Numerical Methods for Computational Science and Engineering

Fall Semester 2017 (HS17)

Prof. Rima Alaifari, SAM, ETH Zurich

Note on stability: $c_{A^TA} = c_A^2$ $\quad c_A = \text{cond}(A)$

↑ condition number squares

Example (3.2.4): $A = \begin{bmatrix} 1 & 1 \\ \delta & 0 \\ 0 & \delta \end{bmatrix}$ $\quad A^TA = \begin{bmatrix} 1+\delta^2 & 1 \\ 1 & 1+\delta^2 \end{bmatrix}$

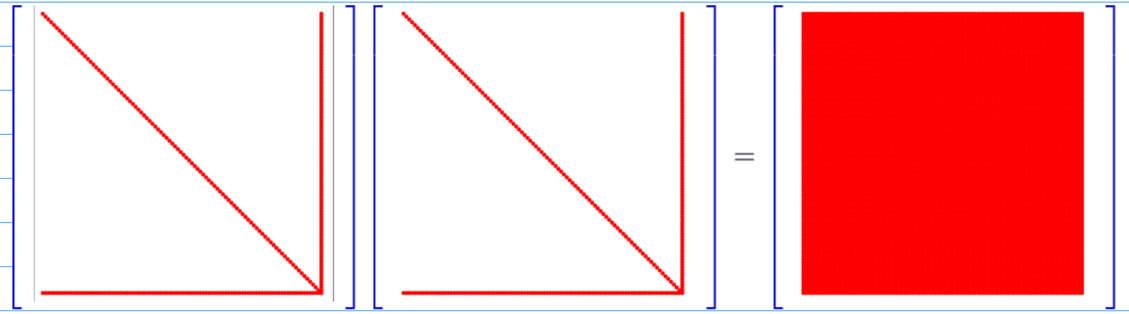
If $\delta \approx \sqrt{EPS}$ e.g. $\delta = \frac{\sqrt{EPS}}{2}$: $1+\delta^2 \approx 1 + \frac{EPS}{4} = 1$

↑ in machine number arithmetic

As an element of $M^{2,2}$, A^TA is not regular.

Further note:

A sparse $\not\Rightarrow$ A^TA sparse



arrow matrix A: sparse ✓ A^TA dense

① Squaring condition number
② Loss of sparsity

} challenges

Extended normal equation: (3.2.7) ← maintain sparsity

$$r := Ax - b$$

$$\mathbf{A}^H\mathbf{A}\mathbf{x} = \mathbf{A}^H\mathbf{b} \quad \Leftrightarrow \quad \mathbf{B}\begin{bmatrix}\mathbf{r}\\ \mathbf{x}\end{bmatrix} := \underbrace{\begin{bmatrix}-\mathbf{I} & \mathbf{A}\\ \mathbf{A}^H & 0\end{bmatrix}}_{\mathbf{B}}\begin{bmatrix}\mathbf{r}\\ \mathbf{x}\end{bmatrix} = \begin{bmatrix}\mathbf{b}\\ 0\end{bmatrix}. \qquad (= Ax-b)$$

$$-r + Ax = b \qquad (I)$$

$$A^H r = A^H(Ax-b) = 0 \qquad (II)$$

if A is sparse $\Rightarrow$ B is also sparse

BUT: conditioning is not improved

More general form: $r := \alpha^{-1}(Ax-b)$

for some choice of parameter $\alpha > 0$

$$\mathbf{A}^H\mathbf{A}\mathbf{x} = \mathbf{A}^H\mathbf{b} \quad \Leftrightarrow \quad \mathbf{B}_\alpha\begin{bmatrix}\mathbf{r}\\ \mathbf{x}\end{bmatrix} := \begin{bmatrix}-\alpha\mathbf{I} & \mathbf{A}\\ \mathbf{A}^H & 0\end{bmatrix}\begin{bmatrix}\mathbf{r}\\ \mathbf{x}\end{bmatrix} = \begin{bmatrix}\mathbf{b}\\ 0\end{bmatrix}. \qquad (= \alpha^{-1}(Ax-b))$$

Good choice of α can give better C_{B_α}

hopefully $C_{B_\alpha} \approx C_A$

(instead of $C_B \approx C_{A^HA}$)

[Note: this is not regularization

$$(A^HA + \alpha I)x = A^Hb$$

$r = Ax-b$ $\qquad \tilde{B}_\alpha = \begin{bmatrix}-I & A\\ A^H & \alpha I\end{bmatrix}$]

3.3. Orthogonal Transformation Methods

Consider least-sq. problem $Ax = b$ $A \in \mathbb{R}^{m,n}$ $m \gg n$

and A has full rank $\operatorname{rank}(A) = n$

Idea: instead of solving $Ax = b$

find $\tilde{A}, \tilde{b}$

with $lsq(\tilde{A}, \tilde{b}) = lsq(A, b)$ s.t.

$\tilde{A}x = \tilde{b}$ is easier to solve.

↑

triangular system matrix

$$\left\| \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} - \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \right\|_2 \to \min \overset{(*)}{\Longrightarrow} \mathbf{x} = \begin{bmatrix} \mathbf{R} \end{bmatrix}^{-1} \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

$\mathbf{x} \mathrel{\hat{=}}$ least squares solution

R will be regular if $\operatorname{rank}(A) = n$.

Idea: If we have a (transformation) matrix $\mathbf{T} \in \mathbb{R}^{m,m}$ satisfying

$$\|\mathbf{Ty}\|_2 = \|\mathbf{y}\|_2 \quad \forall \mathbf{y} \in \mathbb{R}^m, \tag{3.3.1}$$

then $$\underset{\mathbf{y} \in \mathbb{R}^n}{\operatorname{argmin}} \|\mathbf{Ay} - \mathbf{b}\|_2 = \underset{\mathbf{y} \in \mathbb{R}^n}{\operatorname{argmin}} \left\|\tilde{\mathbf{A}}\mathbf{y} - \tilde{\mathbf{b}}\right\|_2,$$

where $\tilde{\mathbf{A}} = \mathbf{TA}$ and $\tilde{\mathbf{b}} = \mathbf{Tb}$.

⇒ multiply our LSE with a matrix $T \in \mathbb{R}^{m,m}$ with property $\|Ty\|_2 = \|y\|_2 \quad \forall y \in \mathbb{R}^m$ yields modified LSE with set lsg unchanged!

⇝ orthogonal / unitary matrices

Definition 3.3.4. Unitary and orthogonal matrices → [?, Sect. 2.8]

- $\mathbf{Q} \in \mathbb{K}^{n,n}$, $n \in \mathbb{N}$, is unitary, if $\mathbf{Q}^{-1} = \mathbf{Q}^H$.
- $\mathbf{Q} \in \mathbb{R}^{n,n}$, $n \in \mathbb{N}$, is orthogonal, if $\mathbf{Q}^{-1} = \mathbf{Q}^T$.

$$Q^H Q = I = Q Q^H$$

Theorem 3.3.5. Preservation of Euclidean norm

A matrix is unitary/orthogonal, if and only if the associated linear mapping preserves the 2-norm:

$$\mathbf{Q} \in \mathbb{K}^{n,n} \text{ unitary} \Leftrightarrow \|\mathbf{Qx}\|_2 = \|\mathbf{x}\|_2 \quad \forall \mathbf{x} \in \mathbb{K}^n .$$

Goal: Transform $Ax = b$ (*) to

$$Q^T A x = Q^T b \quad (**)$$

(Q ← orthogonal)

s.t. $Q^T A$ is upper triangular

Solving $A^T A x = A^T b$ vs. solving

$Q^T A x = Q^T b$ in least-sq. sense

$$A^T \underbrace{Q Q^T}_{=I} A x = A^T \underbrace{Q Q^T}_{=I} b$$

$$\Leftrightarrow \quad \underline{A^T A x = A^T b}$$

Problems (*) and (**) are equivalent in the least-squares sense

Recall: $A \in \mathbb{R}^{m,n}$, $Q \in \mathbb{R}^{m,m}$ orthogonal

if $R := Q^T A \in \mathbb{R}^{m,n}$ was upper triangular

least-squares problem becomes

$$\underbrace{Q^T A}_{R} x = Q^T b$$

$$(**) \Leftrightarrow \quad Rx = Q^T b$$

↑

as cheap as back-substitution

If $A \in \mathbb{R}^{m,n}$ can be decomposed in

$$A = Q R$$

↑ orth. $m \times m$ ↑ triang. $m \times n$

$$A^T A x = A^T b \quad \Leftrightarrow$$

$$R^T \underbrace{Q^T Q}_{=I} R x = R^T Q^T b$$

$$\Leftrightarrow \quad R^T R x = R^T Q^T b$$

$$\underline{x} = (R^T R)^{-1} R^T Q^T b$$

$$= \underline{R^{-1} Q^T b}$$

Instead of solving $A^T A x = A^T b$ with

$$c_{A^T A} = c_A^2$$

solving for $Rx = Q^T b$ is better conditioned

with $c_R = c_A$

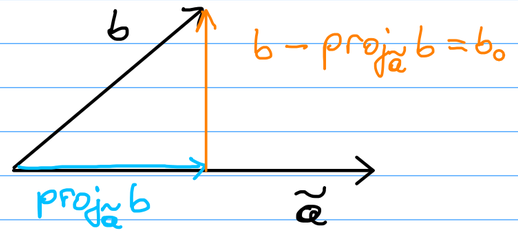
Next question: Does such a QR decomposition always exist? How to find it?

3.3.3 QR-Decomposition

First approach: Gram-Schmidt orthogonalization

Orthogonalization of 2 linearly independant vectors $a, b \in \mathbb{R}^m$

$$\tilde{a} := \frac{a}{\|a\|_2}$$



$\text{proj}_{\tilde{a}} b$ e.g. as least-sq. problem:

find τ that minimizes $\|\tau \cdot \tilde{a} - b\|_2$

$$\Leftrightarrow \quad \underbrace{\tilde{a}^T \tilde{a}}_{=1} \tau - \tilde{a}^T b = 0$$

$$\tau = \langle \tilde{a}, b \rangle$$

$$\text{proj}_{\tilde{a}} b = \langle \tilde{a}, b \rangle \tilde{a} \qquad (= (\tilde{a}^T b)\tilde{a})$$

$$b_0 = b - \text{proj}_{\tilde{a}} b = b - \langle \tilde{a}, b \rangle \tilde{a} \qquad \left(= b - \frac{\langle a, b \rangle}{\|a\|_2^2} a\right)$$

$$\tilde{b} := \frac{b_0}{\|b_0\|_2}$$

ONS: $\{\tilde{a}, \tilde{b}\}$

Gram-Schmidt: Procedure for set of k lin. ind. vectors

1: $\mathbf{q}^1 := \frac{\mathbf{a}^1}{\|\mathbf{a}^1\|_2}$ *% 1st output vector*
2: **for** $j = 2, \ldots, k$ **do**
{ *% Orthogonal projection*
3: $\mathbf{q}^j := \mathbf{a}^j$
4: **for** $\ell = 1, 2, \ldots, j-1$ **do** (GS)
5: $\{ \mathbf{q}^j \leftarrow \mathbf{q}^j - \langle \mathbf{a}^j, \mathbf{q}^\ell \rangle \mathbf{q}^\ell \}$ $\text{proj}_{\mathbf{q}^\ell} \mathbf{a}^j$
6: **if** $(\mathbf{q}^j = \mathbf{0})$ **then** STOP
7: **else** $\{ \mathbf{q}^j \leftarrow \frac{\mathbf{q}^j}{\|\mathbf{q}^j\|_2} \}$
8: }

Theorem 3.3.7. Span property of G.S. vectors

If $\{\mathbf{a}^1, \ldots, \mathbf{a}^k\}$ *is linearly independent, then Algorithm* (GS) *computes orthonormal vectors* $\mathbf{q}^1, \ldots, \mathbf{q}^k$ *satisfying*

$$\text{Span}\{\mathbf{q}^1, \ldots, \mathbf{q}^\ell\} = \text{Span}\{\mathbf{a}^1, \ldots, \mathbf{a}^\ell\}, \tag{1.5.2}$$

for all $\ell \in \{1, \ldots, k\}$.

Given $\{a^1, \dots, a^k\}$ lin. ind. → output $\{q^1, \dots, q^k\}$ orth. normal system (ONS)

For matrix $A = \begin{bmatrix} a^1 & a^2 & \dots & a^n \end{bmatrix} \in \mathbb{R}^{m,n}$

Step 1:

$$\begin{pmatrix} | & | & & | \\ a^1 & a^2 & \dots & a^n \\ | & | & & | \end{pmatrix} \begin{pmatrix} \tilde{t}_{11} & & \\ & 1 & 0 \\ 0 & & \ddots \\ & & & 1 \end{pmatrix} = \begin{pmatrix} | & | & & | \\ \tilde{t}_{11} a^1 & a^2 & \dots & a^n \\ | & | & & | \end{pmatrix}$$

Step 2:

$$\begin{pmatrix} | & | & & | \\ \tilde{t}_{11} a^1 & a^2 & \dots & a^n \\ | & | & & | \end{pmatrix} \begin{pmatrix} 1 & \tilde{\tilde{t}}_{12} & 0 & \\ 0 & \tilde{t}_{22} & 1 & 0 \\ \vdots & 0 & & \ddots \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} | & | & & | \\ \tilde{t}_{11} a^1 & \tilde{t}_{12} a^1 + \tilde{t}_{22} a^2 & \dots & a^n \\ | & | & & | \end{pmatrix}$$

$\tilde{t}_{12} = \tilde{t}_{11} \tilde{\tilde{t}}_{12}$

Altogether: Series of multiplications from the right by upper triangular matrices

$$Q = A \underbrace{T_1 T_2 \cdots T_n}_{=:T}$$

$Q = [q^1 \dots q^n] \in \mathbb{R}^{m,n}$ s.t.

$q^1 = t_{11} a^1$

$q^2 = t_{12} a^1 + t_{22} a^2$

$q^3 = t_{13} a^1 + t_{23} a^2 + t_{33} a^3$

$\vdots$

$q^n = t_{1n} a^1 + \dots + t_{nn} a^n$

and $Q = AT$

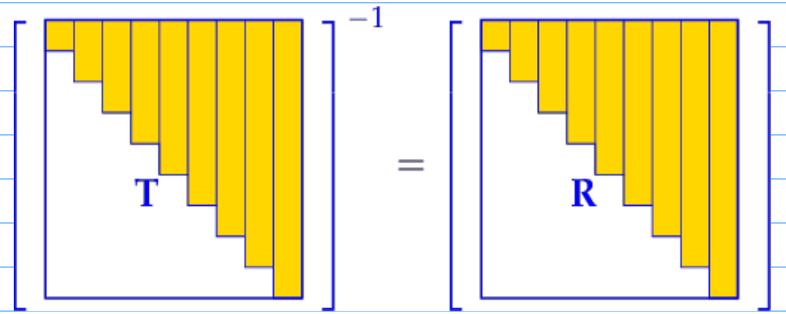
$T \in \mathbb{R}^{n,n}$ $\quad T = (t_{ij})_{i,j=1}^{n}$

upper triangular

T is regular because $\{a^1, \dots, a^n\}$ and $\{q^1, \dots, q^n\}$ are lin. ind.

T reg. upper triangular

$\Rightarrow T^{-1}$ reg. upper triangular

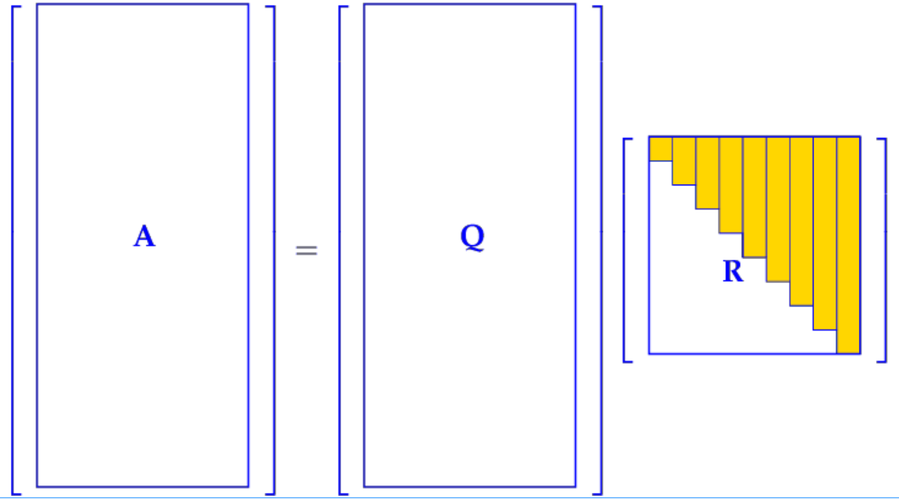


$R := T^{-1}$

$\Rightarrow \quad A = QR$

Thus, by (3.3.8), we have found an *upper triangular* $\mathbf{R} := \mathbf{T}^{-1} \in \mathbb{R}^{n,n}$ such that

$\mathbf{A} = \mathbf{QR} \;\leftrightarrow$

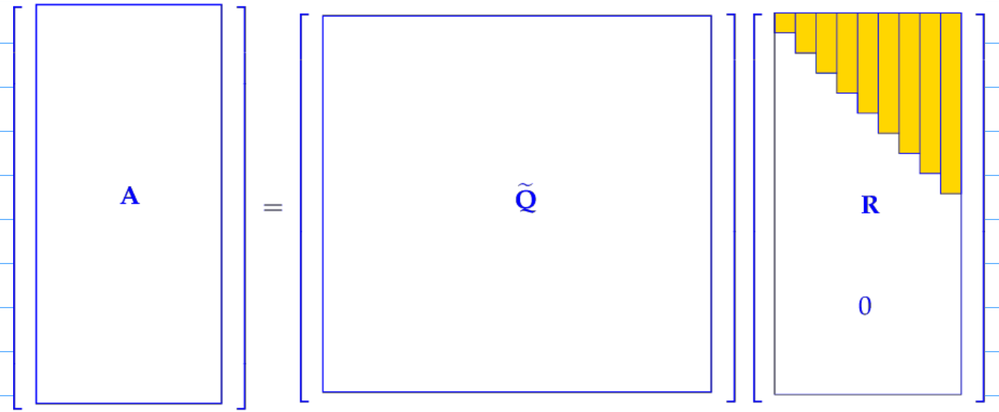


.

Argument:

$\blacktriangleright \quad \mathbf{A} = \tilde{\mathbf{Q}} \begin{bmatrix} \mathbf{R} \\ 0 \end{bmatrix}$

$\updownarrow$



$\Leftrightarrow \quad \tilde{\mathbf{Q}}^{\top} \mathbf{A} = \begin{bmatrix} \mathbf{R} \\ 0 \end{bmatrix}$.

[add $m-n$ zero rows to R and add $m-n$ columns to Q s.t. $\tilde{Q}$ is an orth. (unitary) matrix]

Theorem 3.3.9. QR-decomposition → [?, Satz 5.2], [?, Sect. 7.3]

For any matrix $\mathbf{A} \in \mathbb{K}^{n,k}$ *with* $\operatorname{rank}(\mathbf{A}) = k$ *there exists*

(i) a unique *Matrix* $\mathbf{Q}_0 \in \mathbb{R}^{n,k}$ *that satisfies* $\mathbf{Q}_0^{\mathrm{H}}\mathbf{Q}_0 = \mathbf{I}_k$, *and a* unique *upper triangular Matrix* $\mathbf{R}_0 \in \mathbb{K}^{k,k}$ *with* $(\mathbf{R})_{i,i} > 0$, $i \in \{1,\ldots,k\}$, *such that* (reduced / thin)

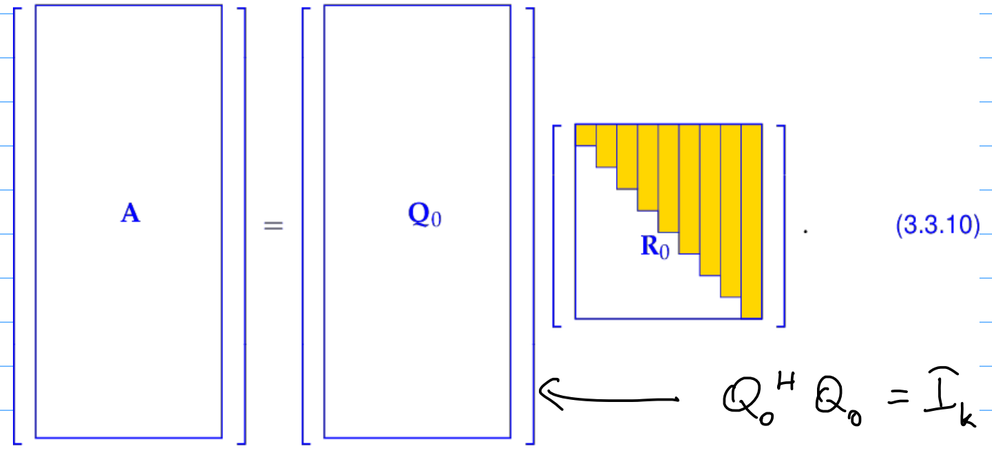
$$\mathbf{A} = \mathbf{Q}_0 \cdot \mathbf{R}_0 \quad \text{(“economical” QR-decomposition)},$$

(ii) a unitary Matrix $\mathbf{Q} \in \mathbb{K}^{n,n}$ *and a* unique *upper triangular* $\mathbf{R} \in \mathbb{K}^{n,k}$ *with* $(\mathbf{R})_{i,i} > 0$, $i \in \{1,\ldots,n\}$, *such that*

$$\mathbf{A} = \mathbf{Q} \cdot \mathbf{R} \quad \text{(full QR-decomposition)}.$$

If $\mathbb{K} = \mathbb{R}$ *all matrices will be real and* $\mathbf{Q}$ *is then orthogonal.*

$\mathbf{A} = \mathbf{Q}_0\mathbf{R}_0$, $\mathbf{Q}_0 \in \mathbb{K}^{n,k}$, $\mathbf{R}_0 \in \mathbb{K}^{k,k}$ upper triangular,

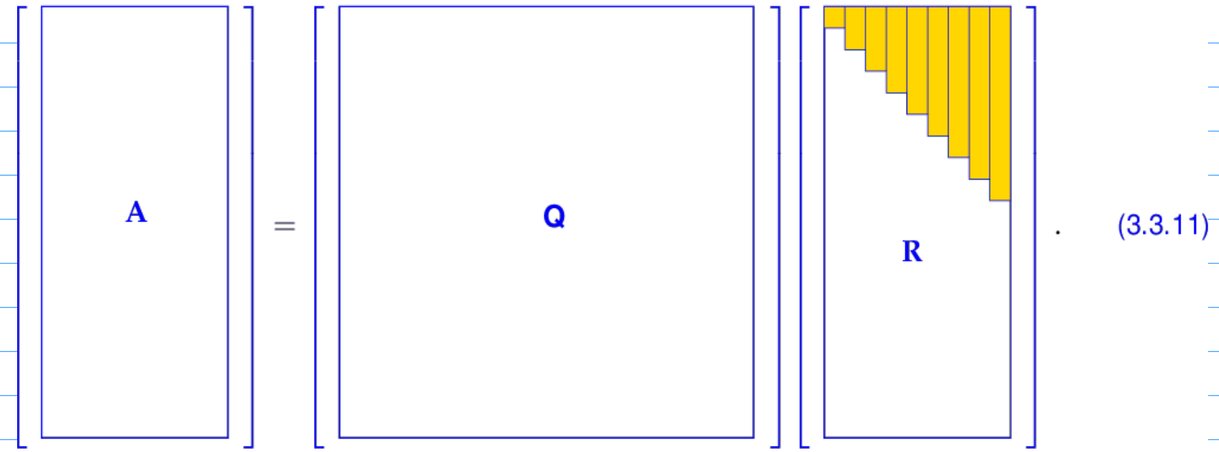


(3.3.10)

$\Leftarrow \quad Q_0^H Q_0 = I_k$ (Q_0 has orthonormal columns)

$Q^H Q = I_n = Q Q^H$

$\mathbf{A} = \mathbf{QR}$, $\mathbf{Q} \in \mathbb{K}^{n,n}$, $\mathbf{R} \in \mathbb{K}^{n,k}$,



(3.3.11)

Corollary 3.3.12. Uniqueness of QR-factorization

The “economical” QR-factorization (3.3.3.1) *of* $\mathbf{A} \in \mathbb{K}^{m,n}$, $m \geq n$, *with* $\operatorname{rank}(\mathbf{A}) = n$ *is unique, if we demand* $(\mathbf{R}_0)_{ii} > 0$.

Proof: Suppose $Q_1 R_1 = A = Q_2 R_2$

$Q_1 R_1 = Q_2 R_2$

$\Rightarrow \quad Q_1 = Q_2 R_2 R_1^{-1}$

$=: R$ (again upper triang.)

$$\Rightarrow \quad \underline{I_k} = Q_1^H Q_1 = R^H \underbrace{Q_2^H Q_2}_{=I_k} R = \underline{R^H R}$$

R is upper triang. & orth.

$$R = \begin{pmatrix} \overset{\neq 0}{r_{11}} & \overbrace{r_{12} \; r_{13} \; \cdots \; r_{1k}}^{=0} \\ r_{22} & \overbrace{r_{23} \; \cdots \; r_{2k}}^{=0} \\ 0 & \ddots \; \vdots \\ & \ddots \; r_{kk} \end{pmatrix}$$

orth.: $\langle r_1, r_j \rangle = 0 \qquad j \neq 1$

$$\Rightarrow \quad r_{12} = r_{13} = \ldots = r_{1k} = 0$$

$\Rightarrow r_{22} \neq 0$; using $\langle r_2, r_j \rangle = 0 \qquad j \neq 2$

ind.

$\Rightarrow$ R is a diag.

$$\Rightarrow R = \begin{pmatrix} r_{11} & & 0 \\ & \ddots & \\ 0 & & r_{kk} \end{pmatrix}$$

and $|r_{ii}|^2 = 1 \qquad (R^H R = I_k)$

if $r_{ii} > 0 \qquad \Rightarrow R = I_k \qquad (R = R_2 R_1^{-1})$

$\Rightarrow \quad R_2 = R_1$ and $Q_1 = Q_2$.

Note: Gram-Schmidt orthogonalization suffers from numerical instabilities

[possible cancellation in subtraction
$\rightsquigarrow$ then dividing by ≈ 0]

Example: $a_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $a_2 = \begin{pmatrix} 1+\varepsilon \\ 1 \end{pmatrix}$ $\varepsilon \ll 1$

$$\operatorname{span}\{a_1, a_2\} = \operatorname{span}\left\{\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right\}$$

but Gram-Schmidt gives

$$q_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \operatorname{proj}_{q_1} a_2 = \langle a_2, q_1 \rangle q_1$$

$$= \frac{2+\varepsilon}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

subtraction of close-by numbers

$$\tilde{q}_2 = \begin{pmatrix} 1+\varepsilon \\ 1 \end{pmatrix} - \frac{2+\varepsilon}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{\varepsilon}{2} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

$$\|\tilde{q}_2\|_2 = \frac{\varepsilon}{\sqrt{2}}$$

$$q_2 = \frac{\tilde{q}_2}{\|\tilde{q}_2\|_2} \leftarrow \text{division by } \varepsilon!$$

Numerically stable QR decomposition?

3.3.3.2 Computation of QR decomposition

Idea: Before: manipulation of columns of A by multiplication from the right [with triangular matrices]

$$Q = A T_1 \dots T_n$$

Now: Instead: Find a series of orthogonal transformations s.t. applied from the left yield triangular matrix:

$$\underbrace{Q_n \dots Q_2 Q_1}_{\text{row operations}} A = R$$

Then $Q := Q_1^T \dots Q_n^T$ yields $A = QR$.

→ idea similar to Gauss elimination
but now with orthogonal trafos

Householder Transformations

Set of orthogonal transformations is rich

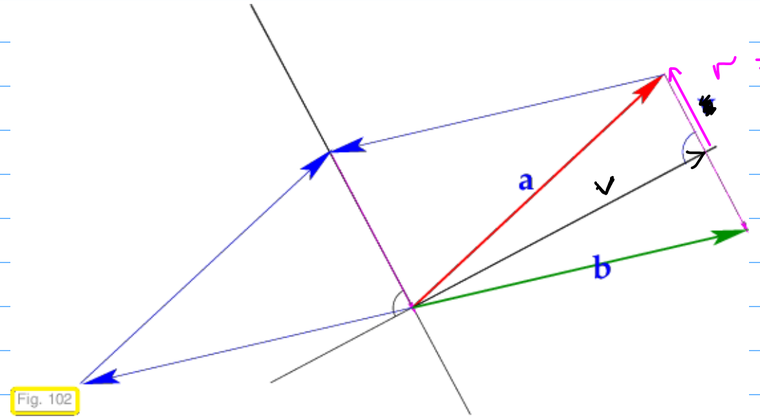
Intuition: orth. trafos preserve
- lengths of vectors
- angles between vectors

→ can only rotate & reflect vectors

Idea: Use only reflections → can be represented by projections

Given a vector $a \in \mathbb{R}^m$ & reflected over a vector $v \in \mathbb{R}^m$

reflection: $b \in \mathbb{R}^m$



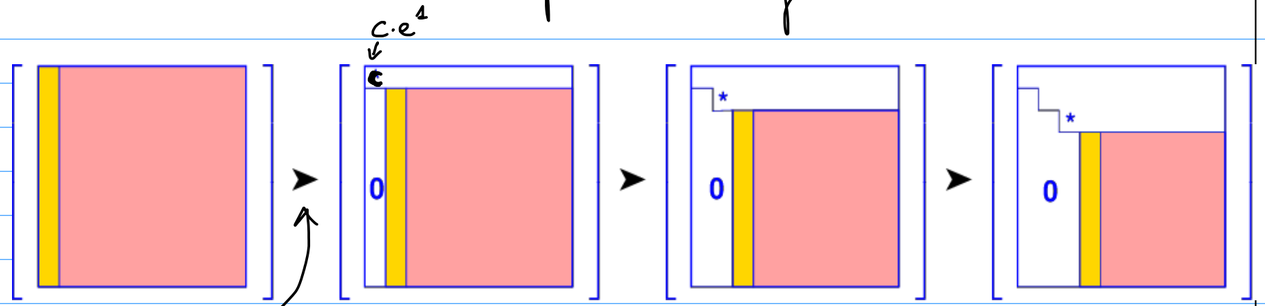
residual r is perpendicular to v

$$b = a - 2r = a - 2(a - \text{proj}_v a) = -a + 2\,\text{proj}_v a$$
$$= -(a - 2\,\text{proj}_v a)$$

$$H_v a = -(a - 2\,\text{proj}_v a) = -\left(a - 2\,\frac{v^T a}{v^T v}\,v\right)$$

$$= -\left(I_m - 2\frac{vv^T}{v^Tv}\right)a$$

How to use this for triangularization:



Step 1: Find constant c s.t.

$$H_v a = c \cdot e^1 \leftarrow \text{first unit vector } \begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix}$$

$$-a + 2\frac{v^Ta}{v^Tv}v = ce^1$$

or equiv. find a constant c s.t.

$$a - 2\frac{v^Ta}{v^Tv}v = ce^1$$

$$\Rightarrow \quad v = (a - ce^1)\frac{v^Tv}{2v^Ta}$$

$\rightsquigarrow$ v must be parallel to $a - ce^1$

Note: Scaling of v doesn't effect the formula

Set $v = a - ce^1$

$$\frac{v^Tv}{2v^Ta} = 1$$

$$\frac{\|a - ce^1\|_2^2}{2\langle a - ce^1, a\rangle} = 1$$

$$\|a\|_2^2 - 2c\langle a, e^1\rangle + c^2 = \|a\|_2^2 - 2c\langle a, e^1\rangle$$

$$c = \pm\|a\|_2$$

Choose c s.t. cancellation is avoided

If a is almost parallel to e^1

either $a + \|a\|_2 e^1$ or $a - \|a\|_2 e^1$

is small $\Rightarrow$ would be dividing by $\|v\|_2$ [could yield instability in HH reflections]

$$v = \begin{cases} \frac{1}{2}\left(a - \|a\|_2 e^1\right) & \text{if } a_1 < 0 \\ \frac{1}{2}\left(a + \|a\|_2 e^1\right) & \text{if } a_1 > 0 \end{cases}$$

For a matrix $A \in \mathbb{R}^{m,n}$: HH reflection on the first column

$$H_{v^1} a^1 = c e^1$$

$$\Rightarrow H_{v^1} A = \begin{bmatrix} c & & \\ 0 & & \\ \vdots & \square & \\ 0 & & \end{bmatrix}$$

↑ first step to triangularisation

j-th step of HH :

vector a^j is split into $\begin{bmatrix} a_1^j \\ a_2^j \end{bmatrix} \} \in \mathbb{R}^j$ # we want to keep
$\} \in \mathbb{R}^{m-j}$

Find v^j s.t. $H_{v^j} a^j = \begin{bmatrix} \tilde{a}_1^j \\ 0 \\ \vdots \\ 0 \end{bmatrix} \} \in \mathbb{R}^j$
$\} m-j$ zeros

Choose $v^j = \begin{bmatrix} 0 \\ a_2^j \end{bmatrix} - c_j e^j$ with $c_j = \pm \|a_2^j\|$

↑ choose s.t. no cancellation

Calculation shows: $\frac{\langle a^j, v^j \rangle}{\|v^j\|_2^2} = \frac{1}{2}$

$$H_{v^j} a^j = a^j - 2 \frac{\langle a^j, v^j \rangle}{\|v^j\|_2^2} v^j = a^j - v^j$$

(a^j and v^j coincide on indices $j+1$ to m)

$$\Rightarrow H_{v^j} a^j = \begin{bmatrix} \tilde{a}^j \\ a_1^j \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \} \in \mathbb{R}^j \\ \} \, m-j \text{ zeros} \end{matrix}$$

Note:

$H_{v^j} q^k$ with $k < j$ $\qquad q^k = H_{v^k} a^k$

$$H_{v^j} q^k = q^k - 2 \frac{\langle v^j, q^k \rangle}{\|v^j\|_2^2} v^j = q^k$$

$\langle v^j, q^k \rangle = 0$ ($k \leq j-1$)

first $j-1$ entries zero; only first k entries nonzero

Altogether:

$$H_{v^n} \cdots H_{v^1} A = R$$

(orthogonal)

$$\Rightarrow Q = H_{v^1}^T \cdots H_{v^n}^T \quad \text{orth.}$$

v^j: first $j-1$ entries are zero

Q is stored implicitly by storing vectors $v^1, \ldots, v^n$ as lower triangular matrix (compressed format)

Householder reflections in EIGEN:

C++-code 3.3.34: QR-decompositions in EIGEN

```
#include <Eigen/QR>

// Computation of full QR-decomposition (3.3.3.1),
// dense matrices built for both QR-factors (expensive!)
std::pair<MatrixXd,MatrixXd> qr_decomp_full(const MatrixXd& A) {
  Eigen::HouseholderQR<MatrixXd> qr(A);
  MatrixXd Q = qr.householderQ(); //
  MatrixXd R = qr.matrixQR().template triangularView<Eigen::Upper>();
  return std::pair<MatrixXd,MatrixXd>(Q,R);
}

// Computation of economical QR-decomposition (3.3.3.1),
// dense matrix built for Q-factor (possibly expensive!)
std::pair<MatrixXd,MatrixXd> qr_decomp_eco(const MatrixXd& A) {
  using index_t = MatrixXd::Index;
  const index_t m = A.rows(),n = A.cols();
  Eigen::HouseholderQR<MatrixXd> qr(A);
  MatrixXd Q = (qr.householderQ()*MatrixXd::Identity(m,n)); //
  MatrixXd R = qr.matrixQR().block(0,0,n,n).template
     triangularView<Eigen::Upper>(); //
  return std::pair<MatrixXd,MatrixXd>(Q,R);
}
```

apply n times HH reflections

apply n times HH reflections

In general: only HH reflections are applied instead of building Q

Asymptotic complexity of Householder QR-decomposition

The computational effort for **HouseholderQR()** of $\mathbf{A} \in \mathbb{R}^{m,n}$, $m > n$, is $O(mn^2)$ for $m,n \to \infty$.

each HH reflection applied to one vector $\sim O(m)$

applied to A $\sim O(mn)$

n such reflections $\rightsquigarrow$ overall $O(n \cdot mn)$

Normal equations vs. orthogonal transformations method

Superior numerical stability ($\to$ Def. 1.5.85) of orthogonal transformations methods:

- Use orthogonal transformations methods for least squares problems (3.1.38), whenever $\mathbf{A} \in \mathbb{R}^{m,n}$ *dense* and n small.

SVD/QR-factorization cannot exploit sparsity:

- Use normal equations in the expanded form (3.2.8)/(3.2.9), when $\mathbf{A} \in \mathbb{R}^{m,n}$ sparse ($\to$ Notion 2.7.1) and m,n big.

Alternative method for QR factorization:

Givens rotations [build Q through rotations]

3.4. Singular Value Decomposition

A different orth. decomposition

Theorem 3.4.1. singular value decomposition → [?, Thm. 9.6], [?, Thm. 11.1]

For any $\mathbf{A} \in \mathbb{K}^{m,n}$ *there are* unitary *matrices* $\mathbf{U} \in \mathbb{K}^{m,m}$, $\mathbf{V} \in \mathbb{K}^{n,n}$ *and a (generalized) diagonal* (*) *matrix* $\mathbf{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m,n}$, $p := \min\{m,n\}$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$ *such that*

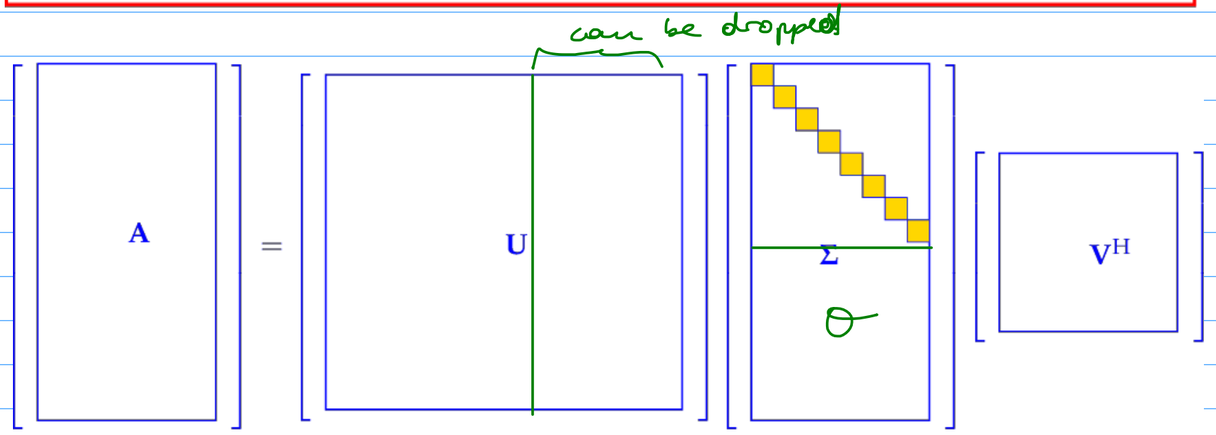
$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H .$$

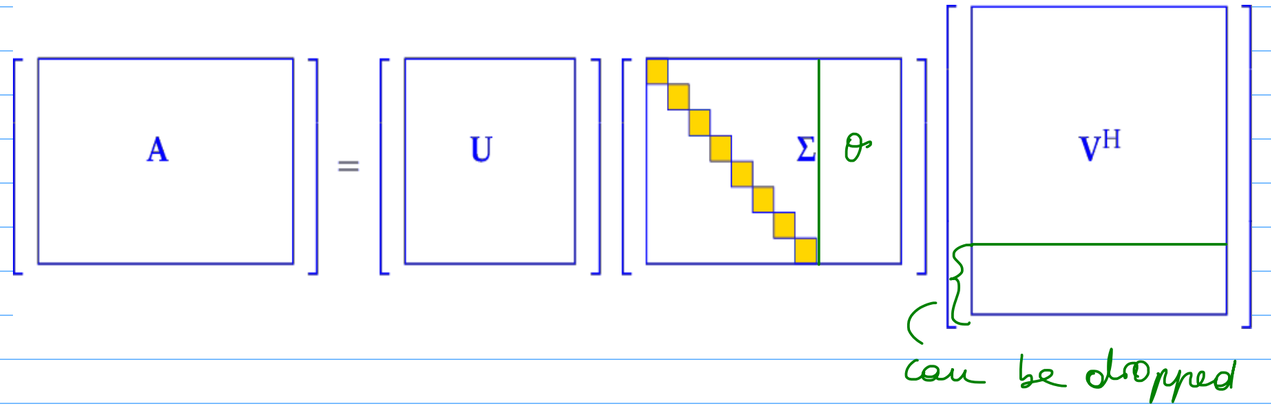
$$U^H U = I_m = U U^H$$

$$V^H V = I_n = V V^H$$

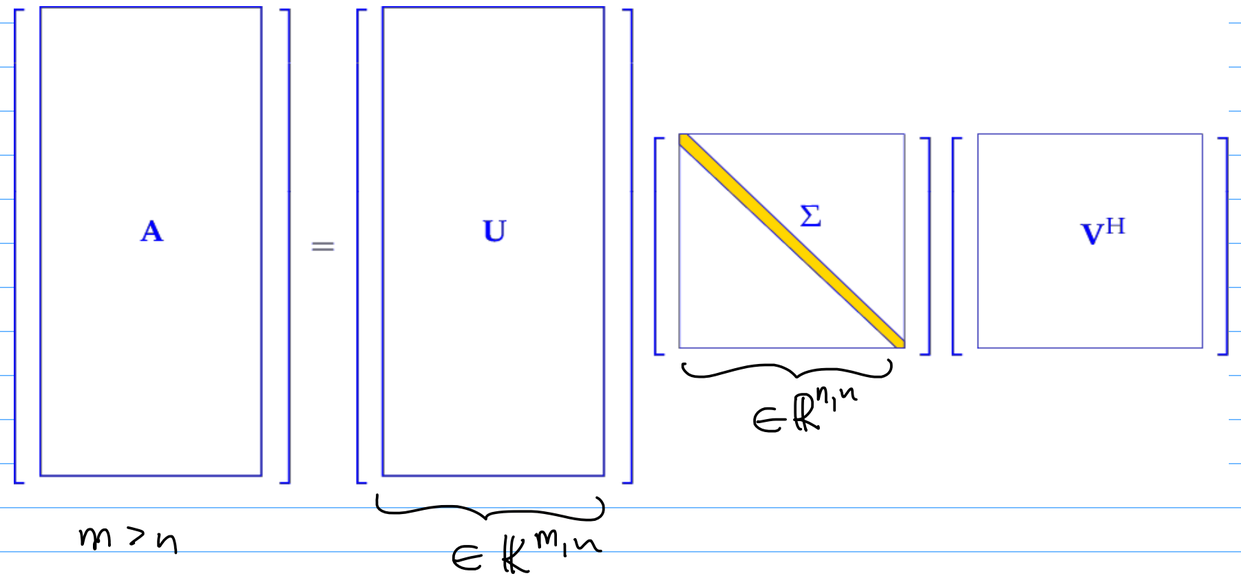
Definition 3.4.3. Singular value decomposition (SVD)

The decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ of Thm. 3.4.1 is called singular value decomposition (SVD) of $\mathbf{A}$. The diagonal entries σ_i of $\mathbf{\Sigma}$ are the singular values of $\mathbf{A}$.





"Reduced / thin SVD" [as for QR]



$m > n$

$m < n$: $\Sigma \in \mathbb{R}^{m,m}$ $V \in \mathbb{K}^{n,m}$

Lemma 3.4.5.

The squares σ_i^2 of the non-zero singular values of $\mathbf{A}$ are the non-zero eigenvalues of $\mathbf{A}^H\mathbf{A}$, $\mathbf{A}\mathbf{A}^H$ with associated eigenvectors $(\mathbf{V})_{:,1},\ldots,(\mathbf{V})_{:,p}$, $(\mathbf{U})_{:,1},\ldots,(\mathbf{U})_{:,p}$, respectively.

Lemma 3.4.11. SVD and rank of a matrix → [?, Cor. 9.7]

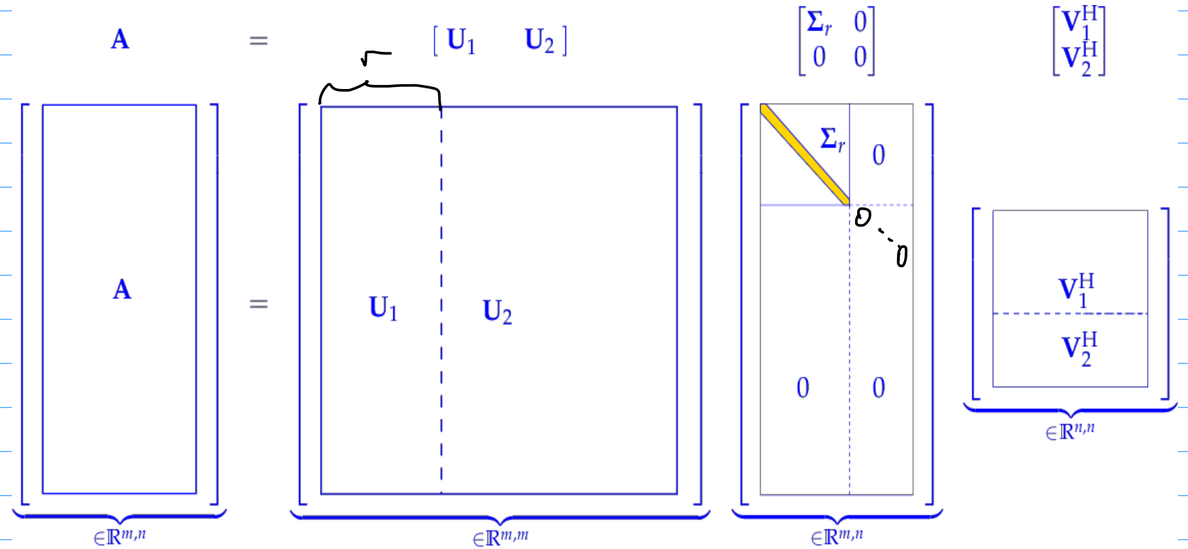
If, for some $1 \le r \le p := \min\{m,n\}$, the singular values of $\mathbf{A} \in \mathbb{K}^{m,n}$ satisfy $\sigma_1 \ge \cdots \ge \sigma_r > \sigma_{r+1} = \cdots \sigma_p = 0$, then

- $\operatorname{rank}(\mathbf{A}) = r$ *(no. of non-zero singular values)*,
- $\mathcal{N}(\mathbf{A}) = \operatorname{Span}\{(\mathbf{V})_{:,r+1},\ldots,(\mathbf{V})_{:,n}\}$,
- $\mathcal{R}(\mathbf{A}) = \operatorname{Span}\{(\mathbf{U})_{:,1},\ldots,(\mathbf{U})_{:,r}\}$.

→ SVD reveals rank of A!

→ U, V encode orthonormal representations of $\mathcal{N}(A)$, $\mathcal{R}(A)$

$r = \operatorname{rank}(A)$



(3.4.22)

- span $\{(V)_{:,r+1}, \dots, (V)_{:,n}\} = \mathcal{N}(A)$:

Take any $(V)_{:,j}$ $j \in \{r+1, \dots, n\}$

$$A(V)_{:,j} = U \Sigma V^T (V)_{:,j} = U \Sigma e^j$$

$\Sigma \leftarrow \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}$

$j > r$

$$U \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} e^j = 0$$

$$\Rightarrow \forall j \in \{r+1, \dots, n\}: \quad A(V)_{:,j} = 0$$

$$\Rightarrow (V)_{:,j} \in \mathcal{N}(A)$$

$\{(V)_{:,r+1}, \dots, (V)_{:,n}\}$ ONB for $\mathcal{N}(A)$

- columns of U_1 to span $\mathcal{R}(A)$

$y \in \mathcal{R}(A) \Leftrightarrow \exists x \in \mathbb{R}^n$ s.t. $Ax = y$

$$U \Sigma V^T x = y$$

$$\underbrace{\begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}}_{U_1 \Sigma_r V_1^T} x = y$$

$$U_1 \underbrace{\Sigma_r V_1^T x}_{=: c \in \mathbb{R}^r} = y \qquad c = \begin{pmatrix} c_1 \\ \vdots \\ c_r \end{pmatrix}$$

$$\Rightarrow \quad y = c_1 \underbrace{(U_1)_{:,1}}_{= (U)_{:,1}} + \ldots + c_r \underbrace{(U_1)_{:,r}}_{(U)_{:,r}}$$

for every $y \in \mathcal{R}(A)$:

$$\exists\, c \in \mathbb{R}^r \text{ s.t. } y = \sum_{j=1}^{r} c_j (U)_{:,j}$$

$$\Rightarrow \quad \operatorname{span}\{(U)_{:,1}, \ldots, (U)_{:,r}\} = \mathcal{R}(A)$$

3.4.2. SVD in EIGEN

JacobiSVD

C++-code 3.4.13: Computing SVDs in EIGEN

```
2  # include <Eigen/SVD>
3
4  // Computation of (full) SVD A = UΣV^H → Thm. 3.4.1
5  // SVD factors are returned as dense matrices in natural order
6  std::tuple<MatrixXd,MatrixXd,MatrixXd> svd_full(const MatrixXd& A) {
7    Eigen::JacobiSVD<MatrixXd> svd(A, Eigen::ComputeFullU |
       Eigen::ComputeFullV);
8    MatrixXd U = svd.matrixU(); // get unitary (square) matrix U
9    MatrixXd V = svd.matrixV(); // get unitary (square) matrix V
10   VectorXd sv = svd.singularValues(); // get singular values as vector
11   MatrixXd Sigma = MatrixXd::Zero(A.rows(), A.cols());
12   const unsigned p = sv.size(); // no. of singular values
13   Sigma.block(0,0,p,p) = sv.asDiagonal(); // set diagonal block of Σ
14   return std::tuple<MatrixXd,MatrixXd,MatrixXd>(U,Sigma,V);
15 }
16
17 // Computation of economical (thin) SVD A = UΣV^H, see (3.4.4)
18 // SVD factors are returned as dense matrices in natural order
19 std::tuple<MatrixXd,MatrixXd,MatrixXd> svd_eco(const MatrixXd& A) { .
20   Eigen::JacobiSVD<MatrixXd> svd(A, Eigen::ComputeThinU |
       Eigen::ComputeThinV);
21   MatrixXd U = svd.matrixU(); // get unitary (~~square~~) matrix U
22   MatrixXd V = svd.matrixV(); // get unitary (~~square~~) matrix V
23   VectorXd sv = svd.singularValues(); // get singular values as vector
24   MatrixXd Sigma = sv.asDiagonal(); // build diagonal matrix Σ
25   return std::tuple<MatrixXd,MatrixXd,MatrixXd>(U,Sigma,V);
26 }
```

without the compute U & V flags:

only singular values are computed

"Numerical rank" e.g. rank()

$$r := \#\left\{ \vec{\sigma}_i : |\vec{\sigma}_i| \geq tol \cdot \max_j \{|\vec{\sigma}_j|\} \right\}$$

Default: tol = EPS setThreshold()

to set tol manually

Cost of thin SVD: $\Theta(mn^2)$ $m > n$

JacobiSVD is numerically stable

3.4.3 Generalized solutions by SVD

Assume $A \in \mathbb{K}^{m,n}$ $m \geq n$ $\operatorname{rank}(A) = r \leq n$

As before:

$$A = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^H \\ V_2^H \end{bmatrix}$$

least squares: $\min \|Ax-b\|_2$

$$\|Ax-b\|_2^2 = \|U\Sigma V^H x - b\|_2^2 = \|U^H(U\Sigma V^H x - b)\|_2^2$$

$$= \|\Sigma V^H x - U^H b\|_2^2$$

$$= \left\| \begin{bmatrix} \Sigma_r V_1^H x \\ 0 \end{bmatrix} - \begin{bmatrix} U_1^H b \\ U_2^H b \end{bmatrix} \right\|_2^2$$

$$= \left\| \begin{bmatrix} \Sigma_r V_1^H x - U_1^H b \\ -U_2^H b \end{bmatrix} \right\|_2^2 = \| \Sigma_r V_1^H x - U_1^H b \|_2^2 + \| U_2^H b \|_2^2$$

$\|\cdot\|_2^2$: $\|\cdot\|_{\mathbb{K}^m}$; $\|\cdot\|_{\mathbb{K}^r}$; $\|\cdot\|_{\mathbb{K}^{m-r}}$

$\| U_2^H b \|_2^2$: fixed

equivalently: $\min_x \| \Sigma_r V_1^H x - U_1^H b \|_2^2$

Choose x s.t.

$$\Sigma_r V_1^H x = U_1^H b \quad \text{LSE}$$

$r \times n$ (possibly underdet)

If $r < n$: generalized solution

$\rightsquigarrow$ non-uniqueness from $\mathcal{N}(V_1^H)$

$$\mathcal{N}(V_1^H) = \mathcal{R}(V_1)^\perp = \mathcal{R}(V_2)$$

choose $x \in \mathcal{R}(V_1)$

$x = V_1 z$ for some $z \in \mathbb{K}^r$

$$\Sigma_r \underbrace{V_1^H V_1}_{=I} z = U_1^H b$$

$$z = \Sigma_r^{-1} U_1^H b$$

Generalized solution: $x = V_1 z = V_1 \Sigma_r^{-1} U_1^H b$

Theorem 3.4.30. Pseudoinverse and SVD

If $\mathbf{A} \in \mathbb{K}^{m,n}$ has the SVD decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ partitioned as in (3.4.22), then its Moore-Penrose pseudoinverse ($\rightarrow$ Thm. 3.1.37)is given by $\mathbf{A}^\dagger = \mathbf{V}_1 \mathbf{\Sigma}_r^{-1} \mathbf{U}_1^H$.

[this requires rank(A) $\rightsquigarrow$ determine numerical rank]