Fitting m points with an $m-1$ degree polynomial is numerically unstable, for m large. Instead, use LSQ and a low order polynomial:

Start with $\{x_i\}_{i=1}^{m}$ $\{y_i\}_{i=1}^{m}$

but use instead a low-order polynomial

$$p(x) = c_0 + c_1 x + \cdots c_{n-1} x^{n-1} \qquad n << m.$$

Let $r_i = y_i - p(x_i)$,

$$\underset{\sim}{r} = \begin{Bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{Bmatrix} \qquad \underset{\sim}{c} = \begin{bmatrix} c_0 \\ \vdots \\ c_{n-1} \end{bmatrix}$$

$$\left.\begin{array}{l} A \in \mathbb{R}^{m \times n} \\ \text{As before } A^* r = 0 \end{array}\right\} \; Ac - Y = r$$

$$A = \underbrace{\begin{bmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ 1 & x_2 & & \vdots \\ & \vdots & & \vdots \\ 1 & x_m & & x_m^{n-1} \end{bmatrix}}_{n} \Bigg\} m \qquad Y = \begin{bmatrix} y_1 \\ \vdots \\ \vdots \\ y_m \end{bmatrix}$$

Minimize $\|r\|^2$ found via LSQ

$r \perp col(A)$
(range(A))

$\min_x \|Y - Ac\|_2 = \hat{x}$

the minimizer $\hat{x}$

is such that $A\hat{x} = PY$

P is projection of Y onto

range of A.

$$A^* A \hat{x} = A^* Y$$

or $PY = A\hat{x}$

Rmk: $\hat{x}$ is unique iff A is fullrank.

This follows from:

if $A^* A$ is singular then

$A^* A y = 0$ for some $y \neq \emptyset$

implying $y^* A^* A y = 0$

$\therefore \; Ay = 0 \Rightarrow A$ is rank deficient.

Conversely if $Ay = 0$ and $y = 0$

then $y^* A^* A y = 0$

implies that $A^* A$ is full rank.

The PseudoInverse of $A \equiv A^+$

A^+ contains whatever part of A is invertible, i.e. the row space of A): It knocks out the null(A^*), by sending it to $\emptyset$, and knocks out null(A) by choosing $x^+ = x_r$ in the row space of A: Hence,

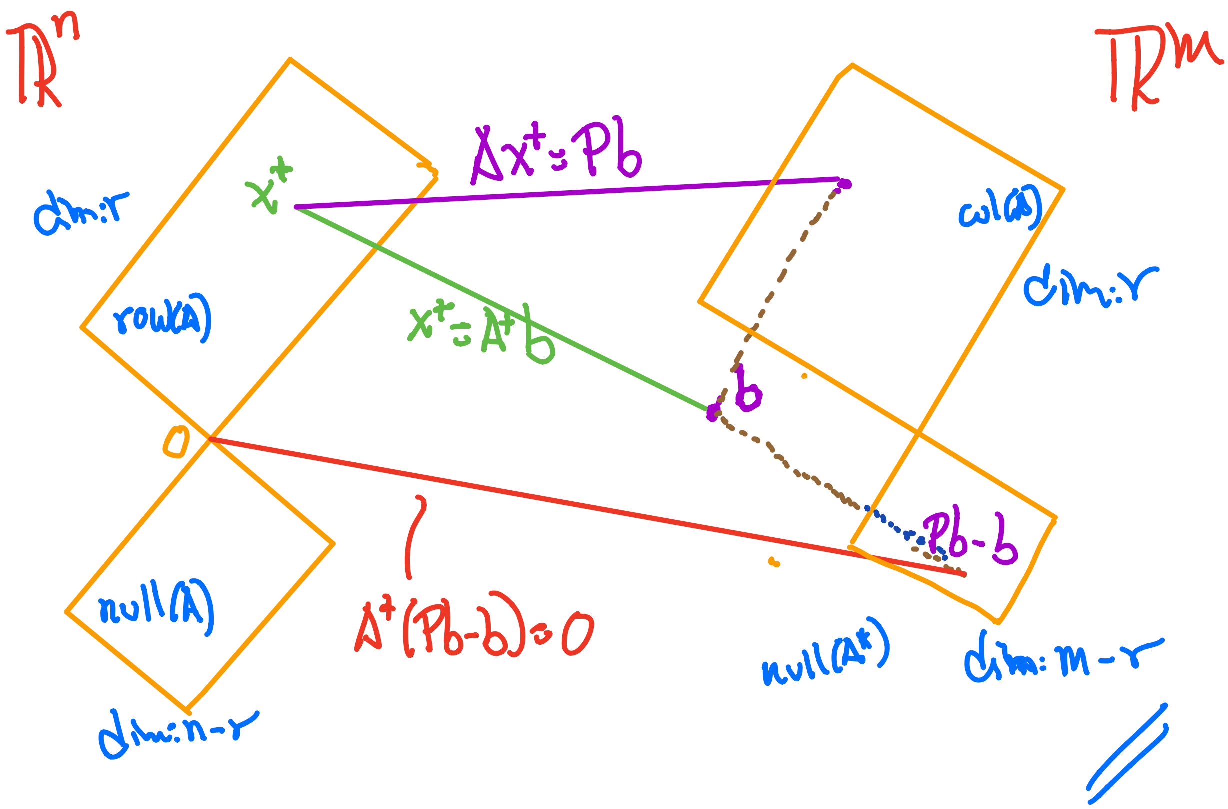
For $Ax = b$ $\quad b \in \mathbb{C}^m$

$A \in \mathbb{C}^{m \times n}$

$$x^+ = A^+ b \qquad A^+ \in \mathbb{C}^{n \times m}$$

(†)

$$\boxed{A^+ = (A^* A)^{-1} A^*}$$



ALGORITHMS FOR LSQ

(I) **NORMAL EQUATIONS** (especially good when A is Hermitian, positive definite matrix, square): the decomposition is unique.

Cholesky Factorization. Use (†)

(II) **QR Factorization**

$$A = \hat{Q}\hat{R}$$

so $P = \hat{Q}\hat{Q}^*$

$$y = Pb = \hat{Q}\hat{Q}^* b \in \text{range}(A)$$

$$\underbrace{\hat{Q}\hat{R}}_{A}\,\hat{x} = \underbrace{\hat{Q}\hat{Q}^*}_{P}\, b$$

multiply from the left by $\hat{Q}^*$

$$\hat{R}\hat{x} = \hat{Q}^* b$$ (if full rank solve by back substitution)

multiply by $\hat{R}^{-1}$

$$\hat{x} = \hat{R}^{-1}\hat{Q}^* b = A^+ b$$

to see this:

$$\hat{R}^{-1}\hat{Q}^* = (A^*A)^{-1}A^* = \left[(\hat{Q}\hat{R})^*\hat{Q}\hat{R}\right]^{-1}(\hat{Q}\hat{R})^*$$

$$= \left[\hat{R}^*\hat{Q}^*\hat{Q}\hat{R}\right]^{-1}\hat{R}^*\hat{Q}^*$$

$$= \left[R^*R\right]^{-1}\hat{R}^*\hat{Q}^* = R^{-1}\hat{Q}^* \quad //$$

ALGORITHM

Cost: $O(mn^2, n^3)$ due QR

1. $A = \hat{Q}\hat{R}$
2. $Q^* b$
3. $\hat{R}\hat{x} = Q^* b$ for $\hat{x}$ //

III SVD:

$$A = \hat{U}\hat{\Sigma}V^*$$

$$\underbrace{y}_{m} = Pb = \underbrace{\hat{U}\hat{U}^*}_{m\times m} \underbrace{b}_{m}$$

$$\hat{U}\hat{\Sigma}V^*\hat{x} = \hat{U}\hat{U}^* b$$

$$\Rightarrow \hat{\Sigma}V^*\hat{x} = \hat{U}^* b$$

multiply both sides by $V\Sigma^{-1}$

$$\boxed{\begin{aligned} \hat{x} &= A^+ b \\ A^+ &= \hat{V}\Sigma^{-1}\hat{U}^* \end{aligned}}$$

ALGORITHM: Cost $O(mn^2, n^3)$

1. SVD $A = \hat{U}\hat{\Sigma}V^*$
2. Compute $U^* b$.
3. $\hat{\Sigma}w = U^* b$ solve for w.
4. $\hat{x} = Vw$ to get $\hat{x}$.

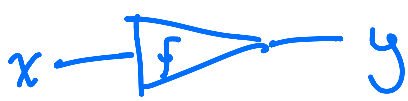
Most stable of the 3 algorithms
(especially useful when rank deficient)

PART III CONDITIONING

CONDITIONING: mathematical problem
STABILITY: algorithm used to approximate mathematical problem

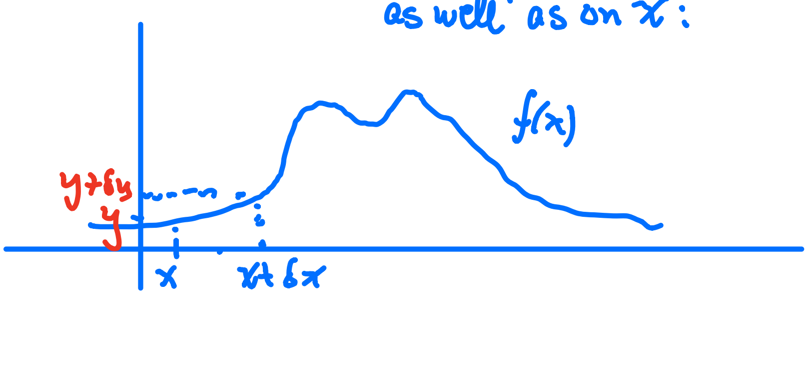
Model: input/output problem

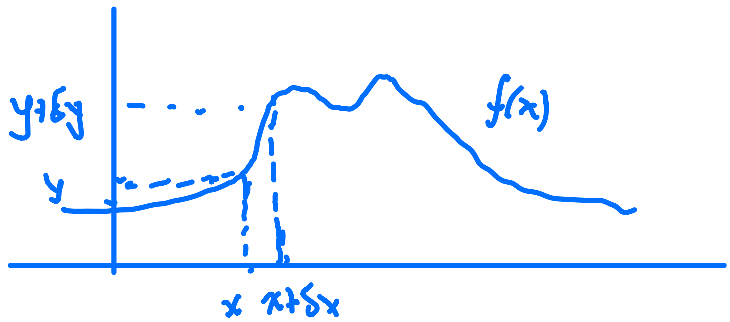
$x \in \mathbb{C}^m$ — input $\qquad y = f(x)$ — output



Make a small change $x \to x + \delta x$ of the input $|\delta x| \ll 1$ want to estimate how large a fluctuation we get in the output?
$y \to y + \delta y$?

It depends on $f(x)$ as well as on x:





For $f(x)$ continuous

$$y+\delta y = f(x+\delta x) \approx f(x) + \overbrace{\frac{\partial f}{\partial x}}^{\text{Jacobian Matrix}}\Bigg|_x \delta x + \frac{1}{2}\, \delta x^T \underbrace{\frac{\partial^2 f}{\partial x^2}}_{\text{Hessian}} \delta x \Bigg|_x + \cdots$$

Take $\|\delta x\| \ll 1$ so that

$$y + \delta y \approx f(x) + J(x)\,\delta x$$

$J(x)$ is the Jacobian

ex) $$y = \begin{pmatrix} \cos x_1 \\ x_1 e^{x_2} \\ x_3 \end{pmatrix} = f(x) = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}$$

$$J(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{pmatrix}$$

$$= \begin{pmatrix} -\sin x_1 & 0 & 0 \\ e^{x_2} & x_1 e^{x_2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$J\left(x = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}\right) = \begin{pmatrix} 0 & 0 & 0 \\ e^1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$y + \delta y \approx f(x) + J(x)\,\delta x$$

since $y = f(x)$

$$\therefore \ \delta y \approx J(x)\,\delta x$$

Want to relate $\frac{\delta y}{y}$ to $\frac{\delta x}{x}$

$$\frac{\delta y}{y} \approx \frac{x J(x)\,\delta x}{x f(x)}$$

THE RELATIVE RATES OF INPUT TO OUTPUT:

$$\left\| \frac{\delta y}{y} \right\| \equiv k \left\| \frac{\delta x}{x} \right\|$$

$$\boxed{k = \frac{\|x\| \, \|J(x)\|}{\|f(x)\|}}$$ "CONDITION NUMBER"

if $k \gg 1$ ill-conditioned

$k < 1$ well-conditioned