{\sim}{y} \in \mathbb{R}^n$

compute outer product $\underset{\sim}{A} = \underset{\sim}{x}\, \underset{\sim}{y}^T$

$\boxed{A_{ij} = x_i y_j}$ $= m\times 1 \ 1\times n$

$\underset{\sim}{A}$ $m\times n$

↑ rank 1 matrix

$$\tilde{f}(\underset{\sim}{x}, \underset{\sim}{y}) = \tilde{A}_{ij}$$

$$= fl(x_i) \otimes fl(y_j)$$

$$\tilde{A}_{ij} = fl(x_i) \otimes fl(y_j)$$

$$= [fl(x_i) \times fl(y_j)](1+\varepsilon_3^{ij})$$

$$= [x_i(1+\varepsilon_1^i)\, y_j(1+\varepsilon_2^j)](1+\varepsilon_3^{ij})$$

$$= x_i y_j (1+\varepsilon_1^i)(1+\varepsilon_2^j)(1+\varepsilon_3^{ij})$$

$$= x_i y_j (1 + \underbrace{\varepsilon_1^i + \varepsilon_2^j + \varepsilon_1^i \varepsilon_2^j}) (1 + \varepsilon_3^{ij}) \leftarrow (i)$$

$$= x_i y_j (1 + \underline{\varepsilon_4^{ij}}) (1 + \underline{\varepsilon_3^{ij}})$$

I want to verify

$$\boxed{\tilde{f}(\underset{\sim}{x}, \underset{\sim}{y}) = \underset{\sim}{\tilde{x}}\, \underset{\sim}{\tilde{y}}^T = (\underset{\sim}{x} + \delta \underset{\sim}{x})(\underset{\sim}{y} + \delta \underset{\sim}{y})^T} \leftarrow (*)$$

This algo is not backward stable because from eqn (i), it is not possible for my algo always to give a perturbed rank 1 matrix of the form $\underset{\sim}{\tilde{x}}\, \underset{\sim}{\tilde{y}}^T$

Is this algo stable? Exercise!

<u>Example 3</u>:-

Adding 1 to floating point number!

Let $x \in \mathbb{R}$ and $f(x) = x + 1$

$$\tilde{f}(x) = fl(x) \oplus 1$$

$$\tilde{f}(x) = (fl(x) + 1)(1 + \varepsilon_1)$$

$$= (x(1 + \varepsilon_2) + 1)(1 + \varepsilon_1)$$

$$= (1 + x + x\varepsilon_2)(1 + \varepsilon_1)$$

$$= 1 + x + x\varepsilon_2 + \varepsilon_1 + \varepsilon_1 x + x\varepsilon_2\varepsilon_1$$

$$= 1 + x\left[1 + \varepsilon_1 + \varepsilon_2 + \frac{\varepsilon_1}{x}\right] \quad — \quad (2)$$

For backward stability is $\tilde{f}(x) = f(\tilde{x})$

$$= \tilde{x} + 1$$

$$= x(1+\varepsilon) + 1 \quad ?$$

From eq (2)

$$1 + x\underbrace{\left[1 + \varepsilon_1 + \varepsilon_2 + \frac{\varepsilon_1}{x}\right]}_{\text{As } x \to 0}$$

$\varepsilon \neq O(\varepsilon_M)$

This is not backward stable

Accuracy of Backward stable Algo:-

$\boxed{\text{Thm:-}}$ If a backward stable algo is applied to solve a problem f with condition number K, the relative forward errors satisfy

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(K(x)\,\varepsilon_M)$$

Pf:- By definition of backward stability we have $\tilde{f}(x) = f(\tilde{x})$

for $\dfrac{\|\tilde{x} - x\|}{\|x\|} = O(\varepsilon_M)$

we also know

$$K(x) = \max_{\delta x} \frac{\frac{\|\delta f\|}{\|f\|}}{\frac{\|\delta x\|}{\|x\|}}$$

$$\frac{\left(\frac{\|f(\tilde{x}) - f(x)\|}{\|f(x)\|}\right)}{\frac{\|\tilde{x} - x\|}{\|x\|}} \leq K(x)$$

But B.S, $f(\tilde{x}) = \tilde{f}(x)$

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} \leq K(x) \underbrace{\frac{\|\tilde{x} - x\|}{\|x\|}}_{O(\varepsilon_M)}$$

$$O(K(x)\,\varepsilon_M) \quad — \quad (*)$$

<u>Note</u>: Even if the algorithm is stable and not backward stable one can actually show the relative forward errors satisfy the above property in $(*)$

Unstable Algos :-

→ Calculation of eigenvalues of a matrix by finding roots of characteristic polynomial

→ Since λ is eigenvalue of a matrix $\underset{\sim}{A}$, then the characteristic polynomial is $p(\lambda) = \det(\underset{\sim}{A} - \lambda \underset{\sim}{I}) = 0$
roots $p(\lambda) = 0$ are eigenvalues of A

Algo :-

① Find the coefficients of $p(\lambda) = \det(\underset{\sim}{A} - \lambda \underset{=}{I})$

② Find its roots

The problem of finding roots of a polynomial given its coefficients is an ill-conditioned problem!

i.e roots are sensitive to small errors in polynomial coefficients!

Eg:- $\underset{\sim}{A} = \begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix}$ Recall $\underset{\sim}{K} = \dfrac{\|A\|_2}{\lambda} = 1$ for our $\underset{\sim}{A}$

Eigen values of $\underset{\sim}{A}$ are not sensitive to perturbations of the entries. If we design a stable algo, we should be able to compute eigenvalues with relative errors of $O(\epsilon_M)$ [See below theorem]

Let us see the order of error in roots if we compute the roots of characteristic polynomial for the above problem:

$$x^2 - 2ax + a^2 = 0$$

$$\Rightarrow \quad x^2 - (a+a)(1+\varepsilon_1) + a^2(1+\varepsilon_2)$$

$\Rightarrow$ ~~$x^2 - 2x + 1$~~

Let us analyse the error in roots for perturbation in the coefficient p for $x^2 - px + a^2$ (where $p = 2a$)

Roots :- $\dfrac{p \pm \sqrt{p^2 - 4a^2}}{2}$

Let us say my error in estimating the coefficient p be $|\varepsilon| < \varepsilon_M$ (Assume for now a^2 has no error)

$$\tilde{p} = p(1+\varepsilon)$$

New Roots :- $\dfrac{p(1+\varepsilon) \pm \sqrt{[p(1+\varepsilon)]^2 - 4a^2}}{2}$

$$= \frac{p(1+\varepsilon) \pm \sqrt{p^2(1+\varepsilon)^2 - 4a^2}}{2}$$

For our problem at hand $p = 2a$

New roots for $p = 2a$:- ~~$\dfrac{2(1+\varepsilon) \pm \sqrt{4(1+\varepsilon)^2 - 4}}{2}$~~

~~$= (1+\varepsilon) \pm \sqrt{1 + \varepsilon^2 + 2\varepsilon - 1}$~~

~~$\Rightarrow (1+\varepsilon) \pm \sqrt{\varepsilon^2 + 2\varepsilon}$~~

$$\frac{2a(1+\varepsilon) \pm \sqrt{4a^2(1+\varepsilon)^2 - 4a^2}}{2}$$

$$\Rightarrow \frac{2a(1+\varepsilon) \pm \sqrt{\cancel{4a^2} + 4a^2\varepsilon^2 + 8a^2\varepsilon - \cancel{4a^2}}}{2}$$

$$= a(1+\varepsilon) \pm \sqrt{a^2\varepsilon^2 + 2a^2\varepsilon}$$

$$= a(1+\varepsilon) \pm a\sqrt{\varepsilon^2 + 2\varepsilon}$$

Relative

Error in root :- $\dfrac{|(1+\varepsilon) \pm \sqrt{2\varepsilon} - 1|}{|1|}$

Dominant Error in my root $\approx O(\sqrt{\varepsilon_M})$

much higher than $O(\varepsilon_M)$

$\varepsilon_M \simeq 10^{-16}$

$\sqrt{\varepsilon_M} \simeq 10^{-8}$

Not good ✗