ETH Lecture 401-0663-00L Numerical Methods for CSE

Numerical Methods for Computational Science and Engineering

Prof. R. Hiptmair, SAM, ETH Zurich

(with contributions from Prof. P. Arbenz and Dr. V. Gradinaru)

Autumn Term 2016

URL: http://www.sam.math.ethz.ch/~hiptmair/tmp/NumCSE/NumCSE16.pdf

III. Direct Methods for Linear Least Squares Problem

Overdetermined linear systems:

may not exist, unless $\underline{b} \in \mathcal{R}(A)$

$$\mathbf{x} \in \mathbb{R}^n: \quad \text{“}\mathbf{A}\mathbf{x} = \mathbf{b}\text{”}, \tag{3.0.1}$$

$\mathbf{b} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{m,n}$, $m \geq n$.

$$\begin{bmatrix} \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \end{bmatrix}$$

$\mathcal{R}(A) \hat{=}$ column space

Example: Parameter estimation in linear models

Physical quantities $y \in \mathbb{R}$, $\underline{x} \in \mathbb{R}^n$, physical law $y = \underline{a}^T \underline{x} + \beta$ for some $\underbrace{\underline{a} \in \mathbb{R}^n, \beta \in \mathbb{R}}_{\text{parameter}}$

Measurement $(\underline{x}_i, y_i)_{i=1}^m$, $m \gg n$

$\Rightarrow \quad \underline{a}^T \underline{x}_i + \beta = y_i \quad , i = 1, \dots, m$

$$\begin{bmatrix} \mathbf{x}_1^\top & 1 \\ \vdots & \vdots \\ \mathbf{x}_m^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \beta \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}, \tag{3.0.6}$$

$\hookrightarrow$ tainted by measurement error

Relative point locations (on a line) from distances

$\hookrightarrow$ positions $x_i \in \mathbb{R}$, $i = 1, \dots, n$, $x_i < x_{i+1}$

Measure: $d_{ij} \approx |x_i - x_j|$, $i \neq j$

Using all data $\Rightarrow$ o.d. LSE

$$\begin{array}{c} x_i - x_j = d_{ij}, \\ 1 \leq j < i \leq n. \\ \updownarrow \\ \mathbf{A}\mathbf{x} = \mathbf{b} \end{array} \quad \leftrightarrow \quad \underbrace{\begin{bmatrix} -1 & 1 & 0 & \dots & \dots & 0 \\ -1 & 0 & 1 & 0 & & \\ \vdots & & \ddots & & & \\ -1 & \dots & & & 0 & 1 \\ 0 & -1 & 1 & & & 0 \\ \vdots & & & & & \vdots \\ 0 & \dots & & & -1 & 1 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_{12} \\ d_{13} \\ \vdots \\ d_{1n} \\ d_{23} \\ \vdots \\ d_{n-1,n} \end{bmatrix} \tag{3.0.11}$$

Here $\mathcal{N}(A) = \operatorname{Span}\{\underline{1}\}$

②

3.1. Least Squares Solution Concepts

o.d. LSE : $A\underline{x} = \underline{b}$, $A \in \mathbb{R}^{m,n}$, $m \geq n$

Idea: seek x that makes residual $\underline{b} - A\underline{x}$ small.

Definition 3.1.3. Least squares solution

For given $\mathbf{A} \in \mathbb{K}^{m,n}$, $\mathbf{b} \in \mathbb{K}^m$ the vector $\mathbf{x} \in \mathbb{R}^n$ is a least squares solution of the linear system of equations $\mathbf{Ax} = \mathbf{b}$, if

$$\mathbf{x} \in \underset{\mathbf{y}\in\mathbb{K}^n}{\operatorname{argmin}} \|\mathbf{Ay} - \mathbf{b}\|_2^2 := \underbrace{\sum_{i=1}^{m}\Big(\sum_{j=1}^{n}(A)_{ij}y_j - b_i\Big)^2}_{\text{polynomial in } y_i}$$

$$\Updownarrow$$

$$\|\mathbf{Ax} - \mathbf{b}\|_2 = \inf_{\mathbf{y}\in\mathbb{K}^n} \|\mathbf{Ay} - \mathbf{b}\|_2 .$$

For parameter estimation

$$y_i = \underline{a}^T \cdot x_i + \beta \Rightarrow (\mathbf{a}, \beta) = \underset{\mathbf{p}\in\mathbb{R}^n, \gamma\in\mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^{m} |y_i - \mathbf{p}^\top \mathbf{x}_i - \gamma|^2 \qquad (3.1.7)$$

Geometric considerations:

Ax closest to $\underline{b}$ ≙ orth. projection of $\underline{b}$ on $\mathcal{R}(A)$

▷ Existence of l.s.s. $\underline{x}$!

$(A\underline{y})^T(\underline{b} - A\underline{x}) = 0 \quad \forall \underline{y} \in \mathbb{R}^n$

[Orthogonality!]

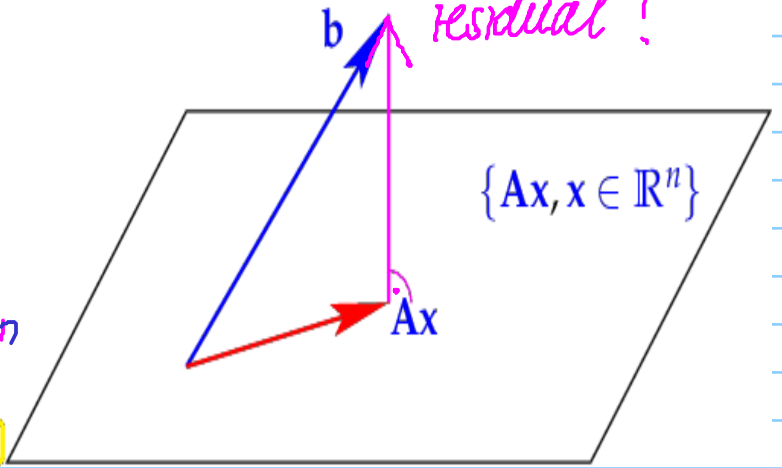


Fig. 96

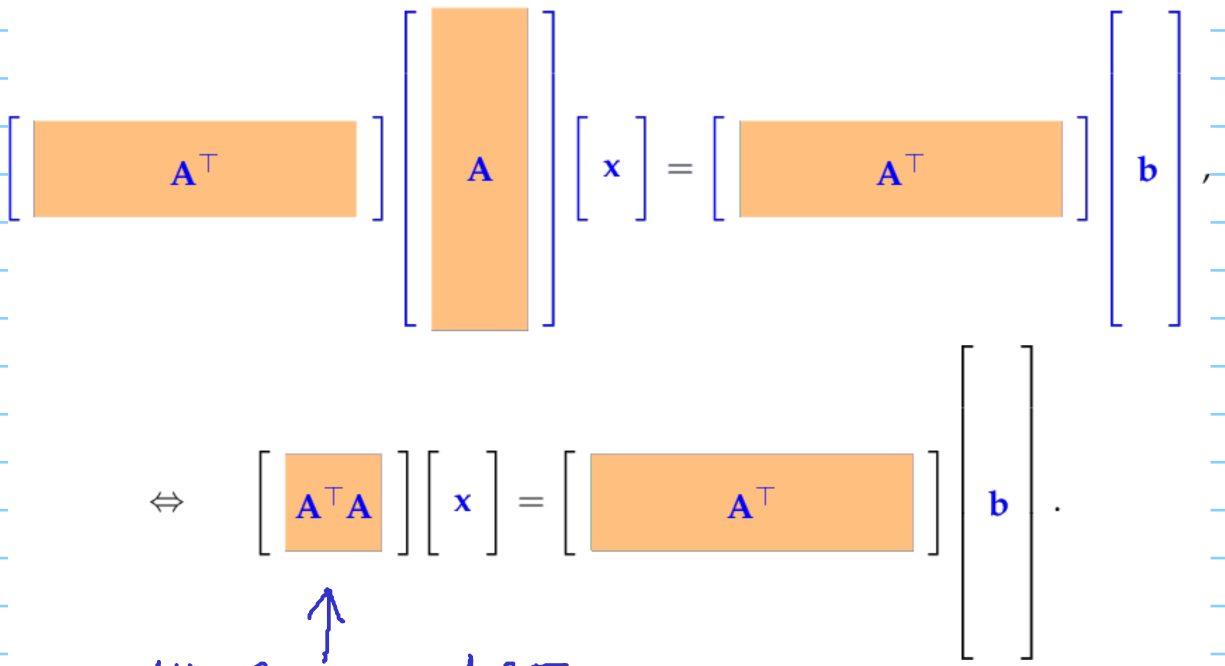
$\Rightarrow \quad A^T(\underline{b} - A\underline{x}) = 0 \quad \forall \underline{x} \in lsq(A,b)$

Set of least squares solutions

Theorem 3.1.10. Obtaining least squares solutions by solving normal equations

The vector $\mathbf{x} \in \mathbb{R}^n$ *is a least squares solution (→ Def. 3.1.3) of the linear system of equations* $\mathbf{Ax} = \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{m,n}$, $\mathbf{b} \in \mathbb{R}^m$, if and only if *it solves the* normal equations (NE)

$$\boxed{\mathbf{A}^\top\mathbf{A}\mathbf{x} = \mathbf{A}^\top\mathbf{b}} . \qquad (3.1.11)$$



NE ≙ n×n LSE

▷ $lsq(A,b)$ is an affine space parallel to $\mathcal{N}(A^TA)$

③

Uniqueness ? $\leftrightarrow$ $\mathcal{N}(A^TA)$

Theorem 3.1.18. Kernel and range of $\mathbf{A}^\top\mathbf{A}$

For $\mathbf{A} \in \mathbb{R}^{m,n}$, $m \geq n$, holds

$$\mathcal{N}(\mathbf{A}^\top\mathbf{A}) = \mathcal{N}(\mathbf{A})\,, \tag{3.1.19}$$
$$\mathcal{R}(\mathbf{A}^\top\mathbf{A}) = \mathcal{R}(\mathbf{A}^\top)\,. \tag{3.1.20}$$

[$x \in \mathcal{N}(A^TA) \Rightarrow x^TA^TAx = 0 \Leftrightarrow \|Ax\|_2^2 = 0 \Leftrightarrow Ax = 0$]

lsq sol. $\underline{x}$ unique $\Leftrightarrow$ $\mathcal{N}(A) = \{0\}$

[$A \in \mathbb{R}^{m,n}$, $m \geq n$] $\Leftrightarrow$ $\mathrm{Rank}(A) = n$

(full-rank condition, FRC)

Corollary 3.1.22. Uniqueness of least squares solutions

If $m \geq n$ and $\mathcal{N}(\mathbf{A}) = \{\mathbf{0}\}$, then the linear system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{m,n}$, $\mathbf{b} \in \mathbb{R}^m$, has a unique least squares solution ($\rightarrow$ 3.1.3)

$$\mathbf{x} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}\,, \tag{3.1.23}$$

that can be obtained by solving the normal equations (3.1.11).

FRC for parameter estimation, $n = 1$

$$\mathrm{rank}\left(\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ \vdots & \vdots \\ x_m & 1 \end{bmatrix}\right) = 2 \quad \Leftrightarrow \quad \exists i,j \in \{1,\dots,m\}\colon x_i \neq x_j\,,$$

3.1.3. Moore-Penrose Ψ-do Inverse

Definition 3.1.32. Generalized solution of a lineasr system of equations

The generalized solution $\mathbf{x}^\dagger \in \mathbb{R}^n$ of a linear system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{m,n}$, $\mathbf{b} \in \mathbb{R}^m$, is defined as

$$\mathbf{x}^\dagger := \mathrm{argmin}\{\|\mathbf{x}\|_2 \colon \mathbf{x} \in \mathrm{lsq}(\mathbf{A},\mathbf{b})\}\,. \tag{3.1.33}$$

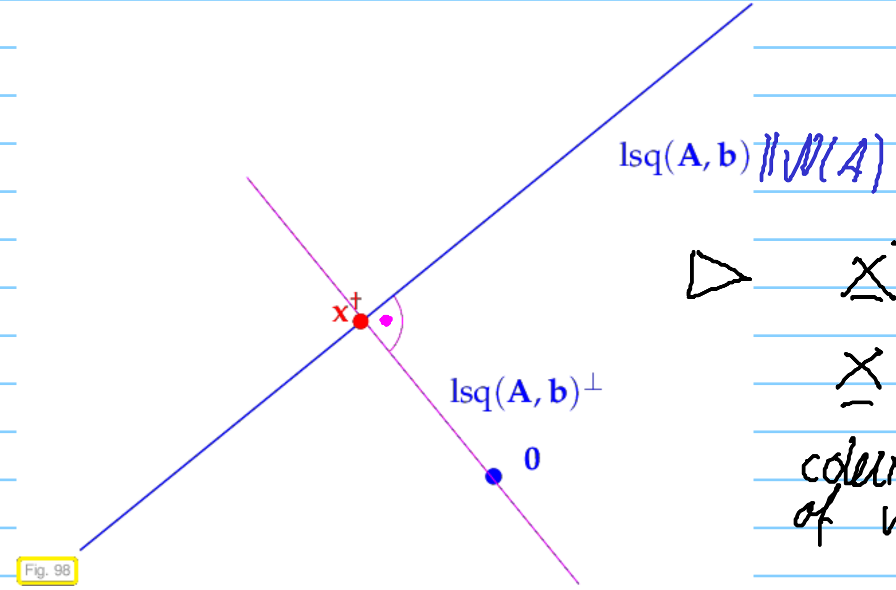


Fig. 98

$\triangleright$ $\underline{x}^+ \in \mathcal{N}(A)^\perp$

$\underline{x}^+ \in \mathcal{R}(V)$, when

columns of V form a basis of $\mathcal{N}(A)^\perp$: $\underline{x}^+ = V\underline{y}$

▷ Plug $x = Vy$ in NE

⇒ LSE for x^+: $\underbrace{V^TA^TAV}_{\text{regular matrix}}y = V^TA^Tb$

▷ $x^+ = \underbrace{V(V^TA^TAV)^{-1}V^TA^T}_{\hat{=}\ \text{M.-P. Ψ-do inverse}}\, b$

[independent of V → exercise!]

3.2. Normal Equation Methods

Algorithm: Normal equation method to solve full-rank least squares problem $\mathbf{Ax} = \mathbf{b}$

❶ Compute regular matrix $\mathbf{C} := \mathbf{A}^\top\mathbf{A} \in \mathbb{R}^{n,n}$.
❷ Compute right hand side vector $\mathbf{c} := \mathbf{A}^\top\mathbf{b}$.
❸ Solve s.p.d. (→ Def. 1.1.8) linear system of equations: $\mathbf{Cx} = \mathbf{c}$ → § 2.8.13

[$x^TCx = x^TA^TAx = \|Ax\|^2 > 0 \quad \forall x \neq 0$]

C++11 code 3.2.1: Solving a linear least squares probel via normal equations

```
//! Solving the overdetermined linear system of equations
//! Ax = b by solving normal equations (3.1.11)
//! The least squares solution is returned by value
VectorXd normeqsolve(const MatrixXd &A,const VectorXd &b) {
  if (b.size() != A.rows()) throw runtime_error("Dimension mismatch");
  // Cholesky solver
  VectorXd x = (A.transpose()*A).llt().solve(A.transpose()*b);
  return x;
}
```

$\underbrace{\texttt{(A.transpose()*A)}}_{\text{cost } O(n^2m)}$, $\underbrace{\texttt{(A.transpose()*b)}}_{\text{cost } O(n\cdot m)}$

`llt()` ↑ GE for s.p.d. LSE: $O(n^3)$

step ❶: cost $O(mn^2)$
step ❷: cost $O(nm)$
step ❸: cost $O(n^3)$

▶ cost $O(n^2m + n^3)$ for $m, n \to \infty$.

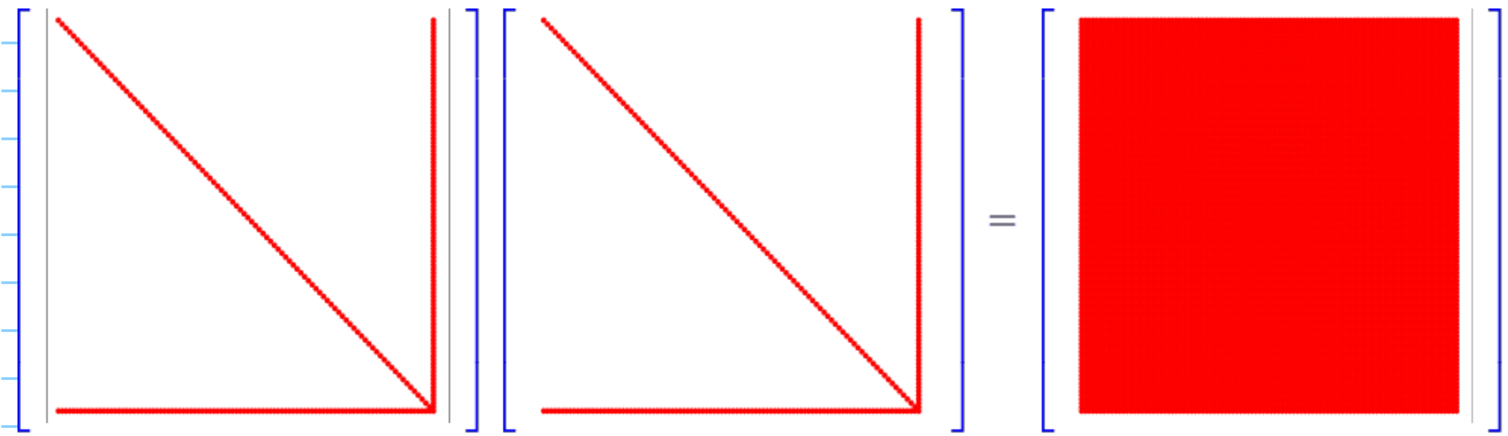
↑ linear in m (for n fixed/small)

NE : "numerically problematic"

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ \delta & 0 \\ 0 & \delta \end{bmatrix} \quad \Rightarrow \quad \mathbf{A}^\top \mathbf{A} = \begin{bmatrix} 1+\delta^2 & 1 \\ 1 & 1+\delta^2 \end{bmatrix}$$

Exp. 1.5.35: If $\delta \approx \sqrt{\mathrm{EPS}}$, then $1+\delta^2 \stackrel{\cdot}{=} 1$ in $\mathbb{M}$. Hence the computed $\mathbf{A}^\top\mathbf{A}$ will fail to be regular, though $\operatorname{rank}(\mathbf{A}) = 2$, $\operatorname{cond}_2(\mathbf{A}) \approx \sqrt{\mathrm{EPS}}$.

NE : Loss of sparsity (A sparse ⇒ A^TA need not be sparse)



Remedy : If A sparse, m > n both large ⇒ use extended normal equation

$$\underbrace{\mathbf{A}^H\mathbf{A}\mathbf{x} = \mathbf{A}^H\mathbf{b}}_{NE} \quad \Leftrightarrow \quad \mathbf{B}\begin{bmatrix}\mathbf{r} \\ \mathbf{x}\end{bmatrix} := \begin{bmatrix} -\mathbf{I} & \mathbf{A} \\ \mathbf{A}^H & 0 \end{bmatrix} \begin{bmatrix}\mathbf{r} \\ \mathbf{x}\end{bmatrix} = \begin{bmatrix}\mathbf{b} \\ 0\end{bmatrix}. \tag{3.2.8}$$

Additional unknown $\underline{r} = b - A\underline{x}$

Even more extended normal equations :

$$\mathbf{A}^H\mathbf{A}\mathbf{x} = \mathbf{A}^H\mathbf{b} \quad \Leftrightarrow \quad \mathbf{B}_\alpha\begin{bmatrix}\mathbf{r} \\ \mathbf{x}\end{bmatrix} := \begin{bmatrix} -\alpha\mathbf{I} & \mathbf{A} \\ \mathbf{A}^H & 0 \end{bmatrix} \begin{bmatrix}\mathbf{r} \\ \mathbf{x}\end{bmatrix} = \begin{bmatrix}\mathbf{b} \\ 0\end{bmatrix}. \tag{3.2.9}$$

3.3. Orthogonal transformation methods

3.3.1 Idea :

$$\text{transform } \underbrace{A\underline{x} = \underline{b} \longrightarrow \overbrace{\tilde{A}\underline{x} = \tilde{b}}^{\text{should be simple}^*}}_{}$$

"equivalent"*: $\operatorname{Lsq}(A, b) = \operatorname{Lsq}(\hat{A}, \hat{b})$

* = triangular

regular, if $\operatorname{rank}(A) = n$

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

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

6 *

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

$$\|\mathbf{T}\mathbf{y}\|_2 = \|\mathbf{y}\|_2 \quad \forall \mathbf{y} \in \mathbb{R}^m , \tag{3.3.3}$$

then
$$\operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{y} - \mathbf{b}\|_2 = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^n} \left\|\widetilde{\mathbf{A}}\mathbf{y} - \widetilde{\mathbf{b}}\right\|_2 ,$$

where $\widetilde{\mathbf{A}} = \mathbf{T}\mathbf{A}$ and $\widetilde{\mathbf{b}} = \mathbf{T}\mathbf{b}$.

3.3.2. Orthogonal Matrices

Definition 3.3.4. Unitary and orthogonal matrices $\rightarrow$ [?, 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 $\mathbf{Q}^{-1} = \mathbf{Q}^T$.

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:

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

3.3.3. QR-Decomposition

Recall: Gram-Schmidt orthog. of $\{a^1, .., a^k\} \subset \mathbb{R}^n$

```
q^1 := a^1 / ||a^1||_2          % 1st output vector
for j = 2,...,k do
  {  % Orthogonal projection
     q^j := a^j
     for ℓ = 1,2,...,j-1 do
        { q^j ← q^j − ⟨a^j, q^ℓ⟩ q^ℓ }
     if ( q^j = 0 ) then STOP
     else { q^j ← q^j / ||q^j||_2 }
  }
```
(GS)

Theorem 3.3.7. Span property of G.S. vectors

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

$$\operatorname{Span}\{\mathbf{q}^1, \dots, \mathbf{q}^\ell\} = \operatorname{Span}\{\mathbf{a}^1, \dots, \mathbf{a}^\ell\} , \tag{1.5.2}$$

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

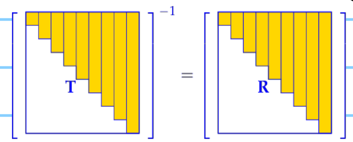
$$\begin{aligned}
\mathbf{q}_1 &= t_{11}\mathbf{a}_1 \\
\mathbf{q}_2 &= t_{12}\mathbf{a}_1 + t_{22}\mathbf{a}_2 \\
\mathbf{q}_3 &= t_{13}\mathbf{a}_1 + t_{23}\mathbf{a}_2 + t_{33}\mathbf{a}_3 \\
&\vdots \\
\mathbf{q}_k &= t_{1n}\mathbf{a}_1 + t_{2n}\mathbf{a}_2 + \cdots + t_{kk}\mathbf{a}_k .
\end{aligned}$$

$$(T)_{ij} := \begin{cases} 0 & \text{else} \\ t_{ij}, & \text{if } j \geq i \end{cases}$$

$\blacktriangleright$ $\exists \mathbf{T} \in \mathbb{R}^{n,n}$ upper triangular: $\mathbf{Q} = \mathbf{A}\mathbf{T}$, (3.3.8)

$Q = [q^1 .. , q^k]$, $A = [a^1 .. , a^k]$

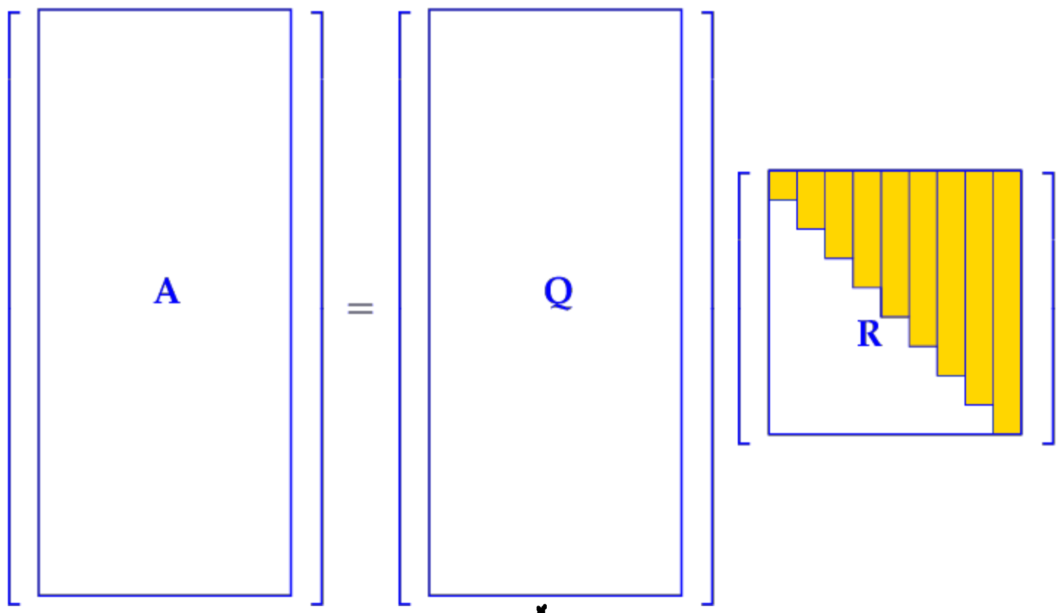
$$\underbrace{\underset{\substack{\uparrow \\ \text{orthonormal} \\ \text{columns}}}{Q} = A \underset{\substack{\uparrow \\ \text{full rank } n}}{T}}_{\text{rank}(T) = n \,:\, T \text{ regular}} \Rightarrow A = Q \underset{\substack{\uparrow \\ \text{again upper triangular}}}{T^{-1}}$$



$R := T^{-1}$

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

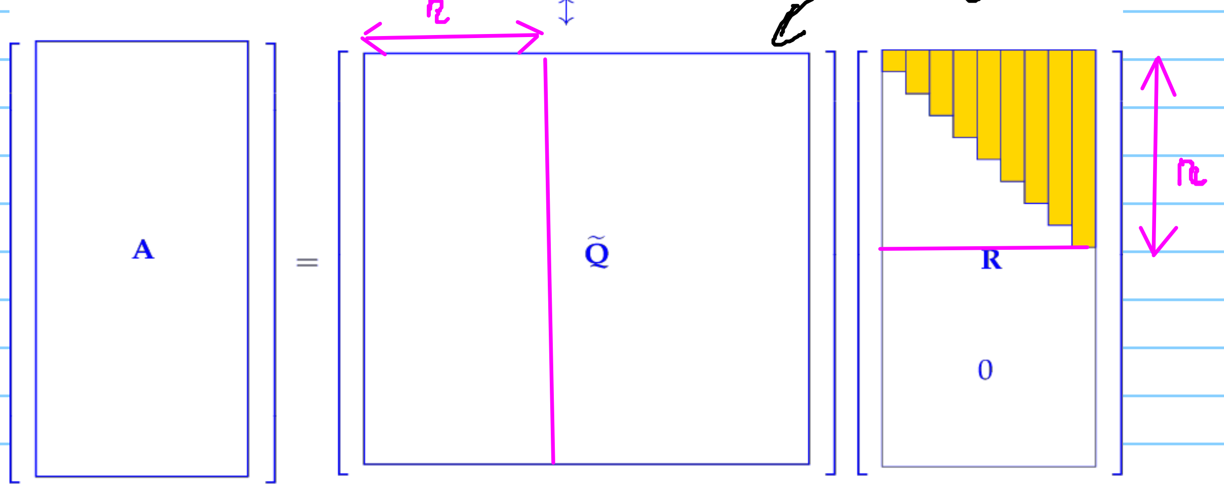
$\mathbf{A} = \mathbf{QR} \quad \leftrightarrow$



can be extended to ONB of $\mathbb{R}^m$ $\Leftarrow$ orthonormal columns

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

orthogonal !



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

Gram - Schmidt shows

Theorem 3.3.9. QR-decomposition $\rightarrow$ **[?, 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* $\mathbf{Q}_0 \in \mathbb{K}^{n,k}$ *that satisfies* $\mathbf{Q}_0^H \mathbf{Q}_0 = \mathbf{I}_k$*, and a unique upper triangular Matrix* $\mathbf{R}_0 \in \mathbb{K}^{k,k}$ *with* $(\mathbf{R})_{i,i} > 0$, $i \in \{1, \dots, k\}$*, such that*

$$\mathbf{A} = \mathbf{Q}_0 \cdot \mathbf{R}_0 \qquad \text{("economical" QR-decomposition)},$$

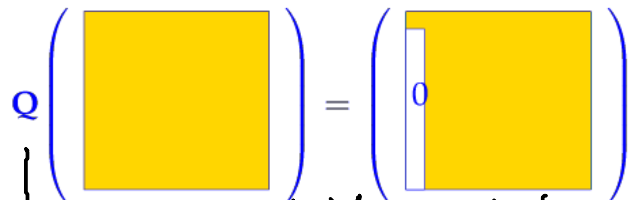
(ii) *a unitary Matrix* $\mathbf{Q} \in \mathbb{K}^{n,n}$ *and a unique upper triangular* $\mathbf{R} \in \mathbb{K}^{n,k}$ *with* $(\mathbf{R})_{i,i} > 0$, $i \in \{1, \dots, n\}$*, such that*

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

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

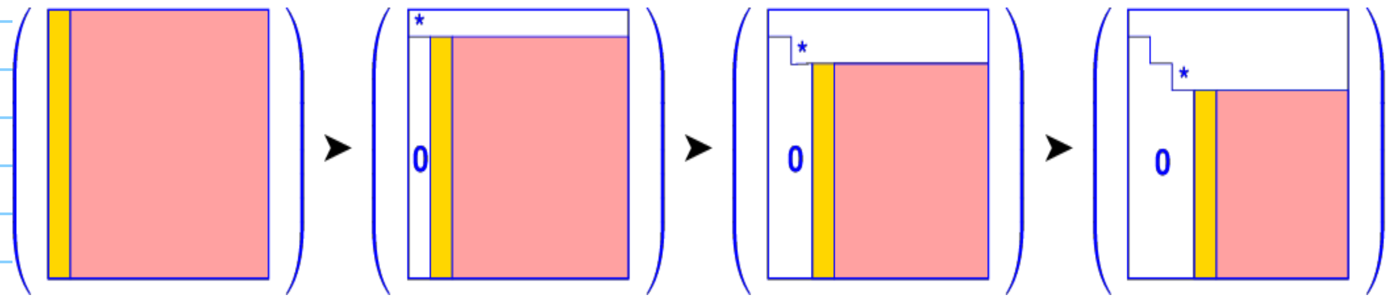
3.3.3.2 Computation of QR-decomposition

Idea: find simple unitary/orthogonal (row) transformations rendering certain matrix elements zero: $\rightarrow$ compare GE



with $\mathbf{Q}^H = \mathbf{Q}^{-1}$.

↳ ① Householder reflection, ② Givens rotation



⑧ Givens rotations

2D

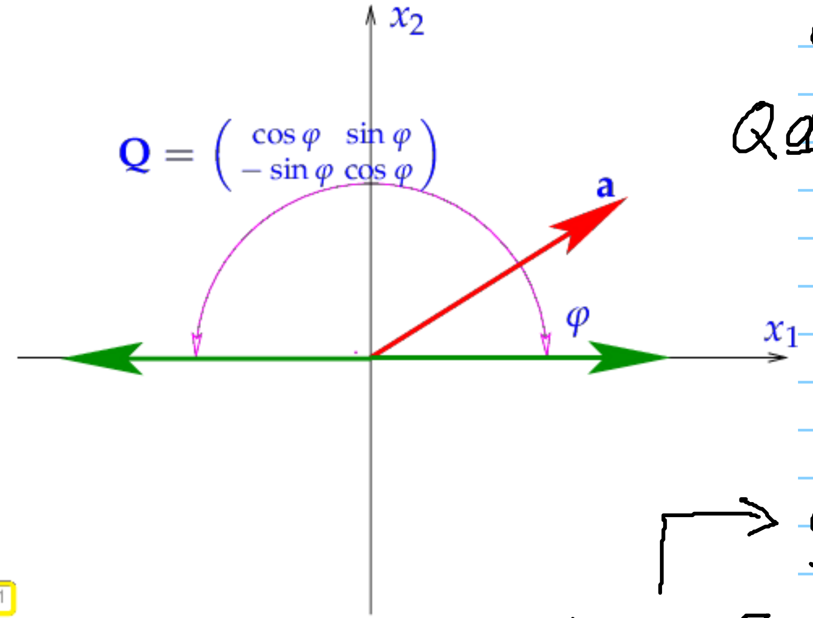


Fig. 101

rotations turning **a** onto x_1-axis. [2 options]

Goal :

$Q a$ has 2nd component $= 0$

→ choose the right one for numerical stability

$\gamma \sim \cos\varphi$
$\sigma \sim \sin\varphi$

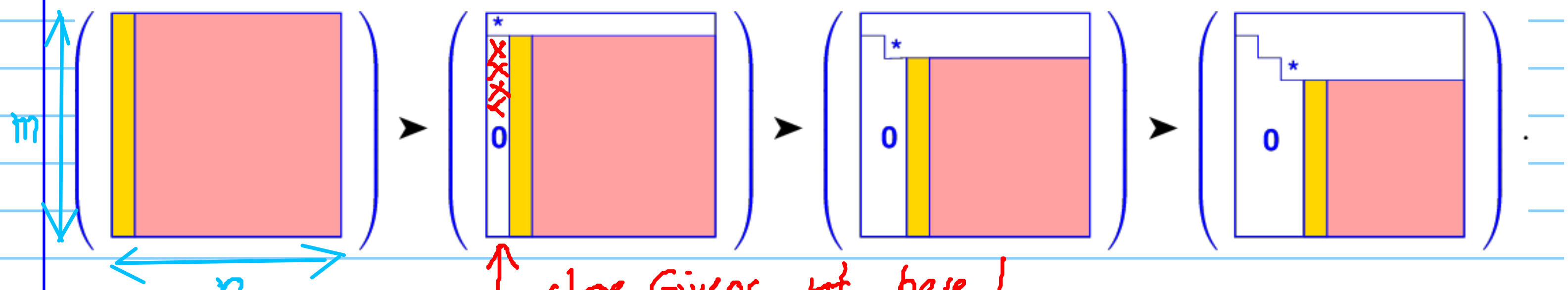
$$\mathbf{G}_{1k}(a_1,a_k)\mathbf{a} := \begin{pmatrix} \gamma & \cdots & \sigma & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ -\sigma & \cdots & \gamma & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_k \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} a_1^{(1)} \\ \vdots \\ 0 \\ \vdots \\ a_n \end{pmatrix}, \text{ if } \begin{aligned} \gamma &= \frac{a_1}{\sqrt{|a_1|^2+|a_k|^2}}, \\ \sigma &= \frac{a_k}{\sqrt{|a_1|^2+|a_k|^2}}. \end{aligned} \tag{3.3.20}$$

↑ orthogonal

▷ $G_{\ell k}(a_\ell, a_k)\cdot$ changes only the components ℓ & k of the vector

$$\begin{pmatrix} a_1 \\ \vdots \\ \vdots \\ \vdots \\ a_n \end{pmatrix} \xrightarrow{\mathbf{G}_{12}(a_1,a_2)} \begin{pmatrix} a_1^{(1)} \\ 0 \\ a_3 \\ \vdots \\ a_n \end{pmatrix} \xrightarrow{\mathbf{G}_{13}(a_1^{(1)},a_3)} \begin{pmatrix} a_1^{(2)} \\ 0 \\ 0 \\ a_4 \\ \vdots \\ a_n \end{pmatrix} \xrightarrow{\mathbf{G}_{14}(a_1^{(2)},a_4)} \cdots \xrightarrow{\mathbf{G}_{1n}(a_1^{(n-2)},a_n)} \begin{pmatrix} a_1^{(n-1)} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \tag{3.3.22}$$

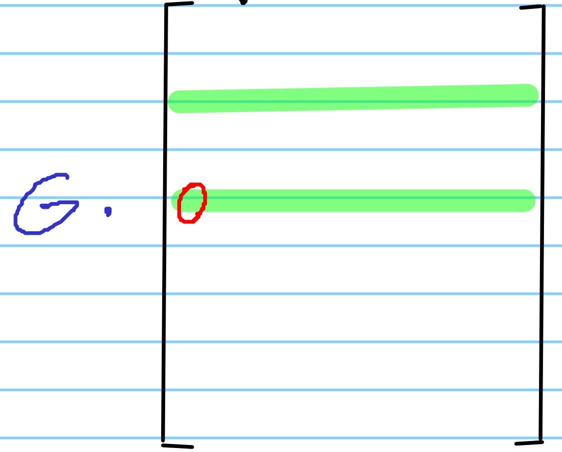
Recall : Product of orth. matrices stays orthogonal



↑ store Givens rot here !

Cost for transforming $m \times n$ matrix $\xrightarrow{\text{Givens}}$ R

$$= O(\underset{\substack{\uparrow \\ \text{from left} \to \text{right}}}{n} \cdot \underset{\substack{\uparrow \\ \text{col} \to 0}}{m} \cdot \underset{\substack{\uparrow \\ \text{single Givens rot.}}}{n}) = O(n^2 m)$$



← replace with lin. comb.

Storing Givens rotations:

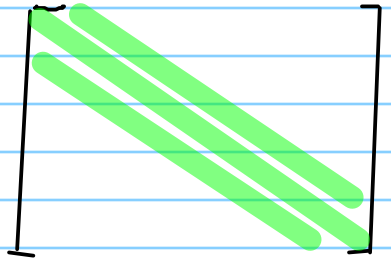
! Each $G_{\ell k}(\ldots)$ can be encoded by a single number ! $\rightarrow$ angle

for $\mathbf{G} = \begin{pmatrix} \gamma & \sigma \\ -\sigma & \gamma \end{pmatrix} \Rightarrow$ store $\rho := \begin{cases} 1 & \text{, if } \gamma = 0\,, \\ \frac{1}{2}\,\mathrm{sign}(\gamma)\sigma & \text{, if } |\sigma| < |\gamma|\,, \\ 2\,\mathrm{sign}(\sigma)/\gamma & \text{, if } |\sigma| \geq |\gamma|\,, \end{cases}$

which means $\begin{cases} \rho = 1 & \Rightarrow \quad \gamma = 0\,,\quad \sigma = 1 \\ |\rho| < 1 & \Rightarrow \quad \sigma = 2\rho\,,\quad \gamma = \sqrt{1-\sigma^2} \\ |\rho| > 1 & \Rightarrow \quad \gamma = 2/\rho\,,\quad \sigma = \sqrt{1-\gamma^2}\,. \end{cases}$

Remark: QR-decomposition of banded matrices

Example: tridiagonal matrix



$\in \mathbb{R}^{m,m}$

$$\begin{pmatrix} * & * & 0 & 0 & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & * & * & 0 & 0 & 0 & 0 \\ 0 & 0 & * & * & * & 0 & 0 & 0 \\ 0 & 0 & 0 & * & * & * & 0 & 0 \\ 0 & 0 & 0 & 0 & * & * & * & 0 \\ 0 & 0 & 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & * & * \end{pmatrix} \xrightarrow{\mathbf{G}_{12}} \begin{pmatrix} * & * & * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & * & * & 0 & 0 & 0 & 0 \\ 0 & 0 & * & * & * & 0 & 0 & 0 \\ 0 & 0 & 0 & * & * & * & 0 & 0 \\ 0 & 0 & 0 & 0 & * & * & * & 0 \\ 0 & 0 & 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & * & * \end{pmatrix} \xrightarrow{\mathbf{G}_{23}\cdots\mathbf{G}_{n-1,n}} \begin{pmatrix} * & * & * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & * & * & 0 & 0 & 0 & 0 \\ 0 & 0 & * & * & * & 0 & 0 & 0 \\ 0 & 0 & 0 & * & * & * & 0 & 0 \\ 0 & 0 & 0 & 0 & * & * & * & 0 \\ 0 & 0 & 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & * \end{pmatrix}$$

Bandwidth 3 $\longrightarrow$ Bandwidth 3

Cost $= O(n)$ for $n \times n$ matrix !

QR-dec. by Givens preserves bandwidth

3.3.3.4 QR-decomposition in Eigen

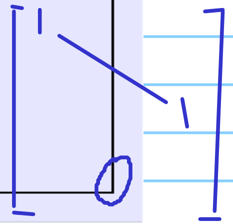
C++-code 3.3.34: QR-decompositions in EIGEN

```
# 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);
}
```

$\longrightarrow$ expensive

Cost $= O(mn^2)$: linear in m

$\in \mathbb{R}^{m,n}$



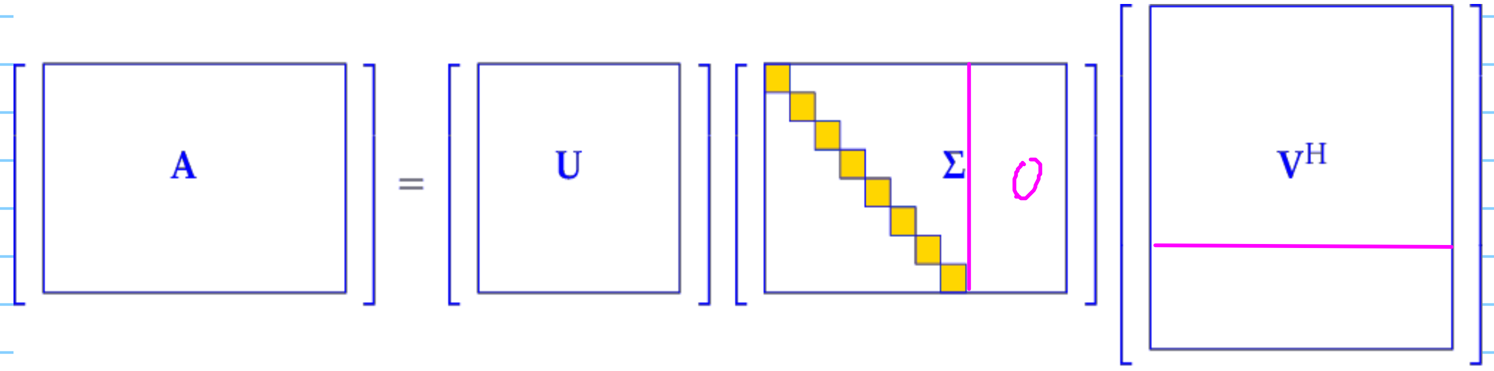
Normal equations vs. orthogonal transformations method

Superior numerical stability (→ 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 (→ Notion 2.7.1) and m, n big.



3.4. Singular Value Decomposition (SVD)

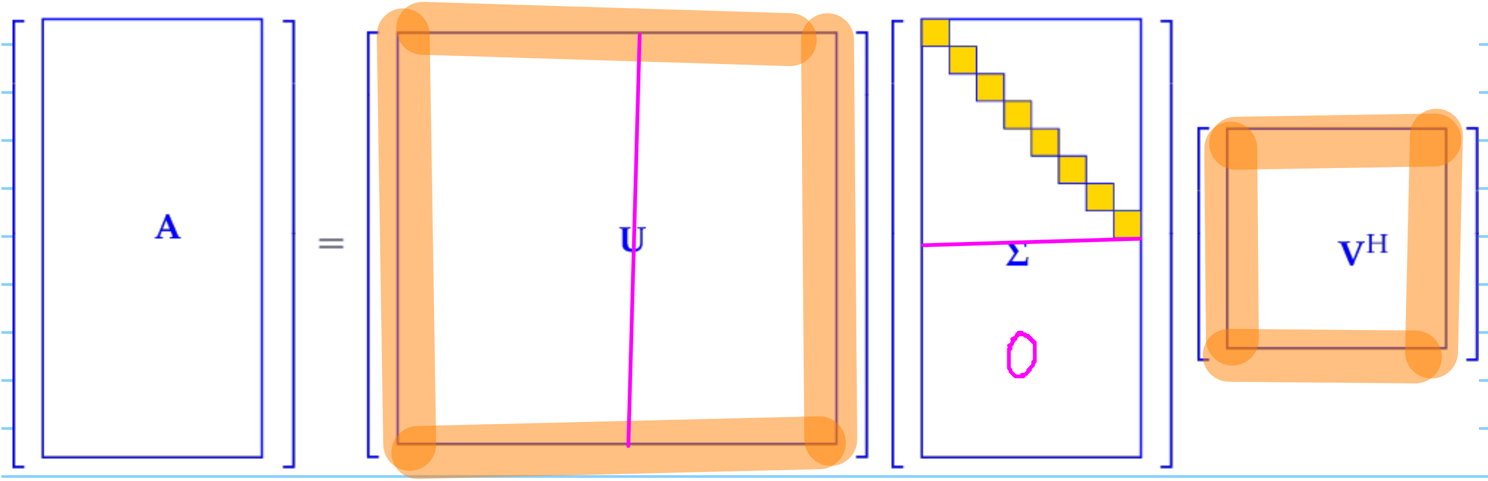
3.4.1 Definition

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

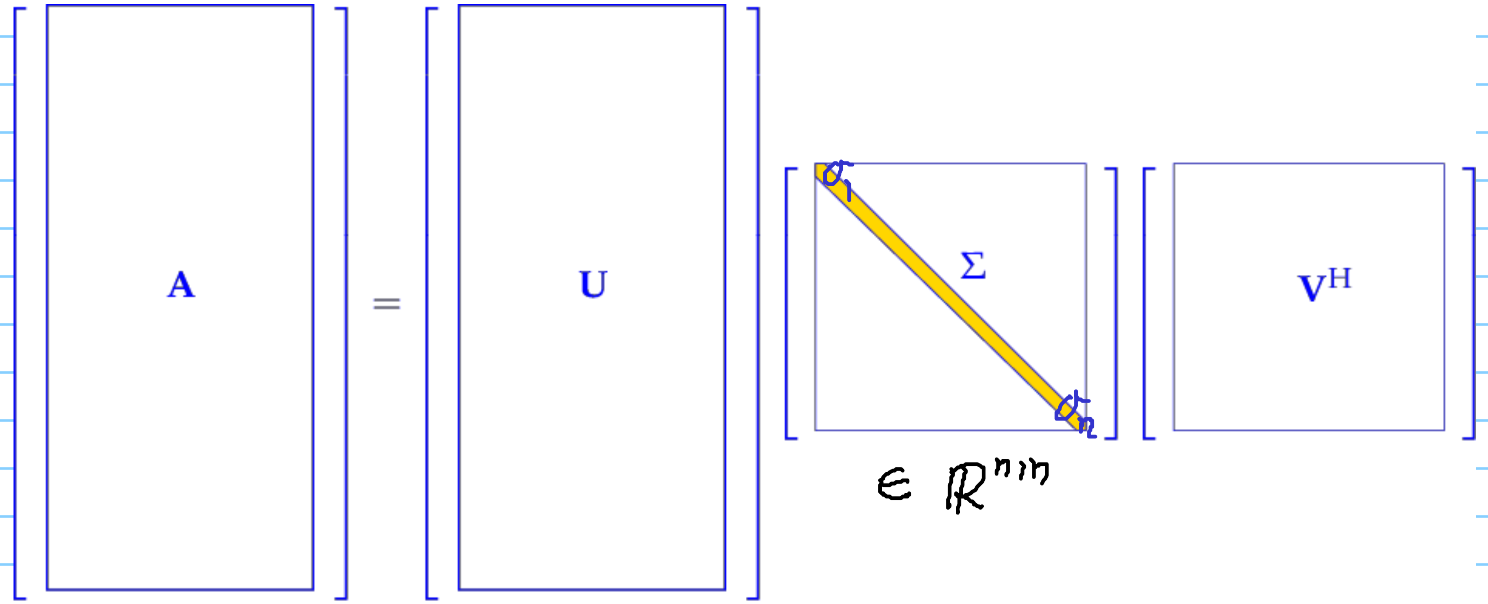
(*) *For any* $\mathbf{A} \in \mathbb{K}^{m,n}$ *there are unitary matrices* $\mathbf{U} \in \mathbb{K}^{m,m}$, $\mathbf{V} \in \mathbb{K}^{n,n}$ *and a (generalized) diagonal matrix* $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{m,n}$, $p := \min\{m,n\}$, $\underbrace{\sigma_1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

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

singular values



▷ Economical SVD : $m \geq n$



$$\Updownarrow$$

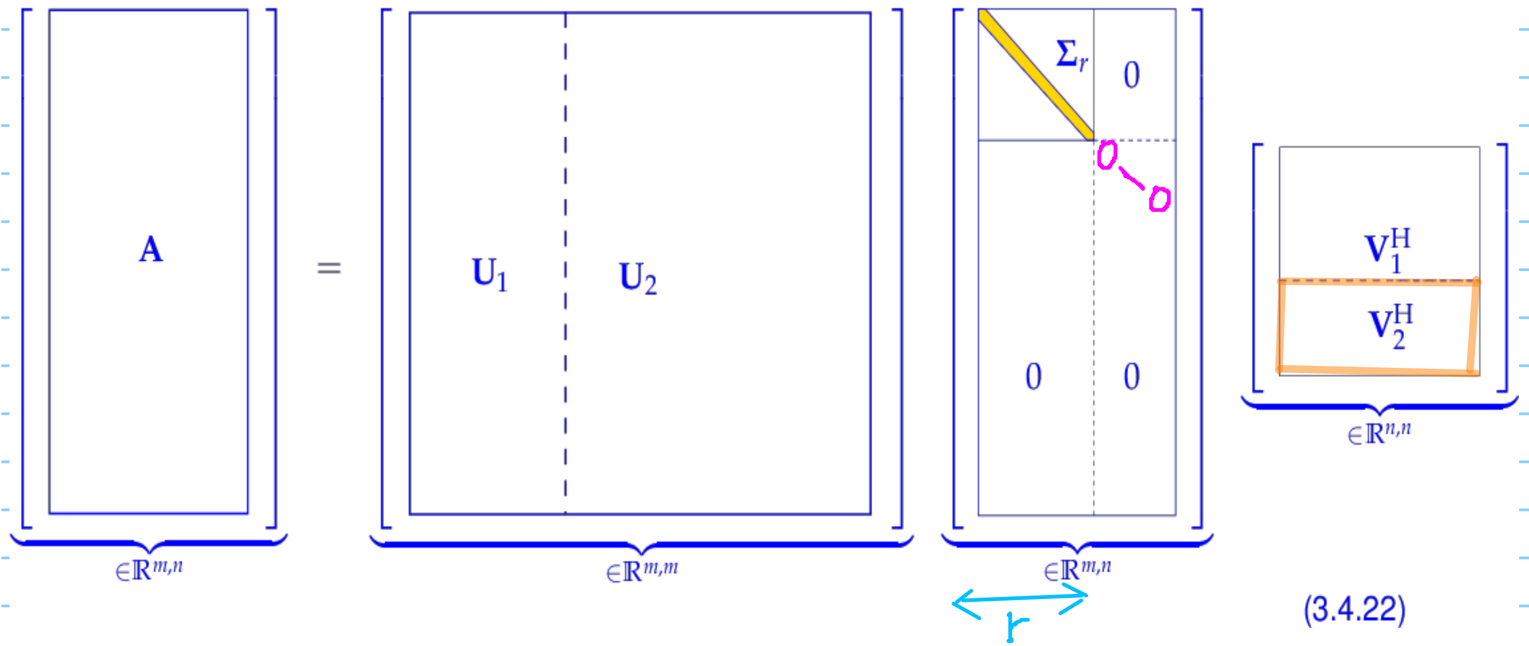
$$A = \sum_{\ell=1}^{n} \sigma_\ell \underbrace{(U)_{:,\ell} ((V)_{:,\ell})^H}_{\in \mathbb{R}^{m,n}} \qquad (\diamond)$$

(11)

If $\underbrace{\sigma_{r+1} = \dots = \sigma_p = 0}$ $\Longrightarrow$ $\mathrm{Rank}(A) = \mathrm{Rank}(\Sigma) = r$

$m \geq n$:

$$\mathbf{A} = [\,\mathbf{U}_1 \quad \mathbf{U}_2\,] \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{bmatrix}$$



(3.4.22)

$A\,(V)_{:,l} = U \Sigma e_l = 0$ if $l > r$

$[l > r]$ $\qquad (V)_{:,l} \in \mathcal{N}(A)$ for $l > r$

$V^T (V)_{:,l} = e_l$ due to orthogonality of V

l-th unit vector $\in \mathbb{R}^n$

Lemma 3.4.11. SVD and rank of a matrix $\rightarrow$ [?, Cor. 9.7]

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

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

3.4.2 SVD in Eigen

C++-code 3.4.13: Computing SVDs in EIGEN

```
# include <Eigen/SVD>

// Computation of (full) SVD A = UΣV^H → Thm. 3.4.1
// SVD factors are returned as dense matrices in natural order
std::tuple<MatrixXd,MatrixXd,MatrixXd> svd_full(const MatrixXd& A) {
  Eigen::JacobiSVD<MatrixXd> svd(A, Eigen::ComputeFullU |
      Eigen::ComputeFullV);
  MatrixXd U = svd.matrixU(); // get unitary (square) matrix U
  MatrixXd V = svd.matrixV(); // get unitary (square) matrix V
  VectorXd sv = svd.singularValues(); // get singular values as vector
  MatrixXd Sigma = MatrixXd::Zero(A.rows(), A.cols());
  const unsigned p = sv.size(); // no. of singular values
  Sigma.block(0,0,p,p) = sv.asDiagonal(); // set diagonal block of Σ
  return std::tuple<MatrixXd,MatrixXd,MatrixXd>(U,Sigma,V);
}

// Computation of economical (thin) SVD A = UΣV^H, see (3.4.4)
// SVD factors are returned as dense matrices in natural order
std::tuple<MatrixXd,MatrixXd,MatrixXd> svd_eco(const MatrixXd& A) {
  Eigen::JacobiSVD<MatrixXd> svd(A, Eigen::ComputeThinU |
      Eigen::ComputeThinV);
  MatrixXd U = svd.matrixU(); // get unitary (square) matrix U
  MatrixXd V = svd.matrixV(); // get unitary (square) matrix V
  VectorXd sv = svd.singularValues(); // get singular values as vector
  MatrixXd Sigma = sv.asDiagonal(); // build diagonal matrix Σ
  return std::tuple<MatrixXd,MatrixXd,MatrixXd>(U,Sigma,V);
}
```

Remark: "Numerical rank"

$$r = \max_j \frac{\sigma_j}{\sigma_1} \geq TOL$$

$$\boxed{\text{Cost (eco SVD)} = O(\min\{m,n\}^2 \cdot \max\{m,n\}) \quad \to \text{ linear in "big dimension"}}$$

3.4.4. SVD-based Optimization & Approximation

3.4.4.1 Norm constrained Extrema

given $A \in \mathbb{R}^{m,n}$, $m \geq n$, find $x \in \mathbb{K}^n$, $\|x\|_2 = 1$, $\|Ax\|_2 \to \min$. (3.4.31)

↳ eco SVD : $A = U \Sigma V^T$

$$\|Ax\|_2^2 = \|U \Sigma V^T x\|_2^2 = \|\Sigma y\|_2^2 = \underbrace{\sum_{\ell=1}^{n} \sigma_\ell^2 y_\ell^2 \to \min \text{ with } \|y\|_2 = 1}$$

orth. col. of U !

$y = V^T x$

$$x = V e_n = (V)_{:,n} \Leftarrow \text{Minimizer } y = e_n$$

Minimum $= \sigma_n$

Ex (Computational Geometry)

Fitting of a hyperplane

Goal: given the points $y_1, \ldots, y_m$, $m > d$, find $\mathcal{H} \leftrightarrow \{c \in \mathbb{R}, \mathbf{n} \in \mathbb{R}^d, \|\mathbf{n}\|_2 = 1\}$, such that

$y_i \in \mathbb{R}^d$

Hesse normal form

$$x = \begin{bmatrix} c \\ n \end{bmatrix} \quad \|Ax\|_2^2 = \sum_{j=1}^{m} \operatorname{dist}(\mathcal{H}, \mathbf{y}_j)^2 = \sum_{j=1}^{m} |c + \mathbf{n}^\top \mathbf{y}_j|^2 \to \min . \quad (3.4.38)$$

$$\mathcal{H} := \{x \in \mathbb{R}^d : n \cdot x + c = 0\}$$

$$(3.4.38) \Leftrightarrow \left\| \underbrace{\begin{bmatrix} 1 & y_{1,1} & \cdots & y_{1,d} \\ 1 & y_{2,1} & \cdots & y_{2,d} \\ \vdots & \vdots & & \vdots \\ 1 & y_{m,1} & \cdots & y_{m,d} \end{bmatrix}}_{=:\mathbf{A}} \begin{bmatrix} c \\ n_1 \\ \vdots \\ n_d \end{bmatrix} \right\|_2 \to \min \quad \text{under constraint } \|\mathbf{n}\|_2 = 1 .$$

Trick : $A = QR$: $\|Ax\|_2 \to \min \Leftrightarrow \|Rx\|_2 \to \min$

$$\|\mathbf{Ax}\|_2 \to \min \Leftrightarrow \|\mathbf{Rx}\|_2 = \left\| \begin{bmatrix} r_{11} & r_{12} & \cdots & \cdots & r_{1,d+1} \\ 0 & r_{22} & \cdots & \cdots & r_{2,d+1} \\ \vdots & & \ddots & & \vdots \\ 0 & & & & r_{d+1,d+1} \\ 0 & \cdots & & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & & \cdots & 0 \end{bmatrix} \begin{bmatrix} c \\ n_1 \\ \vdots \\ \vdots \\ n_d \end{bmatrix} \right\|_2 \to \min . \quad (3.4.39)$$

$c \to$ "free component"

(13)

Idea: C: $r_{11} c + r_{12} n_1 + \dots + r_{1,d+1} n_d = 0$

$$\left\| \begin{bmatrix} r_{22} & r_{23} & \cdots & \cdots & r_{2,d+1} \\ 0 & r_{33} & \cdots & \cdots & r_{3,d+1} \\ \vdots & & \ddots & & \vdots \\ 0 & & & & r_{d+1,d+1} \end{bmatrix} \begin{bmatrix} n_1 \\ \vdots \\ \vdots \\ n_d \end{bmatrix} \right\|_2 \to \min \quad , \quad \|\mathbf{n}\|_2 = 1 \,. \tag{3.4.40}$$

Now "standard form" (3.4.31)

3.4.4.2 Best low-rank Approximation

→ (Lossy) Matrix compression

$A \in \mathbb{R}^{m,n}$ → $m \cdot n$ memory

$\mathrm{Rank}(A) = p \ll \min\{m,n\} \Rightarrow p(m+n)$ memory

$\left[\ (\Diamond) \Rightarrow A = \sum_{\ell=1}^{p} u_\ell v_\ell^T \ \right]$

Theorem 3.4.48. best low rank approximation → [?, Thm. 11.6]

Let $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^H$ *be the SVD of* $\mathbf{A} \in \mathbb{K}^{m.n}$ (→ *Thm. 3.4.1*). *For* $1 \le k \le \mathrm{rank}(\mathbf{A})$ *set* $\mathbf{U}_k := [\mathbf{u}_{:,1}, \dots, \mathbf{u}_{:,k}] \in \mathbb{K}^{m,k}$, $\mathbf{V}_k := [\mathbf{v}_{:,1}, \dots, \mathbf{v}_{:,k}] \in \mathbb{K}^{n,k}$, $\Sigma_k := \mathrm{diag}(\sigma_1, \dots, \sigma_k) \in \mathbb{K}^{k,k}$. *Then, for* $\|\cdot\| = \|\cdot\|_F$ *and* $\|\cdot\| = \|\cdot\|_2$, *holds true*

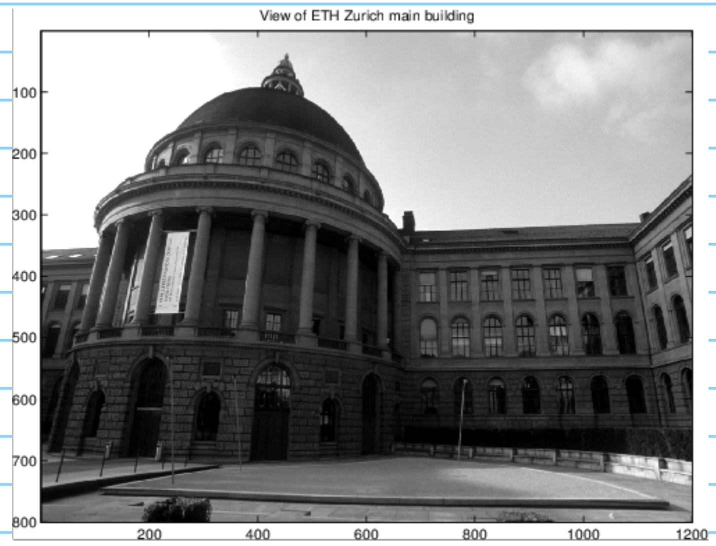
$$\left\|\mathbf{A} - \underbrace{\mathbf{U}_k\Sigma_k\mathbf{V}_k^H}\right\| \le \|\mathbf{A} - \mathbf{F}\| \quad \forall \mathbf{F} \in \mathcal{R}_k(m,n) = \{M \in \mathbb{R}^{m,n} : \mathrm{rank}(M) = k\}$$

$$= \sum_{\ell=1}^{k} \sigma_k \, (\mathcal{U})_{:,\ell} \big((V)_{:,\ell}\big)^T$$

Frobenius norm: $\|A\|_F^2 = \sum_{i,j} (A)_{i,j}^2$

$\|A - U_k \Sigma_k V_k^T\|_2 = \sigma_{k+1} \Rightarrow$ Error estimate

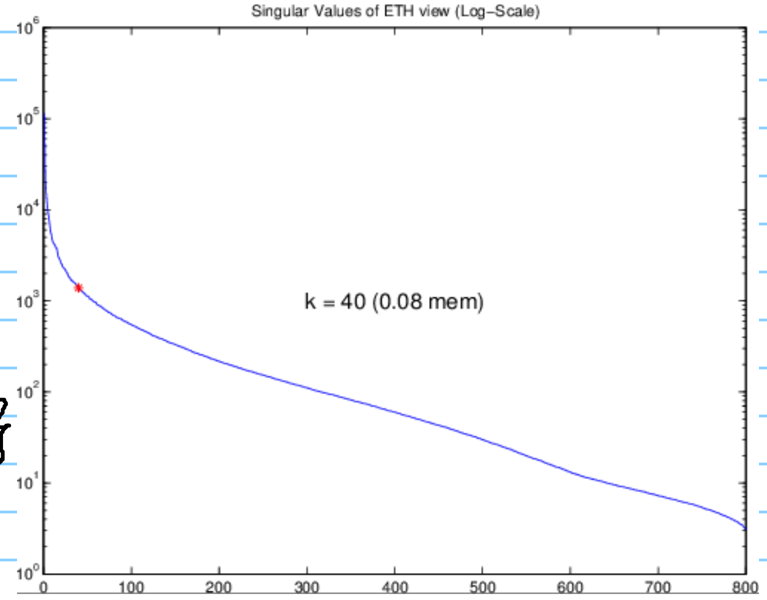
Ex: Image compression



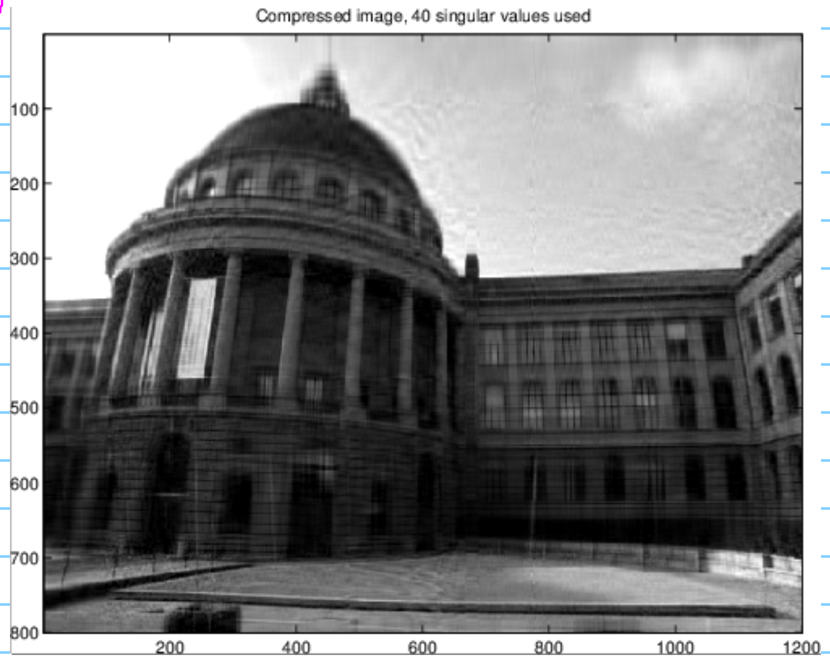
$9.6 \cdot 10^5$ pixel

1200×800

$\triangleq$ a matrix



▷ roughly exponential decay



$K = 40$

$\sim 1\%$ mem.

3.4.4.3 Principal Component Analysis (PCA)

Trend detection

data classification

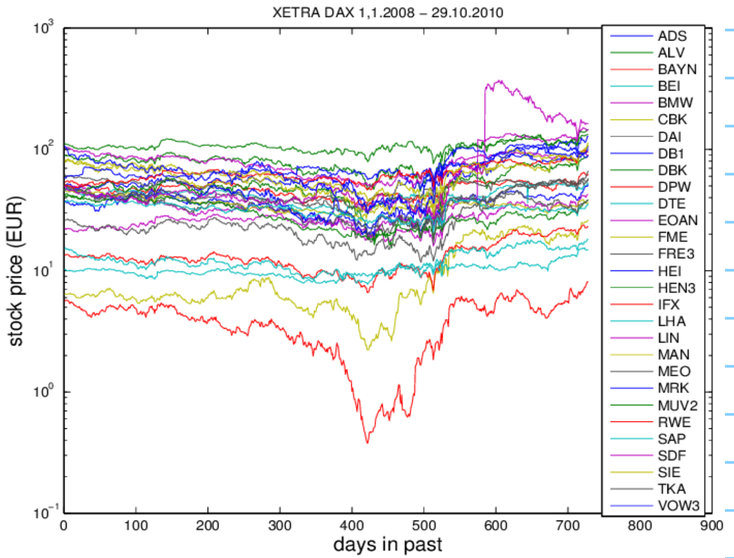


Fig. 106

$\leftarrow$ Time series $a_k \in \mathbb{R}^m$, $k = 1, \dots, n$

Task: Find "trends"

Find a few vectors $\{u_1, \dots, u_p\}$ such that $p \ll n$

$a_k \in \operatorname{Span}\{u_1, \dots, u_p\}$ $(*)$

$(*)$ not realistic, because of "random" perturbations

Find $\{u_1, \dots, u_p\} \subset \mathbb{R}^m$: $a_k \in \operatorname{Span}\{u_1, \dots, u_p\} +$ "small pert."

$A = [a_1, \dots, a_n] \in \mathbb{R}^{m,n}$ & its SVD: $A = U \Sigma V^T$

$(U = [u_1, \dots, u_m], V = [v_1, \dots, v_n])$

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \underbrace{\sigma_{p+1} u_{p+1} v_{p+1}^T + \dots + \sigma_n u_n v_n^T}_{\|\cdot\| = \sigma_{p+1} \quad \text{"small"}}$$

Assume: σ_j decay rapidly

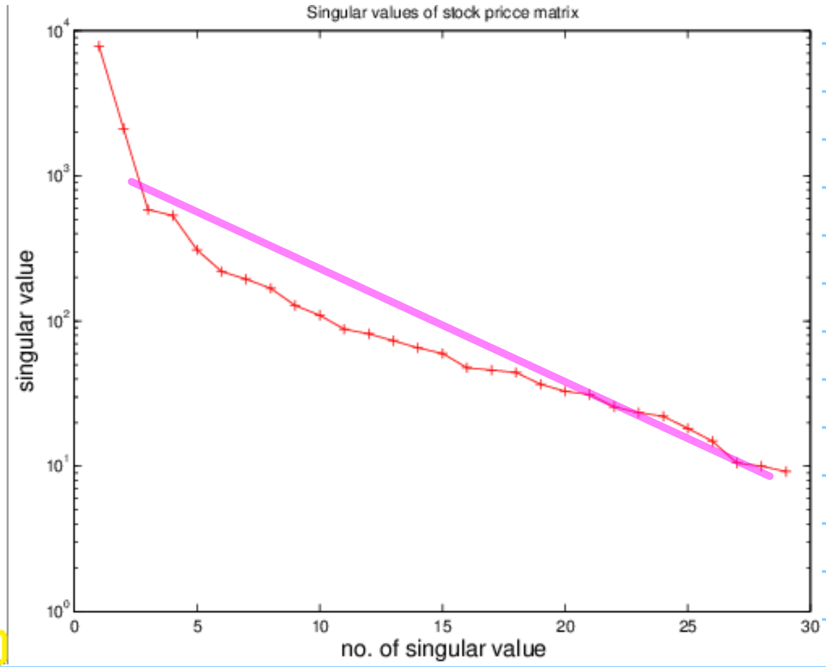


Fig. 121

$\triangleleft$ Stock data

σ_j decay exponentially

$$a_k = \sum_{j=1}^{p} u_j \underbrace{\left(\sigma_j (v_j)_k\right)}_{} + \text{"small perturbation"}$$

↳ weight of j-th trend vector in k-th time series

Classification:

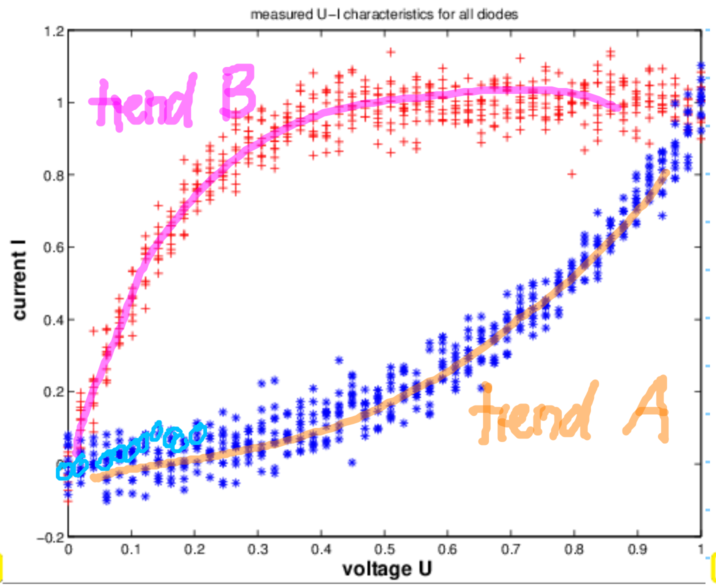


Fig. 109

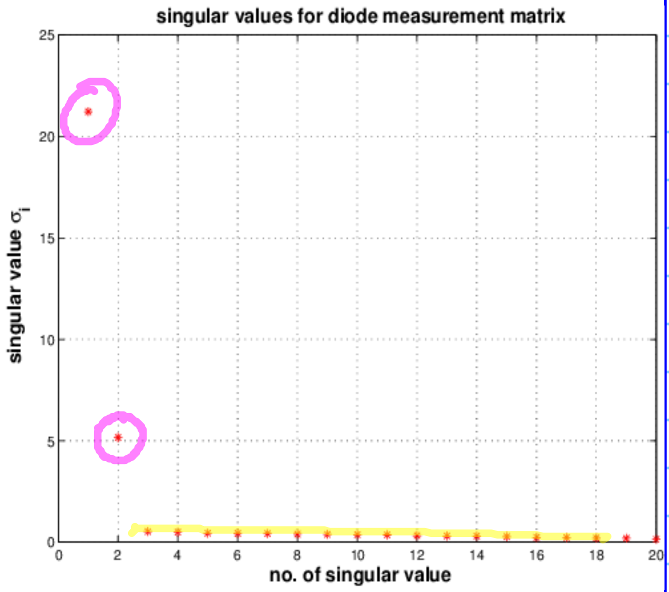


Fig. 113

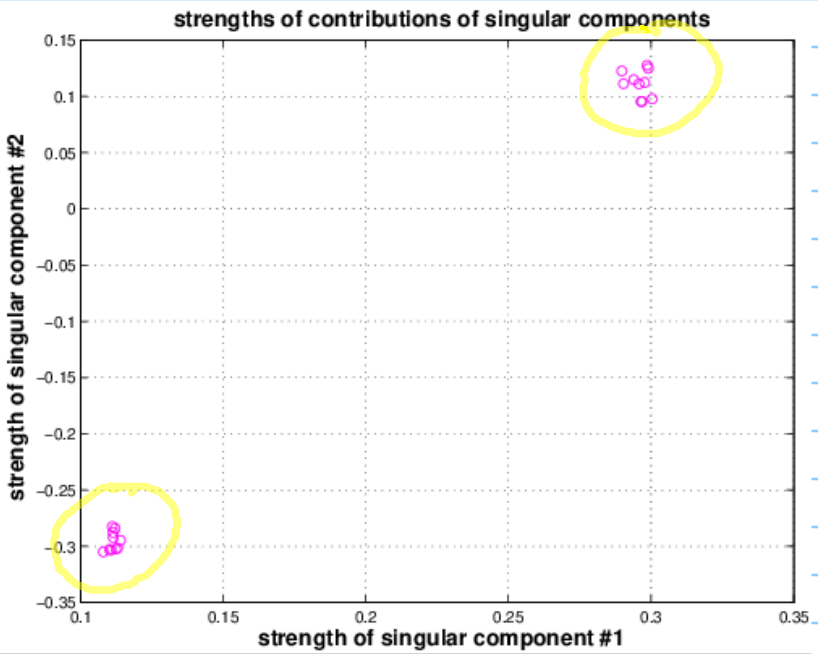


Fig. 114

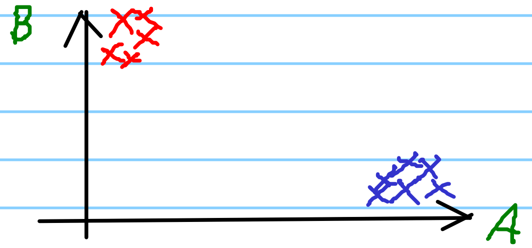
Next step: cluster identification

→ Classification based on strength of trends in a_k:

↳ $\sim \left(\sigma_j (v_j)_k\right)$

A B

$$p = 2 \;(2 \text{ trends}) : \underbrace{\left[\left(\sigma_1 (v_1)_k\right), \left(\sigma_2 (v_2)_k\right)\right]_{k=1}^{n}}_{\in \mathbb{R}^2 \,:\, \text{point-coordinates}}$$



3.5. Total least squares

$A \in \mathbb{R}^{m,n}$, $m > n$: $\qquad A\underline{x} = \underline{b}$

NEW : A, b measured

Task : Find the "nearest" solvable LSE

Total least squares problem:

Given: $\mathbf{A} \in \mathbb{R}^{m,n}$, $m > n$, $\operatorname{rank}(\mathbf{A}) = n$, $\mathbf{b} \in \mathbb{R}^m$,

find: $\widehat{\mathbf{A}} \in \mathbb{R}^{m,n}$, $\widehat{\mathbf{b}} \in \mathbb{R}^m$ with

$$\left\| [\mathbf{A}\ \ \mathbf{b}] - [\widehat{\mathbf{A}}\ \ \widehat{\mathbf{b}}] \right\|_F \to \min \quad , \quad \underbrace{\widehat{\mathbf{b}} \in \mathcal{R}(\widehat{\mathbf{A}})}_{\Rightarrow\ \operatorname{rank}([\hat{A}\ \hat{b}]) = n} . \tag{3.5.1}$$

Frobenius norm

▷ $[\hat{A}\ \hat{b}] \in \mathbb{R}^{m,n+1}$ is the rank-n best approximation of $[A\ \underline{b}]$

▷ SVD of $[A\ b] = U \Sigma V^T$, $V \in \mathbb{R}^{n+1,n+1}$

$$\Rightarrow \quad [\hat{A}\ \hat{b}] = U_{:,1:n} (\Sigma)_{1:n,1:n} ((V)_{:,1:n})^T \in \mathbb{R}^{m,n+1}$$

We have $[\hat{A}\ \hat{b}]\, V_{:,n+1} = 0$, because $((V)_{:,1:n})^T V_{:,n+1} = 0$ due to orthogonality

$$\Updownarrow$$

$$\hat{A}(V)_{1:n,n+1} + (V)_{n+1,n+1}\hat{b} = 0 \quad \to \text{solution of } \hat{A}x = \hat{b}$$

$$x = -(V)^{-1}_{n+1,n+1} (V)_{1:n,n+1}$$

3.6. Constrained Least Squares

▷ Ovdt. LSE, $\tilde{A}\underline{x} = \tilde{\underline{b}}$, $\tilde{A} \in \mathbb{R}^{p+m,n}$, but $p < n$ should be satisfied exactly!

$$\hat{A}\underline{x} = \hat{b} \iff \begin{bmatrix} A \\ C \end{bmatrix} x = \begin{bmatrix} b \\ d \end{bmatrix}$$

Linear least squares problem with linear constraint:

Given: $\mathbf{A} \in \mathbb{R}^{m,n}$, $m \geq n$, $\operatorname{rank}(\mathbf{A}) = n$, $\mathbf{b} \in \mathbb{R}^m$,
$\mathbf{C} \in \mathbb{R}^{p,n}$, $p < n$, $\operatorname{rank}(\mathbf{C}) = p$, $\mathbf{d} \in \mathbb{R}^p$

Find: $\mathbf{x} \in \mathbb{R}^n$ such that $\|\mathbf{Ax} - \mathbf{b}\|_2 \to \min$ and $\boxed{\mathbf{Cx} = \mathbf{d}}$. (3.6.1)

Linear constraint

3.6.1. Solution via Langrangian Multipliers

$$x = \underset{y \in \mathbb{R}^n,\ Cy = \underline{d}}{\operatorname{argmin}} \|A\underline{y} - \underline{b}\|_2$$

$$x = \underset{\underline{y} \in \mathbb{R}^n}{\operatorname{argmin}} \underbrace{\max_{\underline{m} \in \mathbb{R}^p} \tfrac{1}{2}\|A\underline{y} - \underline{b}\|_2 + \underline{m}^T(C\underline{y} - \underline{d})}$$

Lagrangian functional $=: L(\underline{y}, \underline{m}) =$ "∞", if $C\underline{y} \neq d$

Idea: coupling the constraint using the Lagrange multiplier $\mathbf{m} \in \mathbb{R}^p$

$$\mathbf{x} = \underset{\mathbf{x}\in\mathbb{R}^n}{\operatorname{argmin}} \max_{\mathbf{m}\in\mathbb{R}^p} L(\mathbf{x},\mathbf{m}) \ , \tag{3.6.3}$$

$$L(\mathbf{x},\mathbf{m}) := \frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2 + \mathbf{m}^\top(\mathbf{C}\mathbf{x}-\mathbf{d}) \ . \tag{3.6.4}$$

saddle point problem

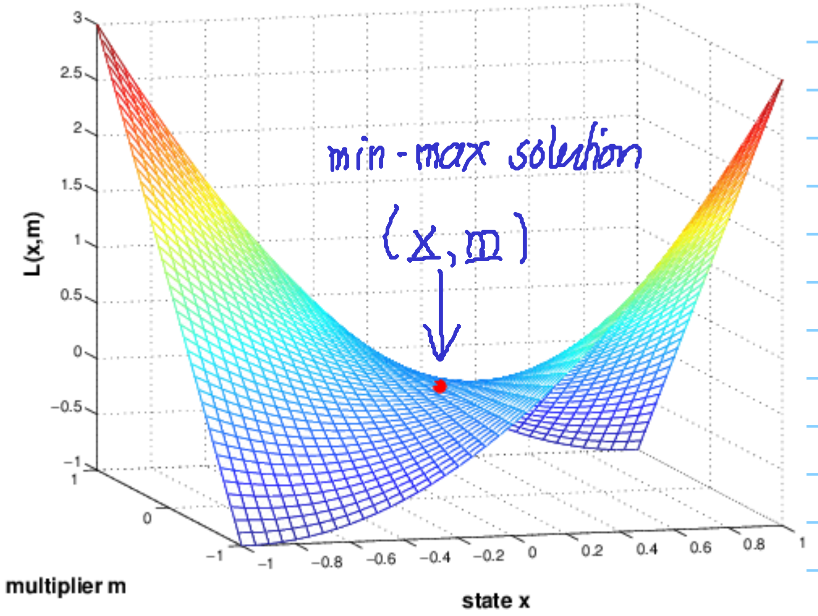


Fig. 126

Graph of L is "flat" in the solution point (x, m)

$\Downarrow$

All derivatives of L vanish in (x, m)

$$\frac{\partial L}{\partial \mathbf{x}}(\mathbf{x}, q) = \mathbf{A}^\top(\mathbf{A}\mathbf{x}-\mathbf{b}) + \mathbf{C}^\top q \stackrel{!}{=} 0 \ , \tag{3.6.6a}$$

$$\frac{\partial L}{\partial \mathbf{m}}(\mathbf{x}, q) = \mathbf{C}\mathbf{x}-\mathbf{d} \stackrel{!}{=} 0 \ . \tag{3.6.6b}$$

$$\begin{bmatrix} \mathbf{A}^\top\mathbf{A} & \mathbf{C}^\top \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ q \end{bmatrix} = \begin{bmatrix} \mathbf{A}^\top\mathbf{b} \\ \mathbf{d} \end{bmatrix} \tag{3.6.7}$$

Augmented normal equations (matrix saddle point problem)

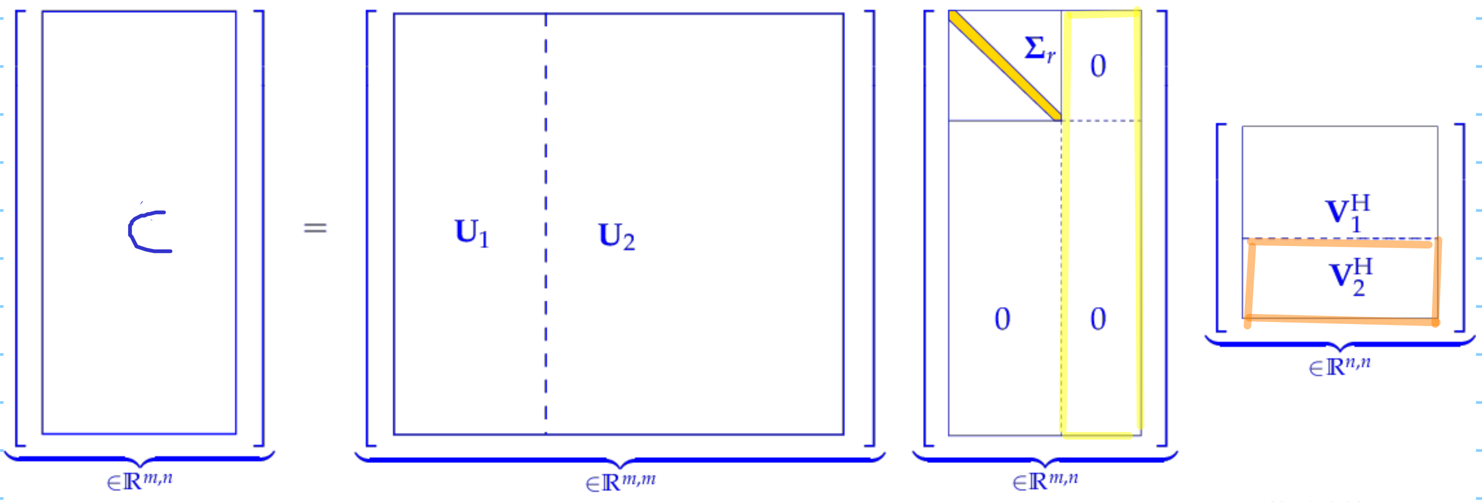
3.6.2. Solution via SVD

Constraint : $C\underline{x} = \underline{d}$ [underdetermined]

$\Rightarrow$ $x \in x_0 + \mathcal{N}(C)$, where $Cx_0 = \underline{d}$

available through SVD of C

$$C = [\,\mathbf{U}_1 \quad \mathbf{U}_2\,] \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{bmatrix}$$



(3.4.22)

$\mathcal{N}(C) = \mathcal{R}(V_2)$

$\Rightarrow$ $x = x_0 + V_2 y$, $y \in \mathbb{R}^{n-p}$

dim $\mathcal{N}(C)$

$\|Ax - b\|_2 \to \min \Rightarrow \|A(x_0 + V_2 y) - b\|_2 \to \min$

Std. unconstrained linear least squares problem!

$$A = U \Sigma V^T$$

$$AA^T = U \Sigma \cancel{V^T} V \Sigma^T U^T = U(\Sigma \Sigma^T) U^T$$

$$A^T A = \qquad = V(\Sigma^T \Sigma) V^T$$