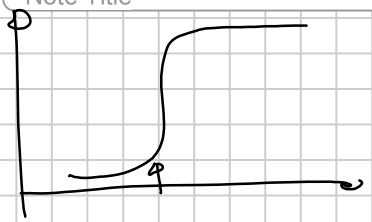
Note Title 2024-11-15



Condition number → good / bad problem

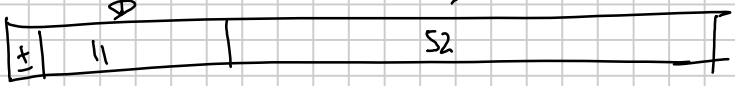
Stability → good / bad algorithm

Floating point representation

periodic

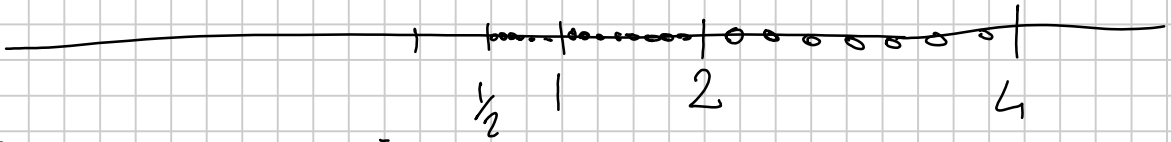
$0.3 \longrightarrow 0.0\underbrace{1001100110011001\ldots}$ truncate! $\longrightarrow$

$= 2^{-2} \cdot 1,\underbrace{00110011001\ldots}_{\text{mantisse}}$



0
±inf
NaN

64 bits



Theorem: For each $x \in \pm[10^{-308}, 10^{308}]$, there is an exactly representable floating point number $\tilde{x}$ such that

$$\frac{|x-\tilde{x}|}{|x|} \leq 2^{-52} \approx 2 \cdot 10^{-16} = u, \text{ "machine precision".}$$

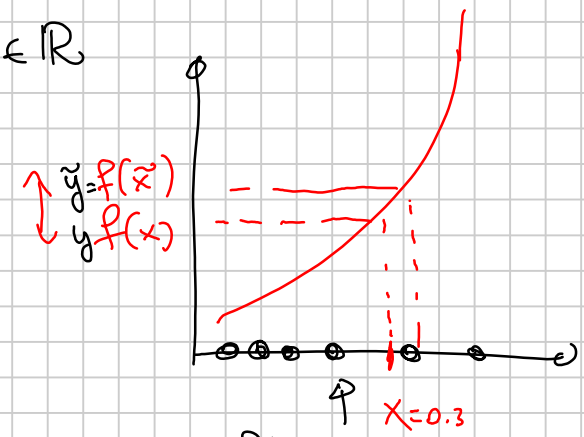
Approximation → inaccuracies

e.g. $\frac{1}{3}+\frac{1}{3}+\frac{1}{3} = 0.333\ldots3 + 0.333\ldots3 + 0.333\ldots3 = 0.999\ldots9 \neq 1$

```
>> a = 0.2   ← error
>> b = 0.3   ← error
>> c = a+b   ← error
```

$y = f(x)$ computational problem

for most possible input $x \in \mathbb{R}$



the distance $\tilde{y} - y$ depends on $f'(x)$

$$\frac{|\tilde{y} - y|}{|y|} \leq K \cdot \frac{|\tilde{x} - x|}{|x|}$$

K: condition number $K_{f,x}$ — $\frac{|\tilde{x} - x|}{|x|} \leq u$

The smallest error bound that I can ensure for all inputs x is $K_{f,x} \cdot u$

($K_{f,x}$: cond. number, u: machine prec)

Intrinsic error

Vector input: $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \to \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{bmatrix}$ $\quad \frac{|\tilde{x}_i - x_i|}{|x_i|} \leq u$ for each i

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq u \cdot O(n)$$

Usually, one writes $O(n)$ to denote $u \cdot$ polynomial in the dimension.

$f(x : \text{double}) : \text{double}$

Def: an algorithm is **stable** (on some input x) if the computed $\tilde{y}$ satisfies $\frac{|\tilde{y} - y|}{|y|} \approx K_{f,x} \cdot O(u)$

Proving stability directly involves keeping track of errors in all operations

$$a \oplus b = \text{floating pt appr.}(a+b)$$
$$= (a+b)(1+\delta) \qquad |\delta| \le u$$

Indeed, the floating pt approx. $\tilde{y}$ of y satisfies

$$\frac{\tilde{y}-y}{y} = \delta \qquad |\delta| \le u$$
$$\Updownarrow$$
$$\tilde{y} = y(1+\delta)$$

Ex: Keep track of the algorithmic error in computing

$$y = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = a_1b_1 + a_2b_2 + a_3b_3$$

Assume a_i, b_i already floating pt numbers

$$\tilde{y} = (a_1 \odot b_1 \oplus a_2 \odot b_2) \oplus a_3 \odot b_3$$

$$= \left(a_1b_1(1+\delta_1) \oplus a_2b_2(1+\delta_2) \right) \oplus a_3b_3(1+\delta_3)$$

$$= \Big(\big(a_1b_1(1+\delta_1) + a_2b_2(1+\delta_2) \big)(1+\delta_4) + a_3b_3(1+\delta_3) \Big)(1+\delta_5) \qquad |\delta_i| \le u$$

$$= \underbrace{a_1b_1 + a_2b_2 + a_3b_3}_{y} + a_1b_1(\delta_1+\delta_4+\delta_5) + a_2b_2(\delta_2+\delta_4+\delta_5) + a_3b_3(\delta_3+\delta_5) + O(u^2)$$

$$|\tilde{y}-y| \le |a_1b_1| \cdot 3u + |a_2b_2| \cdot 3u + |a_3b_3| \cdot 2u$$

$$\leq 3u\left(|a_1||b_1| + |a_2||b_2| + |a_3 b_3|\right) = \begin{bmatrix} |a_1| & |a_2| & |a_3| \end{bmatrix} \begin{bmatrix} |b_1| \\ |b_2| \\ |b_3| \end{bmatrix}$$

In dimension n,

$$|\tilde{y} - y| \leq nu \cdot \underbrace{|a^T| \cdot |b|}_{\text{not } y}$$

$|a^T||b|$ can be much larger than $|y|$, if there is cancellation, e.g.

$$y = \begin{bmatrix} 1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 10^6 \\ 10^6 \\ 1 \end{bmatrix} = 1$$

$$|a^T| \cdot |b| = 2 \cdot 10^6 + 1$$

Is this large error due to an unstable algorithm or an ill-conditioned problem?

Backward stability Idea: sometimes, you can see the exactly computed $\tilde{y}$ as the exact result of a computation with perturbed inputs

$$\tilde{y}(a, b) = y(\hat{a}, \hat{b})$$

$$\tilde{y} = a_1 b_1 \underbrace{(1+\delta_1+\delta_4+\delta_5)}_{\hat{b}_1} + a_2 b_2 \underbrace{(1+\delta_2+\delta_4+\delta_5)}_{\hat{b}_2} + a_3 b_3 \underbrace{(1+\delta_3+\delta_5)}_{\hat{b}_3} + O(u^2)$$

$$= a_1 \hat{b}_1 + a_2 \hat{b}_2 + a_3 \hat{b}_3$$

$$\left|\frac{\hat{b}_1 - b_1}{b_1}\right| = |\delta_1 + \delta_4 + \delta_5| \leq 3|u| \qquad \frac{\|\hat{b} - b\|}{\|b\|} \leq 3u + O(u^2) = O(u)$$

We can use condition number bounds:

$$\frac{|\tilde{y} - y|}{|y|} \leq \kappa_{y,b} \cdot \frac{\|\hat{b} - b\|}{\|b\|} = \kappa_{y,b} \cdot O(u)$$

of the same order of magnitude as the intrinsic error $\kappa_{y,b} \cdot u$

$\Rightarrow$ we can conclude that this algorithm is stable!

Def: An algorithm is called <u>backward stable</u> (on input x) if the result $\tilde{y}(x)$ it computes in machine arithmetic can be seen as the <u>exact</u> result of a slightly perturbed input, $y(\hat{x})$, $\frac{\|\hat{x} - x\|}{\|x\|} = O(u)$

Backward stability $\Rightarrow$ stability!

⚠ We can't apply this trick to every algorithm!

Ex: outer product $Y = ab^T = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} [b_1 \; b_2 \; b_3]$

$$= \begin{bmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix}$$

$$\tilde{Y} = \begin{bmatrix} a_1 \odot b_1 & a_1 \odot b_2 & a_1 \odot b_3 \\ \vdots & \vdots & \vdots \\ - & - & a_3 \odot b_3 \end{bmatrix} \neq \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \end{bmatrix} [\hat{b}_1 \; \hat{b}_2 \; \hat{b}_3],$$

in general, because the errors do not always make that matrix rank 1

Ex: $Y = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} [4 \;\; 5 \;\; 6] = \begin{bmatrix} 4 & 5 & 6 \\ 8 & 10 & 12 \\ 12 & 15 & 18 \end{bmatrix}$

$$\tilde{Y} = \begin{bmatrix} 3.99 & 4.01 & 5.99 \\ 7.99 & 9.99 & 12.01 \\ 12.01 & 14.99 & 18.02 \end{bmatrix}$$

↑ ↑ ↑

in general, the columns will not be multiples of the same vector

⇒ we cannot write it as $\tilde{Y} = \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \end{bmatrix} [\hat{b}_1 \;\; \hat{b}_2 \;\; \hat{b}_3]$

"A posteriori" backward stability

Ex: linear system $Ax = y$, $A \in \mathbb{R}^{n \times n}$

Given a computed solution $\tilde{x}$, let us consider the residual $\tilde{r} = A\tilde{x} - y$

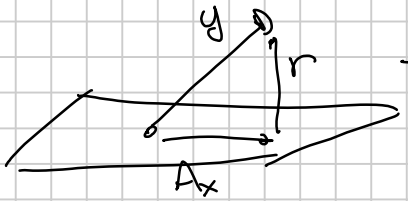
Note that $A\tilde{x} = y + \tilde{r}$ ⇒ $\tilde{x}$ is the exact solution of the linear system $A\tilde{x} = \hat{y}$ with $\hat{y} = y + \tilde{r}$

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq k(A) \frac{\|\hat{y} - y\|}{\|y\|} = k(A) \boxed{\frac{\|\tilde{r}\|}{\|y\|}}$$

(we can usually ignore the fact that $\tilde{r}$ and the other quantities are computed in machine arithmetic, their order of magnitude is still reliable).

Can we do it also for least-squares problems?

$$\min_{x \in \mathbb{R}^n} \|Ax - y\|_2^2$$



⚠ r is usually not small!

However, the gradient $g(\tilde{x}) = A^TA\tilde{x} - A^Ty$ is zero in the exact solution; we can use the previous a-posteriori bound on the normal equations:

$$\frac{\|\tilde{x}-x\|}{\|x\|} = \underbrace{\kappa(A^TA)}_{\uparrow} \frac{\|A^TA\tilde{x}-A^Ty\|}{\|A^Ty\|}.$$

⚠ $\kappa(A)^2$ larger than the cond. number of our problem, if $\theta \approx 0°$

$\Rightarrow$ even if $\frac{\|A^TA\tilde{x}-A^Ty\|}{\|A^Ty\|} = O(u)$,

this does not mean that the algorithm is stable.

Obs: small residual usually come from algorithms that are backward stable.

Ex: $Ax = y$, $A \in \mathbb{R}^{n\times n}$ $\quad \hat{A}\tilde{x} = \hat{y}$

$\hat{A} = A+E$, $\frac{\|E\|}{\|A\|} = O(u)$ $\quad \hat{y} = y+f$ $\quad \frac{\|f\|}{\|y\|} = O(u)$

$$(A+E)\tilde{x} = y+f \Rightarrow \|A\tilde{x}-y\| = \|f-E\tilde{x}\| \le \|f\| + \|E\|\cdot\|\tilde{x}\|$$

$$= O(u)\left(\|y\| + \|A\|\cdot\|\tilde{x}\|\right)$$

$\Rightarrow$ the residual $\|A\tilde{x}-y\|$ is always small with respect not

to $\|y\|$ but to $\|y\| + \|A\| \cdot \|\tilde{x}\|$ $\qquad$ $\|y\| = \|Ax\| \leq \|A\| \cdot \|x\|$

Variant of the residual test for least-squares problems:

x solves $\min_x \|Ax - y\|$ with $r = Ax - y$

$$\Leftrightarrow \begin{bmatrix} I_m & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} -r \\ x \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix} \Leftrightarrow \begin{cases} -r + Ax = y \\ -A^T r = 0 \end{cases} \Leftrightarrow A^T(Ax - y) = 0$$

$(m+n) \times (m+n)$

This can be used for a residual test:

given $\tilde{x}$ with residual $\tilde{r}$

$$\frac{\left\| \begin{bmatrix} -\tilde{r} \\ \tilde{x} \end{bmatrix} - \begin{bmatrix} -r \\ x \end{bmatrix} \right\|}{\left\| \begin{bmatrix} -r \\ x \end{bmatrix} \right\|} \leq \kappa\left(\begin{bmatrix} I_m & A \\ A^T & 0 \end{bmatrix} \right) \frac{\left\| \begin{bmatrix} 0 \\ -A^T \tilde{r} \end{bmatrix} \right\|}{\left\| \begin{bmatrix} y \\ 0 \end{bmatrix} \right\|}$$

$$\Rightarrow \frac{\|\tilde{x} - x\|}{\left\| \begin{bmatrix} -r \\ x \end{bmatrix} \right\|^2} \leq \kappa\left(\begin{bmatrix} \alpha I & A \\ A^T & 0 \end{bmatrix} \right) \cdot \frac{\frac{1}{\alpha} \|A^T \tilde{r}\|}{\|y\|}$$

"A posteriori" stability tests, using the computed result.