Lecture 2 Subspace, range and kernel

- **subspace**: $G \subset \mathbb{C}^n$

  $\mathbb{C}^n$ is complex linear space

  G is subset of $\mathbb{C}^n$, which is a vector space

- **linear independance**

  $G = \{a_1, a_2, \dots a_k\} \subset \mathbb{C}^n$ is lin. ind. $\Leftrightarrow$

  $\alpha_1 a_1 + \alpha_2 a_2 + \dots + \alpha_k a_k = 0 \Rightarrow \alpha_1 = \alpha_2 = \dots \alpha_k = 0$

- **linear span**

  $span(G) = \{ z \in \mathbb{C}^n \mid z = \sum_{i=1}^{k} \alpha_i a_i \}$

  $\alpha_i \in \mathbb{C}$, $\{a_i\}_{i=1}^{k} \in \mathbb{C}^n$

- **basis**

  G is basis of $span(G)$, if G is lin. ind.

- **direct sum**

  $S = S_1 \oplus S_2$, if

  1) $\forall v \in S \quad v = a_1 + a_2, \quad a_1 \in S_1, \; a_2 \in S_2$

  2) $S_1 \cap S_2 = \{0\}$, otherwise the sum is not direct
     intersection

  If $S = \mathbb{C}^n \Rightarrow \forall x \in S \quad \exists!\, x_1 \in S_1, x_2 \in S_2:$

  $x = x_1 + x_2$

  $\exists!$ means exists and unique

2)

Ⓓ <u>Range of A</u> область значений

$Ran(A) = \{Ax \mid x \in \mathbb{C}^m\}$, $A \in \mathbb{C}^{n\times m}$

$Ran(A) \subset \mathbb{C}^n$

Ⓓ <u>Kernel of A</u> ядро

$Null(A) = \{x \in \mathbb{C}^m \mid Ax = 0\}$, $A \in \mathbb{C}^{n\times m}$

$Null(A) \subset \mathbb{C}^m$

P1 $Ran(A) = span\{a_1, a_2, \dots a_m\}$

P2 $\dim(Null(A)) + \dim(Ran(A)) = n$

Ⓓ <u>rank of A</u>

A rank is a number of lin. ind. columns

$rank(A) = \dim(Ran(A))$

Column rank is equal to row rank

Ⓓ <u>matrix of full rank</u>

$rank(A) = \min\{m, n\}$, $A \in \mathbb{C}^{n\times m}$

P
$$\begin{Vmatrix} \mathbb{C}^n = Ran(A) + Null(A^T) \\ \mathbb{C}^n = Ran(A^T) + Null(A) \end{Vmatrix}$$

Ⓓ <u>invariant subspace of A</u>

If $AS \subset S$, then S is called invariant subspace of A

P $Null(A - \lambda I)$ is invariant subspace of A

Ⓓ <u>eigenspace</u>

$Null(A - \lambda I)$ is called eigenspace of A

Existence of solution

Consider a linear system

$Ax=b$, $A\in\mathbb{C}^{n\times n}$, $b\in\mathbb{C}^n$

$x\in\mathbb{C}^n$ is a vector of unknowns

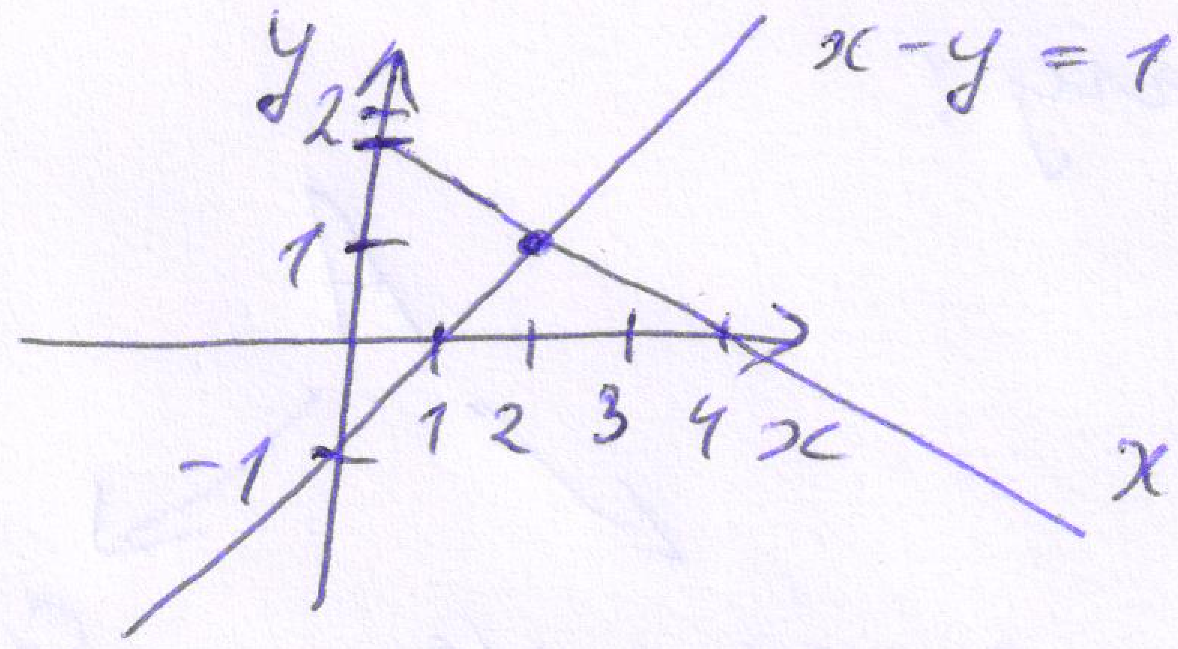
There are three cases:

1) unique solution

A is nonsingular ($\det A\neq 0$, $\exists A^{-1}$)

$\exists!\ x=A^{-1}b$ is solution

Ex. $\begin{cases} x-y=1 \\ x+2y=4 \end{cases}$ $x=2;\ y=1$ is solution



2) infinitely many solutions

A is singular, $b\in \mathrm{Range}(A)$

$\mathrm{Ran}(A)\overset{\text{def}}{=}\{Au \mid u\in\mathbb{C}^n\}$

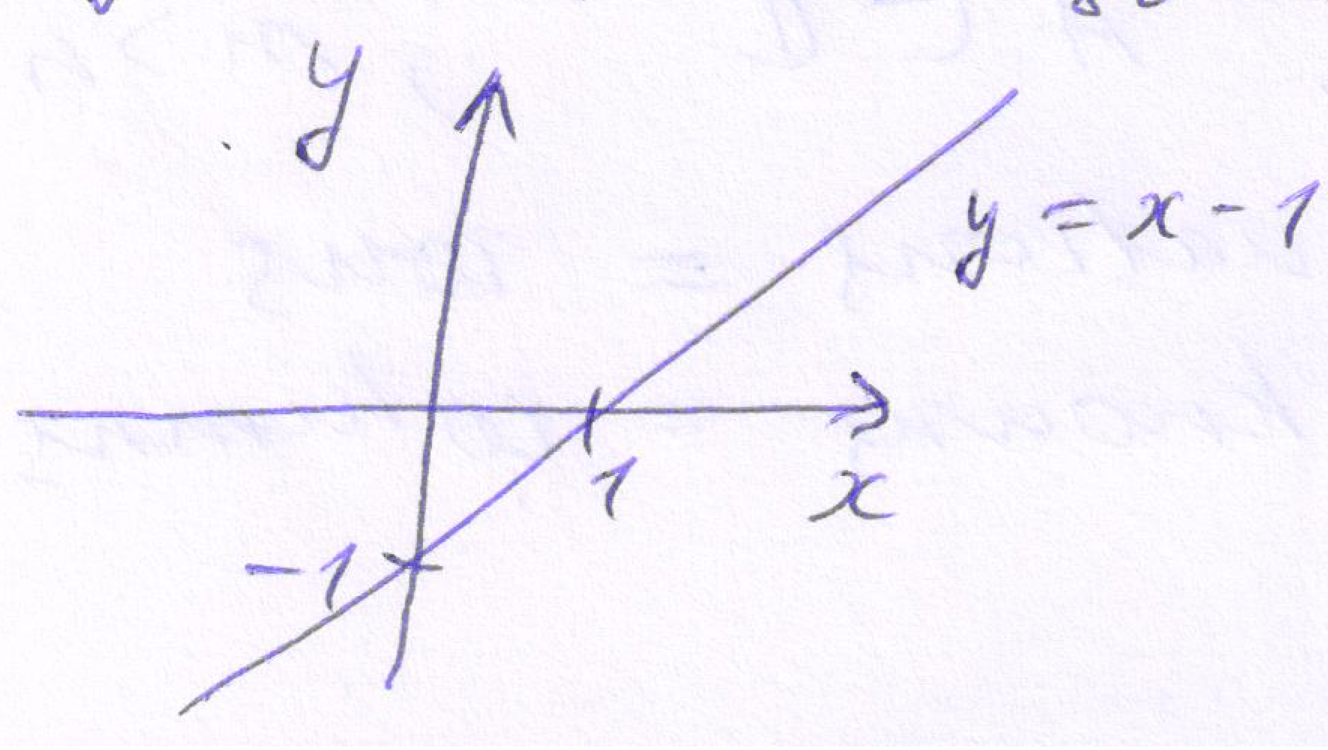
$\exists x_0 : Ax_0=b \Rightarrow x_0+v$ is also solution

$\forall v\in \mathrm{Null}(A)\overset{\text{def}}{=}\{u \mid Au=0\}$

$\dim(\mathrm{Null}(A))\geq 1$, at least 1D

Ex. $\begin{cases} x-y=1 \\ -2x+2y=-2 \end{cases}$ $y=x-1$ is solution

This system is consistent

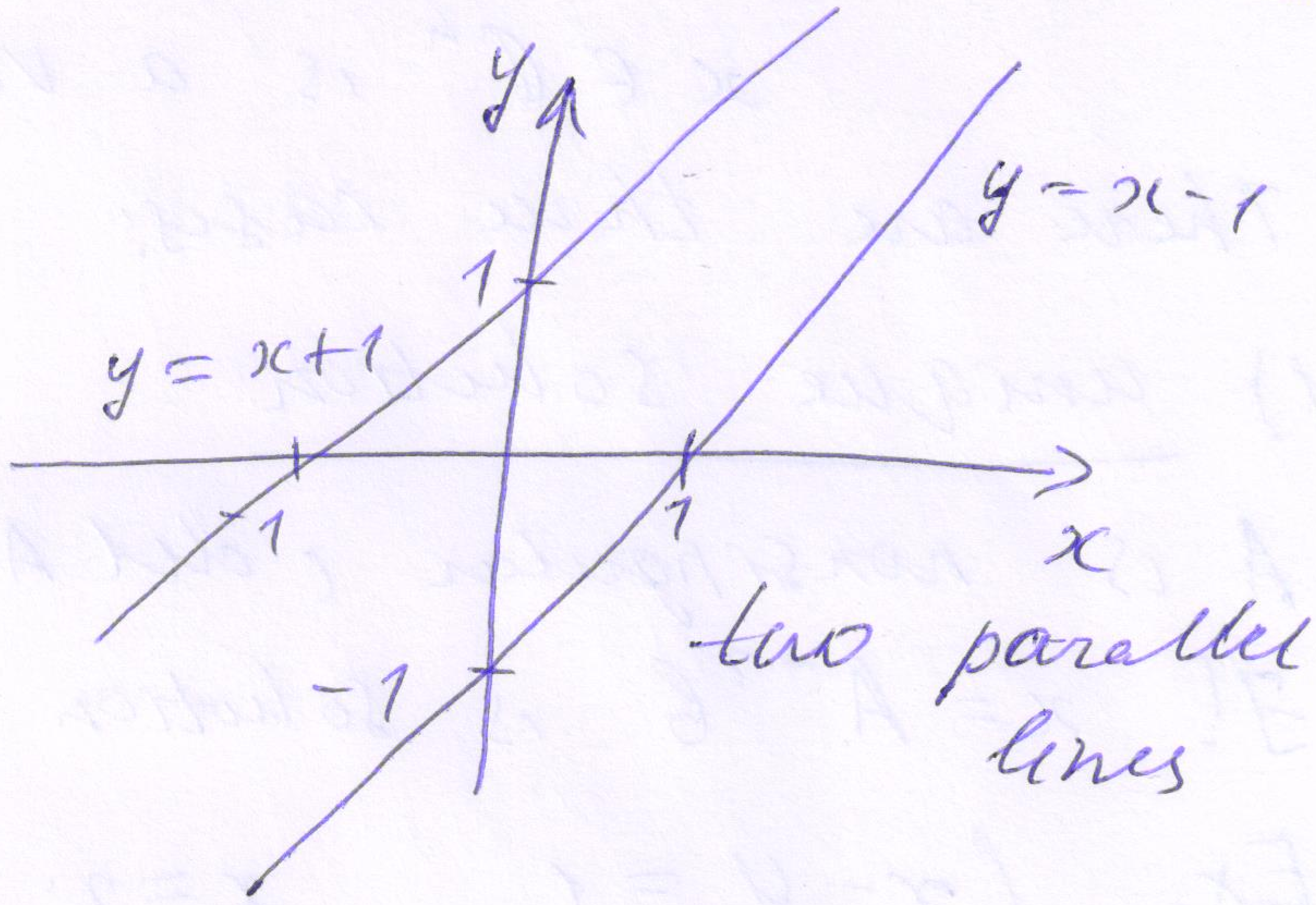


3) <u>no solutions (inconsistent system)</u>

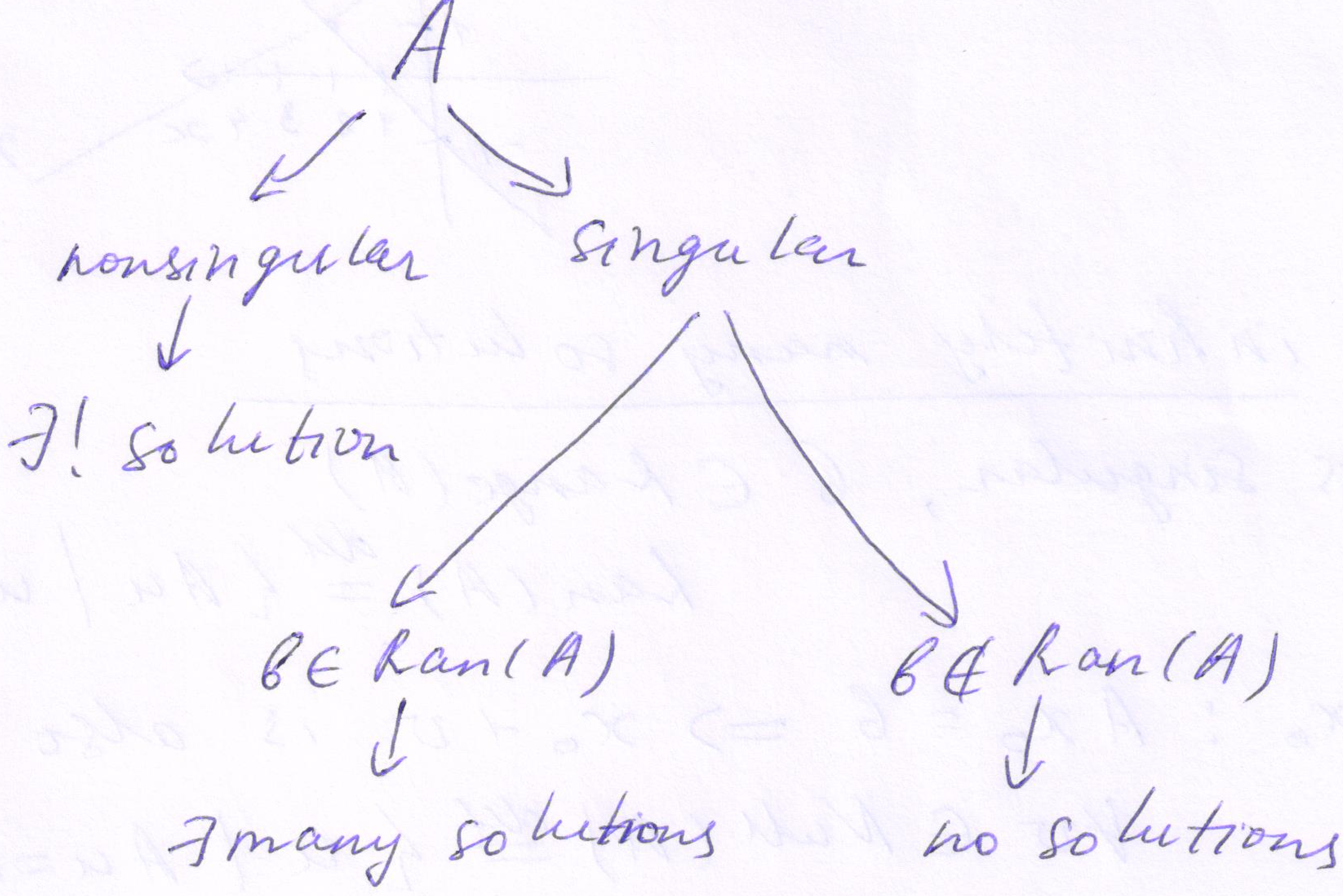
несовместная

A is singular ; $b \notin Ran(A) \Rightarrow \nexists$ solutions

Ex. $\begin{cases} x - y = 1 \\ x - y = -1 \end{cases}$



Summary:



Consider a linear system with rectangular matrix:

$Ax = b, \quad A \in \mathbb{C}^{m \times n}, \ m > n ; \ x, b \in \mathbb{C}^n$

m equations = rows

n unknowns = columns

When $m > n$, the system is overdefined

When $m < n$, the system is underdefined

For $m > n$ we can find solution of a least squares problem:

find $x \in \mathbb{C}^n$: $\|b - Ax\|_2 \to \min$

$b - Ax = r$ is residual (невязка)

Orthogonality of vectors and subspaces

① orthogonal vectors

$G = \{a_1, a_2, \ldots, a_k\}$

Set G is orthogonal, if $(a_i, a_j) = 0 \ \forall i \neq j$

② orthonormal vectors

$$(a_i, a_j) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

$\|a_i\|_2 = 1$, $\|a_i\|_2 \overset{\text{def}}{=} \sqrt{(a_i, a_i)}$ Euclidian vector norm

③ orthogonality to a subspace

$x \perp s \ \ \forall s \in S \Rightarrow x \perp S$

vector s from subspace S

④ orthogonal complement

$S^\perp = \{x \mid x \perp s \ \ \forall s \in S\}$

$\mathbb{C}^n = S \oplus S^\perp$

6)

⊖ orthogonal projector

mapping $S' \to S^{\perp}$

отображение

⊖ normalization of a vector

$$\frac{x}{\|x\|_2} = q, \quad \|q\|_2 = 1$$

Gram-Schmidt process

This is a process of orthonormalization of a set of vectors

Consider a set of vectors
$\{x_1, x_2, \ldots x_k\} \in \mathbb{C}^n$

Build an orthonormal set

$\{q_1, q_2, \ldots, q_k\} \in \mathbb{C}^n$, where $\{q_i, q_j\} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$

Note: $(x, y) = y^H x$ therefore in Matlab we can compute inner product as $y' * x$

When the set $\{x_i\}_{i=1}^{k}$ is lin. ind., then orthonormal set will contain the same number of vectors $\{q_i\}_{i=1}^{k}$

If $\{x_i\}_{i=1}^{k}$ contains linear dependent vectors, then orthonormal set $\{q_i\}_{i=1}^{p}$, $p < k$

Get formulas to compute q_j:

$$\begin{cases} x_1 = r_{11} q_1 \\ x_2 = r_{12} q_1 + r_{22} q_2 \\ \vdots \\ x_k = r_{1k} q_1 + r_{2k} q_2 + \dots + r_{kk} q_k \end{cases}$$

Form matrices

$$X = \begin{pmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_k \\ | & | & & | \end{pmatrix}_{n \times k}$$

$$Q = \begin{pmatrix} | & | & & | \\ q_1 & q_2 & \cdots & q_k \\ | & | & & | \end{pmatrix}_{n \times k} \quad \text{is unitary}$$

$$R = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1k} \\ 0 & r_{22} & & r_{2k} \\ \vdots & & & \\ 0 & \cdots & \cdots & r_{kk} \end{pmatrix}_{k \times k}$$

$R = (\ 0 \ \ )$ upper triangular

$$\begin{matrix} | \\ r_2 \\ | \end{matrix}$$

$$r_{ij} = (x_j, q_i)$$

G-S process:

1) $x_1 \Rightarrow q_1 = \dfrac{x_1}{\|x_1\|_2}$

2) $x_2 \Rightarrow x_2 := x_2 - (x_2, q_1) q_1$

$$q_2 = \frac{x_2}{\|x_2\|_2}$$

i) $x_i \Rightarrow x_i := x_i - \overbrace{(x_i, q_{i-1})}^{r_{i-1,i}} q_{i-1}$

$q_i = x_i / \|x_i\|_2$

8)

Classic G-S algorithm

1. $r_{11} := \|x_1\|_2$

   $q_1 = x_1 / r_{11}$

2. Loop for $j = \overline{2, k}$

   2.1) $r_{ij} = (x_j, q_i)$, $i = \overline{1, j-1}$

   2.2) $\hat{q} = x_j - \sum_{i=1}^{j-1} r_{ij} q_i$

   $\hat{q}$ is auxilary vector orthogonal to all previous vectors

   2.3) $r_{jj} = \|\hat{q}\|_2$. If $r_{jj} = 0 \Rightarrow$ exit

   2.4) $q_j = \hat{q} / r_{jj}$

Note that in this algorithm all diagonal entries of R r_{jj} are computed as norms $\Rightarrow$ diagonal entries $r_{jj} \geq 0$

For actual programming use

Modified G-S algorithm

1. $r_{11} := \|x_1\|_2$, $q_1 = x_1 / r_{11}$

2. Loop for $j = \overline{2, k}$

   2.1) $\hat{q} := x_j$

   2.2) Loop for $i = \overline{1, j-1}$

   $r_{ij} = (\hat{q}, q_i)$

   $\hat{q} = \hat{q} - r_{ij} q_i$

   2.3) $r_{jj} = \|\hat{q}\|_2$

   2.4) $q_j = \hat{q} / r_{jj}$

In modified G-S the sum is computed in iterative process.
Modified G-S is more stable than classic G-S when vectors $\{x_i\}$ are almost linear dependent

QR-factorization

$$\underset{m\times n}{A} = \underset{m\times n}{Q}\,\underset{n\times n}{R}, \quad A\in\mathbb{C}^{m\times n}, \quad m\geq n$$

G-S process applied to ~~a~~ the columns of A gives a thin (reduced) QR-factorization

$$A = \begin{pmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & & | \end{pmatrix}, \quad a_j\in\mathbb{C}^m$$

T1 $\forall A\in\mathbb{C}^{m\times n}$ $\exists$ thin QR-factorization $\underset{m\times n}{A} = \underset{m\times n}{Q}\,\underset{n\times n}{R}$ $\iff$ column of A are lin. ind. $\iff$ $\operatorname{rank}(A)=n$, i.e. A is a matrix of full rank

Note that in general thin QR is not unique.

T2 $\forall A\in\mathbb{C}^{m\times n}$ ~~thin~~ thin QR-factorization is unique ($\exists!\,R$, $\exists!\,Q$) when all diagonal entries of R are positive (or all are negative)

$r_{ii}>0$

$(r_{ii}<0)$

10) If we don't choose r_{ii} to be positive, then thin QR is not unique

For example

$$A = \underbrace{Q_1 R_1}_{r_{ii} > 0} = \underbrace{Q_2 R_2}_{r_{ii} \gtrless 0}$$

$r_{ii} > 0$ (your function in PA2)

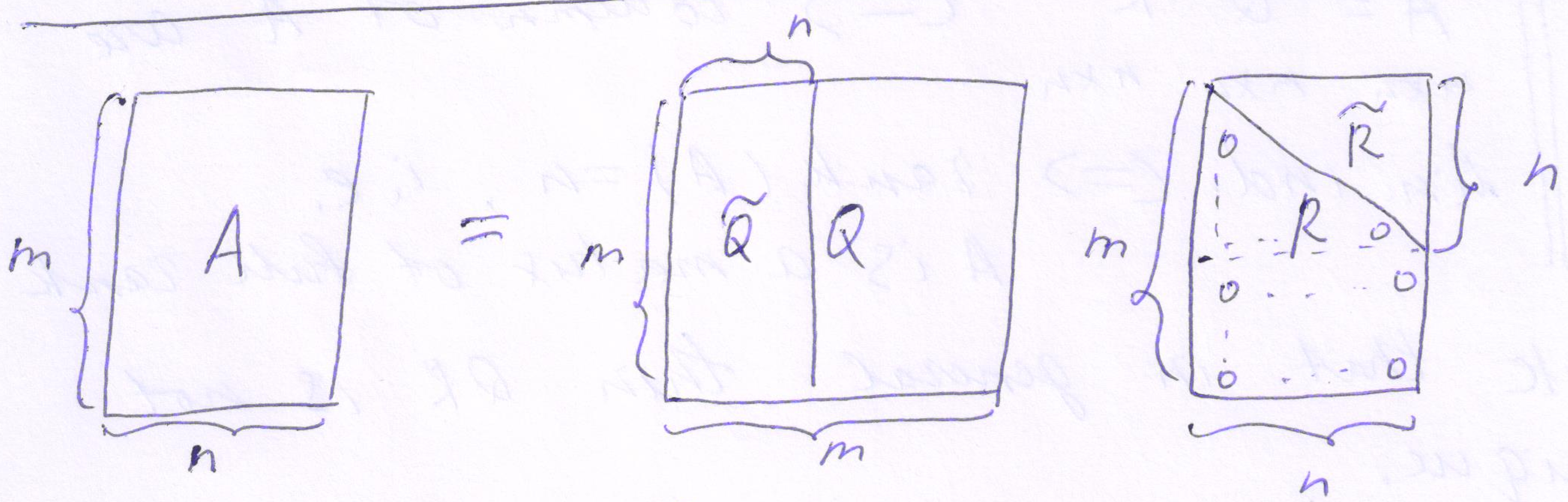
$r_{ii} \begin{smallmatrix} >0 \\ <0 \end{smallmatrix}$ (Matlab algorithm)

$\boxed{\text{Matlab thin QR: } qr(A, 0)}$

Full QR-factorization

(T) $\forall A \in \mathbb{C}^{m \times n},\ m \geq n$
$\exists$ full QR-factorization: $\underset{m\times n}{A} = \underset{m \times m}{Q}\ \underset{m \times n}{R}$

Relation between thin and full QR



QR

full: $\underset{m\times n}{A} = \underset{m\times m}{Q}\ \underset{m\times n}{R}$

thin: $\underset{m\times n}{A} = \underset{m\times n}{\tilde{Q}}\ \underset{n\times n}{\tilde{R}}$

$\tilde{Q}$ is Q with the last $m-n$ columns skipped

$\tilde{R}$ is R with the last $m-n$ rows skipped

Any matrix factorization can be used either to solve or analyze some problem

Example Use QR-factorization to solve
$$Ax = b$$

If $Ax = b \Rightarrow x = A^{-1}b$ (formal writing, we will never compute A^{-1} in numerical algorithms.

$A = QR \Rightarrow x = (QR)^{-1}b = R^{-1}Q^{-1}b =$
$= R^{-1}Q^{H}b$ ($Q^{-1} = Q^{H}$, as Q is unitary)

Hence, $x = R^{-1}Q^{H}b$

x is solution of $Rx = Q^{H}b$

R is upper triangular, this system is faster to solve.

Matlab: `x = R \ (Q' * b)`

Ways to solve $Ax = b$ in Matlab

1) `x = A \ b`
   `x = mldivide (A, b)`

See PA 3-3
PA 3-2

2) `x = linsolve (A, b)`

3) `x = inv(A) * b`