2 Intermezzo on (Numerical) Linear Algebra

For $n \in \mathbb{N}$ we denote in the following by

$$\mathbf{e}_1^n = (1, 0, \ldots, 0), \quad \mathbf{e}_2^n = (0, 1, 0, \ldots, 0), \quad \ldots, \quad \mathbf{e}_n^n = (0, \ldots, 0, 1) \tag{2.0.1}$$

the unit vectors of the $\mathbb{R}^n$. Furthermore, in the following let $n, m \in \mathbb{N}$ be natural numbers and let $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$. Moreover, we denote

$$\langle \mathbf{v}, \mathbf{w} \rangle = \mathbf{v}^H \mathbf{w} = \overline{\mathbf{v}_1} \mathbf{w}_1 + \ldots + \overline{\mathbf{v}_n} \mathbf{w}_n \tag{2.0.2}$$

for all $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_n) \in \mathbb{C}^n$ and all $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_n) \in \mathbb{C}^n$.

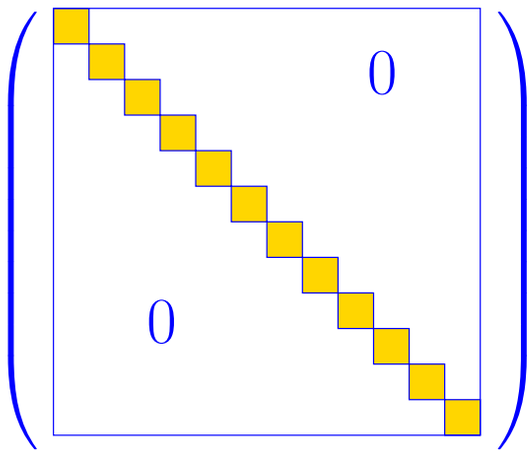
2.1 Types of matrices

Definition 2.1.1 (Types of matrices).

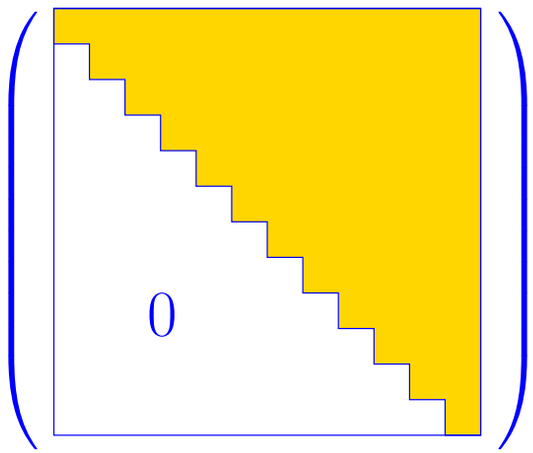
A matrix $\mathbf{A} = (a_{i,j})_{i\in\{1,\ldots,m\},j\in\{1,\ldots,n\}} \in \mathbb{K}^{m,n}$ *is called*

- *upper triangular matrix if* $\forall\, i \in \{1,\ldots,m\}, j \in \{1,\ldots,n\}, i > j : a_{i,j} = 0$,
- *lower triangular matrix if* $\forall\, i \in \{1,\ldots,m\}, j \in \{1,\ldots,n\}, i < j : a_{i,j} = 0$,
- *diagonal matrix if* $\forall\, i \in \{1,\ldots,m\}, j \in \{1,\ldots,n\}, i \neq j : a_{i,j} = 0$.

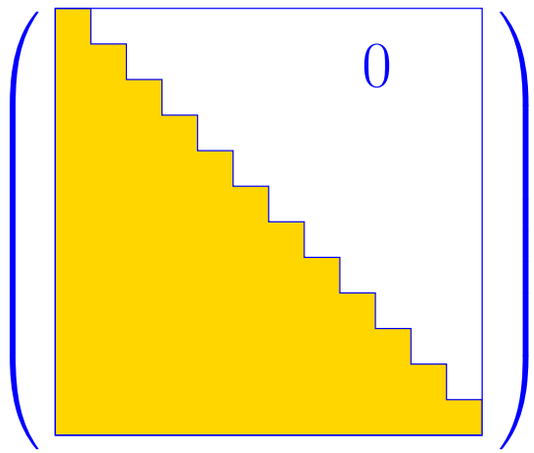
An upper/lower triangular matrix is called normalized if $\forall\, i \in \{1,\ldots,\min(m,n)\} : a_{i,i} = 1$.



diagonale matrix



upper triangular matrix



lower triangular matrix

Definition 2.1.2. *A matrix $\mathbf{A} \in \mathbb{C}^{n,n}$ is called unitary if*

$$\mathbf{A}^H \mathbf{A} = \mathbf{A}\mathbf{A}^H = \mathbf{I}.$$

In addition, a unitary matrix $\mathbf{A} \in \mathbb{C}^{n,n}$ is called orthogonal if $\mathbf{A} \in \mathbb{R}^{n,n}$.

For all unitary matrices $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{n,n}$ and all $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$ it holds that:

- $\mathbf{A}^H = \mathbf{A}^{-1}$,
- $\mathbf{A}^H$ and $\mathbf{A} \cdot \mathbf{B}$ are unitary,
- $|\det(\mathbf{A})| = 1$,
- $\|\mathbf{A}\mathbf{x}\|_2 = \|\mathbf{x}\|_2$ and $\langle \mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle$ (isometry),
- $\|\mathbf{A}\|_2 = 1$
- the columns of $\mathbf{A}$ form an orthonormal basis of the $\mathbb{C}^n$,
- the rows of $\mathbf{A}$ form an orthonormal basis of the $\mathbb{C}^n$,
- ...

Definition 2.1.3 (Hausholder reflections (reflection at a hyperplane)). *If $\mathbf{v} \in \mathbb{C}^m \backslash \{0\}$, then the matrix $\mathbf{H}(\mathbf{v}) \in \mathbb{C}^{m,m}$ defined by*

$$\mathbf{H}(\mathbf{v}) := \mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^H}{\|\mathbf{v}\|_2^2} = \mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^H}{\mathbf{v}^H\mathbf{v}} \tag{2.1.1}$$

is called Hausholder reflection associated to $\mathbf{v}$.

Remarks:

- if $\mathbf{v} \in \mathbb{R}^m \subset \mathbb{C}^m$, then $\mathbf{H}(\mathbf{v}) \in \mathbb{R}^{m,m} \subset \mathbb{C}^{m,m}$,
- it holds that $\mathbf{H}(\mathbf{v})^H = \mathbf{H}(\mathbf{v})$ (hermitian/symmetric matrix),
- it holds that $\mathbf{H}(\mathbf{v})$ is unitary,
- for every $\mathbf{x}, \mathbf{v} \in \mathbb{K}^m$ it holds that

$$\mathbf{H}(\mathbf{v})\mathbf{x} = \left(\mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^H}{\|\mathbf{v}\|_2^2}\right)\mathbf{x} = \mathbf{x} - 2\frac{\mathbf{v}\,\langle \mathbf{v}, \mathbf{x}\rangle}{\|\mathbf{v}\|_2^2} = \mathbf{x} - 2\frac{\mathbf{v}}{\|\mathbf{v}\|_2}\left\langle \frac{\mathbf{v}}{\|\mathbf{v}\|_2}, x\right\rangle, \tag{2.1.2}$$

  i.e., $\mathbf{H}(\mathbf{v})\mathbf{x}$ is the reflection of $\mathbf{x}$ at the hyperplane $\{\mathbf{y} \in \mathbb{K}^m : \langle \mathbf{v}, \mathbf{y}\rangle = 0\}$.

2.2 Condition of a matrix

Definition 2.2.1 (Condition of a matrix). *Let $\|\cdot\| : \mathbb{K}^n \to [0,\infty)$ be a norm and let $\mathbf{A} \in \mathbb{K}^{n,n}$. Then we call $\mathrm{cond}_{\|\cdot\|}(\mathbf{A}) \in [1,\infty]$ defined by*

$$\mathrm{cond}_{\|\cdot\|}(\mathbf{A}) := \begin{cases} \sup_{b,\delta\in\mathbb{R}^n\setminus\{0\}} \left(\frac{\left(\frac{\|\mathbf{A}^{-1}\delta\|}{\|\mathbf{A}^{-1}b\|}\right)}{\left(\frac{\|\delta\|}{\|b\|}\right)} \right) = \left\|\mathbf{A}^{-1}\right\|_{L(\|\cdot\|)} \|\mathbf{A}\|_{L(\|\cdot\|)} & : \mathbf{A} \text{ invertible} \\ \infty & : \text{else} \end{cases}$$

the condition(number) of $\mathbf{A} \in \mathbb{R}^{n,n}$ w.r.t. (the norm) $\|\cdot\|$. Instead of $\mathrm{cond}_{\|\cdot\|_p}(A)$ wir also write $\mathrm{cond}_p(A)$ for every $p \in \{1,2,\infty\}$.

- If $\mathrm{cond}_{\|\cdot\|}(\mathbf{A}) \gg 1$, then *relatively small disturbances* in the data $b \in \mathbb{R}^n$ can lead to *large relative errors* in the solution of the Linear System of Equations (LSE).
- If $\mathrm{cond}_{\|\cdot\|}(\mathbf{A}) \gg 1$, then an algorithm may produce solutions with *large relative errors*!

- For every invertible $\mathbf{A} \in \mathbb{K}^{n,n}$ it holds that:

$$1 = \|\mathbf{I}\|_{L(\|\cdot\|)} = \left\|\mathbf{A}^{-1} \cdot \mathbf{A}\right\|_{L(\|\cdot\|)} \leq \left\|\mathbf{A}^{-1}\right\|_{L(\|\cdot\|)} \|\mathbf{A}\|_{L(\|\cdot\|)} \tag{2.2.1}$$

  and hence $\mathrm{cond}_{\|\cdot\|}(\mathbf{A}) \geq 1$.

- For every unitary $\mathbf{A} \in \mathbb{C}^{n,n}$ and every $\mathbf{B} \in \mathbb{C}^{n,n}$ it holds that:

$$\mathrm{cond}_2(\mathbf{A}) = 1 \qquad \text{and} \qquad \mathrm{cond}_2(\mathbf{AB}) = \mathrm{cond}_2(\mathbf{B}). \tag{2.2.2}$$

2.3 Matrix decompositions

2.3.1 LU decomposition

Definition 2.3.1. *Let $\mathbf{A} \in \mathbb{C}^{n,n}$. A pair $(\mathbf{L}, \mathbf{U})$ consisting of a normalized lower triangular matrix $\mathbf{L} \in \mathbb{C}^{n,n}$ and an upper triangular matrix $\mathbf{U} \in \mathbb{C}^{n,n}$ is called LU decomposition of $\mathbf{A}$ if $\mathbf{LU} = \mathbf{A}$.*

Let $\mathbf{A} \in \mathbb{K}^{n,n}$ and let $\mathbf{b} \in \mathbb{K}^n$. Gaussian elimination without pivoting for solving the LSE

$$\mathbf{Ax} = \mathbf{b} \tag{2.3.1}$$

for $\mathbf{x} \in \mathbb{K}^n$ corresponds to the computation of an LU decomposition of $\mathbf{A}$.

Example 2.3.1 (Gaussian elimination and LU decomposition).

$$\begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ 3 & -1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ -3 \end{pmatrix} \quad \longleftrightarrow \quad \begin{aligned} x_1 + x_2 &= 4 \\ 2x_1 + x_2 - x_3 &= 1 \\ 3x_1 - x_2 - x_3 &= -3 \end{aligned} .$$

$$\begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ 3 & -1 & -1 \end{pmatrix} \begin{pmatrix} 4 \\ 1 \\ -3 \end{pmatrix} \Longrightarrow \begin{pmatrix} 1 & & \\ 2 & 1 & \\ & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{1} & 1 & 0 \\ 0 & -1 & -1 \\ 3 & -1 & -1 \end{pmatrix} \begin{pmatrix} 4 \\ -7 \\ -3 \end{pmatrix} \Longrightarrow$$

$$\begin{pmatrix} 1 & & \\ 2 & 1 & \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{1} & 1 & 0 \\ 0 & -1 & -1 \\ 0 & -4 & -1 \end{pmatrix} \begin{pmatrix} 4 \\ -7 \\ -15 \end{pmatrix} \Longrightarrow \underbrace{\begin{pmatrix} 1 & & \\ 2 & 1 & \\ 3 & 4 & 1 \end{pmatrix}}_{=L} \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & \mathbf{-1} & -1 \\ 0 & 0 & 3 \end{pmatrix}}_{=U} \begin{pmatrix} 4 \\ -7 \\ 13 \end{pmatrix}$$

■ = pivot row, pivot element **bold**, *negative* multipliers red

◇

Perspective: link Gaussian elimination to matrix factorization

(row transformation = multiplication with elimination matrix)

$$a_1 \neq 0 \quad \blacktriangleright \quad \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -\frac{a_2}{a_1} & 1 & & & 0 \\ -\frac{a_3}{a_1} & & \ddots & & \\ \vdots & & & & \\ -\frac{a_n}{a_1} & 0 & & & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} a_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} .$$

$\blacktriangleright$ $n-1$ steps of Gaussian elimination: $\Longrightarrow$ matrix factorization

(non-zero pivot elements assumed)

$$\mathbf{A} = \mathbf{L}_1 \cdot \cdots \cdot \mathbf{L}_{n-1} \mathbf{U} \qquad \text{with} \qquad \begin{array}{l} \text{elimination matrices } \mathbf{L}_i,\ i = 1, \ldots, n-1 \ , \\ \text{upper triangular matrix } \mathbf{U} \in \mathbb{R}^{n,n} \ . \end{array}$$

$$\begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ l_2 & 1 & & & 0 \\ l_3 & & \ddots & & \\ \vdots & & & & \\ l_n & 0 & & & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & & & 0 \\ 0 & h_3 & 1 & & \\ \vdots & \vdots & & \ddots & \\ 0 & h_n & 0 & & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ l_2 & 1 & & & 0 \\ l_3 & h_3 & 1 & & \\ \vdots & \vdots & & \ddots & \\ l_n & h_n & 0 & & 1 \end{pmatrix}$$

$\mathbf{L}_1 \cdot \cdots \cdot \mathbf{L}_{n-1}$ are normalized lower triangular matrices

(entries = multipliers $-\frac{a_{ik}}{a_{kk}}$)

Solving a linear system of equations by LU decomposition:

Algorithm 2.3.2 (Using LU decomposition to solve a linear system of equations).

$\mathbf{Ax} = \mathbf{b}$:

① LU-decomposition $\mathbf{A} = \mathbf{LU}$, #elementary operations $\frac{2}{3}n^3 + O(n^2)$

② forward substitution, solve $\mathbf{Lz} = \mathbf{b}$, #elementary operations $O(n^2)$

③ backward substitution, solve $\mathbf{Ux} = \mathbf{z}$, #elementary operations $O(n^2)$

Remark 2.3.3. Gaussian elimination without pivoting / LU decomposition is not always possible. E.g., there exists no LU decomposition for the matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{2.3.2}$$

However, Gaussian elimination with pivoting is possible for any invertible matrix (see Lemma 2.3.2 below).

△

Example 2.3.4 (Condition of row transformations).

Let $T(\mu) \in \mathbb{R}^{2,2}$, $\mu \in \mathbb{R}$, be row transformation matrices (*cf.* Gaussian elimination) defined by

$$\mathbf{T}(\mu) := \begin{pmatrix} 1 & 0 \\ \mu & 1 \end{pmatrix}$$

for all $\mu \in \mathbb{R}$. Conditionnumbers of $\mathbf{T}(\mu)$, $\mu \in [0, \infty)$ ▷

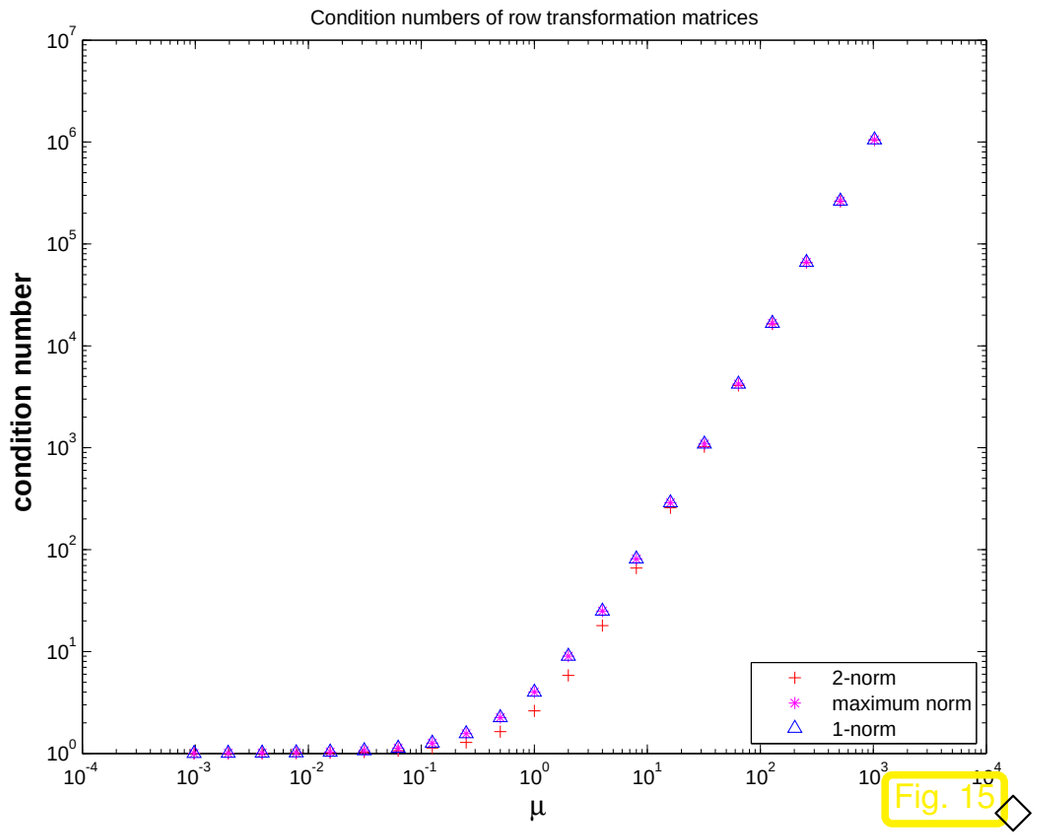


Fig. 15 ◇

Stability needs pivoting!

Example 2.3.5.

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 2 \\ 2 & -7 & 2 \\ 1 & 24 & 0 \end{pmatrix} \xrightarrow{❶} \begin{pmatrix} 2 & -7 & 2 \\ 1 & 2 & 2 \\ 1 & 24 & 0 \end{pmatrix} \xrightarrow{❷} \begin{pmatrix} 2 & -7 & 2 \\ 0 & 5.5 & 1 \\ 0 & 27.5 & -1 \end{pmatrix} \xrightarrow{❸} \begin{pmatrix} 2 & -7 & 2 \\ 0 & 27.5 & -1 \\ 0 & 5.5 & 1 \end{pmatrix} \xrightarrow{❹} \begin{pmatrix} 2 & -7 & 2 \\ 0 & 27.5 & -1 \\ 0 & 0 & 1.2 \end{pmatrix}$$

$$\mathbf{U} = \begin{pmatrix} 2 & -7 & 2 \\ 0 & 27.5 & -1 \\ 0 & 0 & 1.2 \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0.5 & 0.2 & 1 \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} .$$

◇

Lemma 2.3.2 (Existence of a LU decomposition with row pivoting (a.k.a. partial pivoting))**.** *For every invertible $\mathbf{A} \in \mathbb{K}^{n,n}$ there are a permutation matrix $\mathbf{P} \in \mathbb{K}^{n,n}$, a normalized lower triangular matrix $\mathbf{L} \in \mathbb{K}^{n,n}$ and an invertible upper triangular matrix $\mathbf{U} \in \mathbb{K}^{n,n}$ such that $\mathbf{PA} = \mathbf{LU}$.*

MATLAB-functions: `[L,U,P] = lu(A)`

`L,U` = matrices L and U

`P` = permutation matrix P

Note: Typically $\mathrm{cond}_2(\mathbf{L}) > 1$ in Lemma 2.3.2.

Round-off errors can be dangerous.

Remark 2.3.6. Not discussed in this lecture, but of essential importance in applications are the **sparse matrices** (i.e. having the number of non-zero elements much smaller than n^2). Special storing schemes and algorithms can sometimes keep the factors L and U sparse, but in general this is difficult or impossible. For such cases, **iterative methods** for LSE (as e.g. preconditioned conjugate gradient) are the methods of choice.

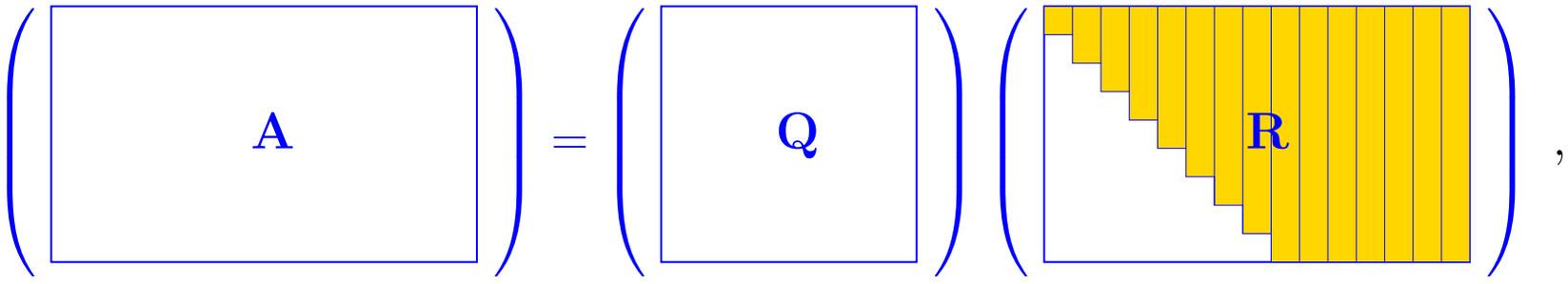
△

2.3.2 QR decomposition

Definition 2.3.3. *Let $\mathbf{A} \in \mathbb{C}^{m,n}$. A pair $(\mathbf{Q}, \mathbf{R})$ consisting of a unitary matrix $\mathbf{Q} \in \mathbb{C}^{m,m}$ and an upper triangular matrix $\mathbf{R} \in \mathbb{C}^{m,n}$ is called QR decomposition of $\mathbf{A}$ if*

$$\mathbf{QR} = \mathbf{A}. \tag{2.3.3}$$

Example 2.3.8. A QR decomposition of $\mathbf{A} \in \mathbb{C}^{m,n}$ in the case $m < n$:



,

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

where $\mathbf{Q}$ unitary and $\mathbf{R}$ upper triangular matrix.

◇

How to construct a QR decomposition of a matrix $\mathbf{A} \in \mathbb{C}^{m,n}$?

Goal: Find a unitary transformation matrix which renders certain matrix elements zero

$$\mathbf{Q} \left(\begin{array}{|c|} \hline \\ \\ \\ \hline \end{array} \right) = \left(\begin{array}{|c|} \hline \\ 0 \\ \\ \hline \end{array} \right) \quad \text{with} \quad \mathbf{Q}^H = \mathbf{Q}^{-1} \, .$$

F or a given $\mathbf{a} = (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in \mathbb{K}^m$ find a unitary matrix $\mathbf{Q} \in \mathbb{K}^{m,m}$ such that

$$\mathbf{Qa} = \|\mathbf{a}\|_2 \, \mathbf{e}_1^m \qquad \text{(or, more generally: } \mathbf{Qa} = \lambda \, \|\mathbf{a}\|_2 \, \mathbf{e}_1^m \text{ with } |\lambda| = 1\text{).} \tag{2.3.4}$$

- Choice 1: Householder reflections: Reflection at the hyperplane

$$\{\mathbf{x} \in \mathbb{K}^m : \ \langle \mathbf{a} - \|\mathbf{a}\|_2 \, \mathbf{e}_1^m, \mathbf{x} \rangle = 0\} \tag{2.3.5}$$

in the case $\mathbf{a} \neq \|\mathbf{a}\|_2 \mathbf{e}_1$, i.e.,

$$\mathbf{Q} = \begin{cases} \mathbf{H}(\mathbf{a} - \|\mathbf{a}\|_2 \mathbf{e}_1) & : \mathbf{a} \neq \|\mathbf{a}\|_2 \mathbf{e}_1 \\ \mathbf{I} & : \text{else} \end{cases} \tag{2.3.6}$$

- Choice 2: successive Givens rotations:

$$\underbrace{\mathbf{G}_{1,k}(a_1, a_k)}_{\in \mathbb{C}^{n,n}} \mathbf{a} := \begin{pmatrix} \bar{\gamma} & \cdots & \bar{\sigma} & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ -\sigma & \cdots & \gamma & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_k \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} a_1^{(1)} \\ \vdots \\ 0 \\ \vdots \\ a_n \end{pmatrix} \quad \text{with} \quad \begin{aligned} \gamma &= \frac{a_1}{\sqrt{|a_1|^2+|a_k|^2}}, \\ \sigma &= \frac{a_k}{\sqrt{|a_1|^2+|a_k|^2}}. \end{aligned} \tag{2.3.7}$$

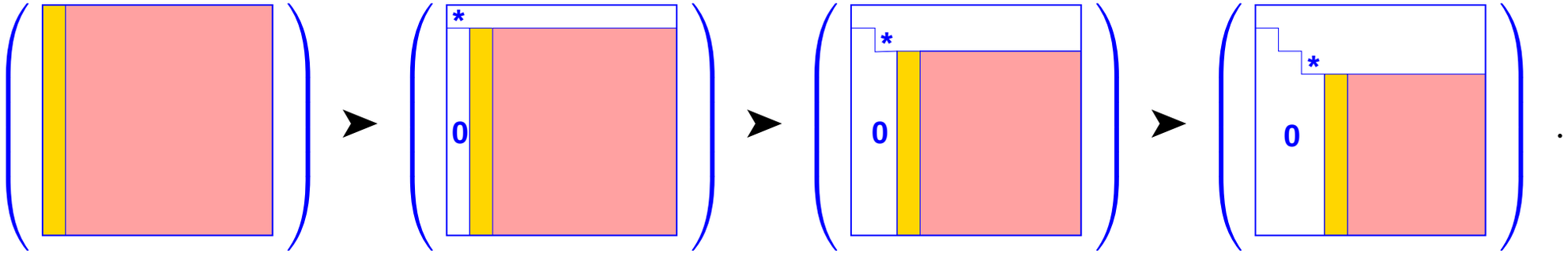
MATLAB-Function: `[G,x] = planerot(a);`

Code 2.1: (plane) Givens rotation

```
function [G,x] = planerot(a)
%plane Givens rotation.
if (a(2) ~= 0), r = norm(a); G = [a'; -a(2) a(1)]/r; x = [r; 0];
else, G = eye(2); end
```

- Other choices possible . . .

Transformation to *upper triangular form* ($\rightarrow$ Def. 2.1.1) by successive unitary transformations:



▭ = "target column $\mathbf{a}$" (determines unitary transformation),

▭ = modified in course of transformations.

$$\mathbf{Q}_{n-1}\mathbf{Q}_{n-2}\cdot\cdots\cdot\mathbf{Q}_1\mathbf{A}=\mathbf{R}\ ,$$

QR factorization
(QR decomposition)

of $\mathbf{A}\in\mathbb{C}^{m,n}$: $\mathbf{A}=\mathbf{QR}$, $\mathbf{Q}:=\mathbf{Q}_1^H\cdot\cdots\cdot\mathbf{Q}_{n-1}^H$ unitary matrix ,
$\mathbf{R}$ upper triangular matrix .

We have thus proved the following theorem.

Theorem 2.3.6 (Existence of QR decomposition)**.**

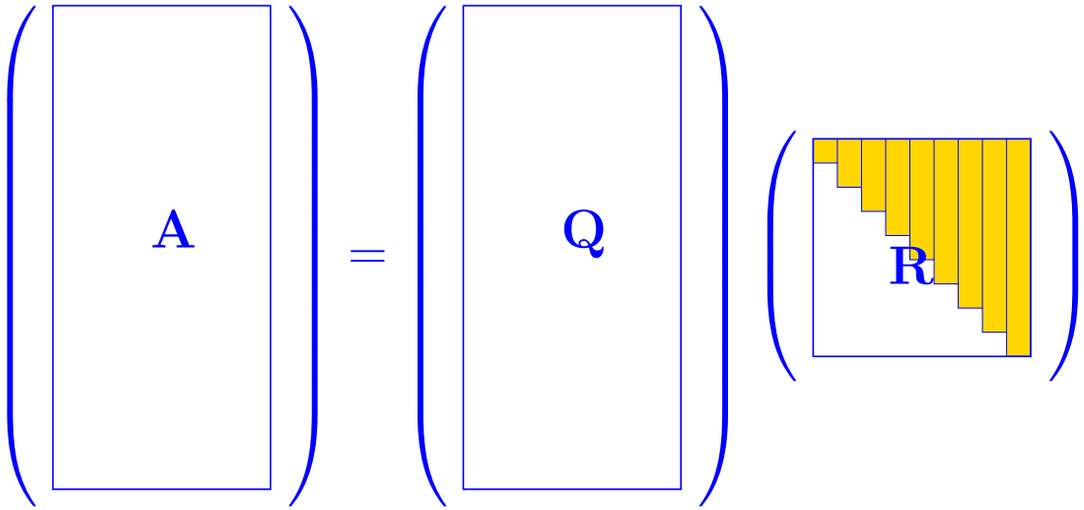
Let $\mathbf{A} \in \mathbb{K}^{m,n}$. Then there are a unitary matrix $\mathbf{Q} \in \mathbb{K}^{m,m}$ and an upper triangular matrix $\mathbf{R} \in \mathbb{K}^{m,n}$ such that $(\mathbf{Q}, \mathbf{R})$ is a QR decomposition of A, i.e., such that

$$\mathbf{QR} = \mathbf{A}. \tag{2.3.8}$$

Definition 2.3.7. *Let $\mathbf{A} \in \mathbb{C}^{m,n}$. A pair $(\mathbf{Q}, \mathbf{R})$ consisting of a matrix $\mathbf{Q} \in \mathbb{C}^{m,\min(n,m)}$ with $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$ and an upper triangular matrix $\mathbf{R} \in \mathbb{C}^{\min(n,m),n}$ is called economical QR decomposition of $\mathbf{A}$ if*

$$\mathbf{QR} = \mathbf{A}. \tag{2.3.9}$$

Example 2.3.10. An economical QR decomposition of $\mathbf{A} \in \mathbb{K}^{m,n}$ in the case $m > n$:



$$, \mathbf{A} = \mathbf{QR}, \quad \begin{matrix} \mathbf{Q} \in \mathbb{K}^{m,n} \\ \mathbf{R} \in \mathbb{K}^{n,n} \end{matrix},$$

where $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$ (orthonormal columns) and $\mathbf{R}$ upper triangular matrix.

◇

Lemma 2.3.8 (Uniqueness of economical QR decompositions)**.**

Let $\mathbf{A} \in \mathbb{K}^{m,n}$ with $\operatorname{rank}(\mathbf{A}) = n$. Then there exists exactly one pairing $(\mathbf{Q}, \mathbf{R})$ of a matrix $\mathbf{Q} \in \mathbb{K}^{m,n}$ with $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$ and of an upper triangular matrix $\mathbf{R} \in \mathbb{K}^{n,n}$ satisfying

$$\mathbf{QR} = \mathbf{A} \tag{2.3.10}$$

and

$$\langle \mathbf{e}_i^n, R\mathbf{e}_i^n \rangle > 0 \tag{2.3.11}$$

for all $i \in \{1, \ldots, n\}$.

MATLAB functions: for every $\mathbf{A} \in \mathbb{K}^{m,n}$:

`[Q,R] = qr(A)` $\mathbf{Q} \in \mathbb{K}^{m,m}$, $\mathbf{R} \in \mathbb{K}^{m,n}$

`[Q,R] = qr(A,0)` $\mathbf{Q} \in \mathbb{K}^{m,\min(n,m)}$, $\mathbf{R} \in \mathbb{K}^{\min(n,m),n}$ (*economical* QR-factorization)

Computational effort for Householder QR-factorization of $\mathbf{A} \in \mathbb{K}^{m,n}$, $m > n$:

`[Q,R] = qr(A)` → Costs: $O(m^2 n)$

`[Q,R] = qr(A,0)` → Costs: $O(mn^2)$

Example 2.3.12 (Complexity of Householder QR-factorization).

Code 2.2: timing MATLAB QR-factorizations

```
1 % Timing QR factorizations
2
3 K = 3; r = [];
4 for n=2.^(2:6)
5   m = n*n;
6
7   A = (1:m)'*(1:n) + [eye(n);ones(m-n,n)];
```

```
8   t1 = 1000; for k=1:K, tic; [Q,R] = qr(A); t1 = min(t1,toc); clear
      Q,R; end
9   t2 = 1000; for k=1:K, tic; [Q,R] = qr(A,0); t2 = min(t2,toc); clear
      Q,R; end
10  t3 = 1000; for k=1:K, tic; R = qr(A); t3 = min(t3,toc); clear R; end
11  r = [r; n, m, t1, t2, t3];
12 end
```

`tic-toc`-timing of different variants of QR-factorization in MATLAB ▷

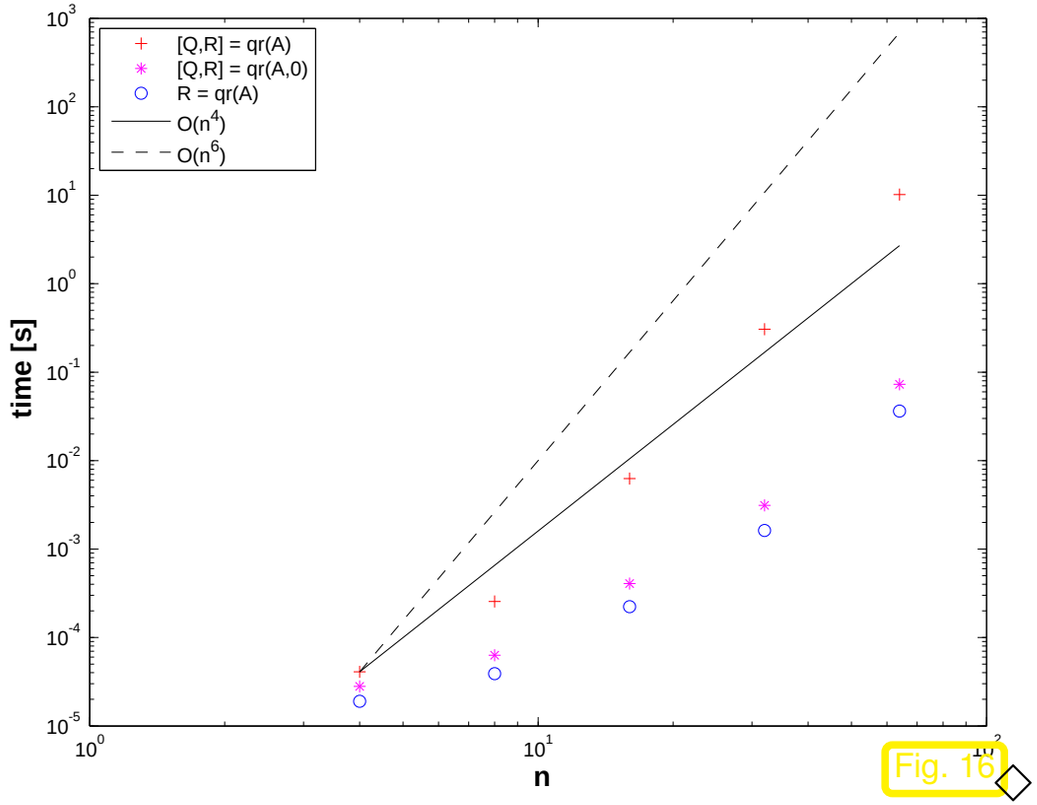
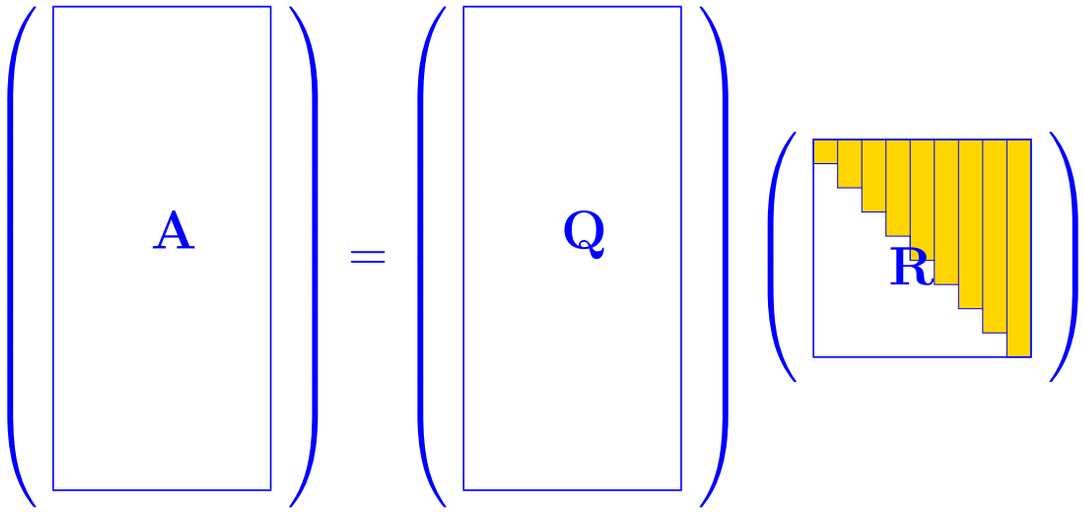


Fig. 16 ◇

Use `[Q,R] = qr(A,0)`, if output sufficient !

Remark 2.3.14 (QR-orthogonalization). Let $m \geq n$, let $\mathbf{A} \in \mathbb{K}^{m,n}$ with $\mathrm{rank}(\mathbf{A}) = n$ and let $(\mathbf{Q}, \mathbf{R})$

be an economical QR decomposition of $\mathbf{A}$.



Then $\mathrm{rank}(\mathbf{R}) = n$ (i.e., $\mathbf{R}$ is invertible) and the columns $\mathbf{Q}_{:,1}, \ldots, \mathbf{Q}_{:,n}$ of $\mathbf{Q}$ are an orthonormal basis of $\mathrm{Im}(\mathbf{A})$ with

$$\mathrm{Span}\left\{\mathbf{q}_{:,1}, \ldots, \mathbf{q}_{:,k}\right\} = \mathrm{Span}\left\{\mathbf{A}_{:,1}, \ldots, \mathbf{A}_{:,k}\right\}$$

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

△

Algorithm 2.3.16 (Solving linear system of equations by means of QR-decomposition).

$\mathbf{Ax} = \mathbf{b}$:

① QR-decomposition $\mathbf{A} = \mathbf{QR}$, #elementary operations $\frac{4}{3}n^3 + O(n^2)$ (about twice as expensive as LU-decomposition without pivoting)

② orthogonal transformation $\mathbf{z} = \mathbf{Q}^H\mathbf{b}$, #elementary operations $O(n^2)$ (in the case of *compact storage* of reflections/rotations)

③ Backward substitution, solve $\mathbf{Rx} = \mathbf{z}$, #elementary operations $O(n^2)$

✌ For *any* invertible matrix the associated LSE can be solved by means of

QR-decomposition + orthogonal transformation + backward substitution

in a stable manner.

Example 2.3.17 (Stable solution of LSE by means of QR-decomposition).

Code 2.3: QR-fac. ↔ Gaussian elimination

```
res = [];
for n=10:10:1000
  A=[tril(-ones(n,n-1))+2*[eye(n-1);...
       zeros(1,n-1)],ones(n,1)];
  x=((-1).^(1:n))';
  b=A*x;
  [Q,R]=qr(A);

  errlu=norm(A\b-x)/norm(x);
  errqr=norm(R\(Q'*b)-x)/norm(x);
  res=[res; n,errlu,errqr];
end
semilogy(res(:,1),res(:,2),'m-*',...
         res(:,1),res(:,3),'b-+');
```

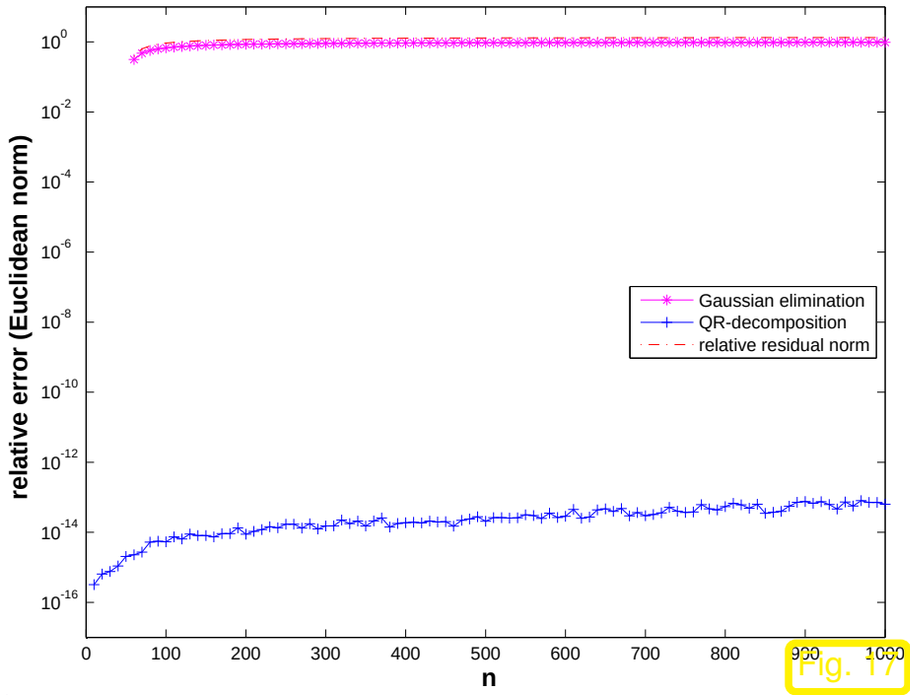


Fig. 17

◁ superior stability of QR-decomposition !

◇

2.3.3 Singular value decomposition (SVD)

Definition 2.3.9 (Diagonal matrices)**.**

For every $k \in \{1,\ldots,\min(m,n)\}$ *and every* $\sigma_1,\ldots,\sigma_k \in \mathbb{C}$ *we denote by* $\mathrm{diag}_{m,n}(\sigma_1,\ldots,\sigma_k) = \big([\mathrm{diag}_{m,n}(\sigma_1,\ldots,\sigma_k)]_{i,j}\big)_{i\in\{1,\ldots,m\},j\in\{1,\ldots,n\}} \in \mathbb{C}^{m,n}$ *the* $m \times n$*-matrix satisfying*

$$[\mathrm{diag}_{m,n}(\sigma_1,\ldots,\sigma_k)]_{i,j} = \begin{cases} \sigma_i & : i = j \le k \\ 0 & : \textit{else} \end{cases} \tag{2.3.12}$$

for all $i \in \{1,\ldots,m\}$ *and all* $j \in \{1,\ldots,n\}$*.*

Definition 2.3.10 (Singular value decomposition (SVD) and singular values)**.**

Let $A \in \mathbb{C}^{m,n}$, let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)} \geq 0$ be real numbers and let $\mathbf{U} \in \mathbb{C}^{m,m}$ and $\mathbf{V} \in \mathbb{C}^{n,n}$ be unitary matrices with

$$\mathbf{A} = \mathbf{U}\, \mathrm{diag}_{m,n}(\sigma_1, \ldots, \sigma_{\min(m,n)})\, \mathbf{V}^H\,. \tag{2.3.13}$$

Then the triple $(\mathbf{U}, \mathrm{diag}_{m,n}(\sigma_1, \ldots, \sigma_{\min(m,n)}), \mathbf{V})$ is called singular value decomposition (SVD) of $\mathbf{A}$ and

- $s_1(\mathbf{A}) := \sigma_1$ *is called the* 1*st singular value of* $\mathbf{A}$*,*
- $s_2(\mathbf{A}) := \sigma_2$ *is called the* 2*nd singular value of* $\mathbf{A}$*,*
- $\ldots$*,*
- $s_{\min(m,n)}(\mathbf{A}) := \sigma_{\min(m,n)}$ *is called the* $\min(m,n)$*-th singular value of* $\mathbf{A}$*.*

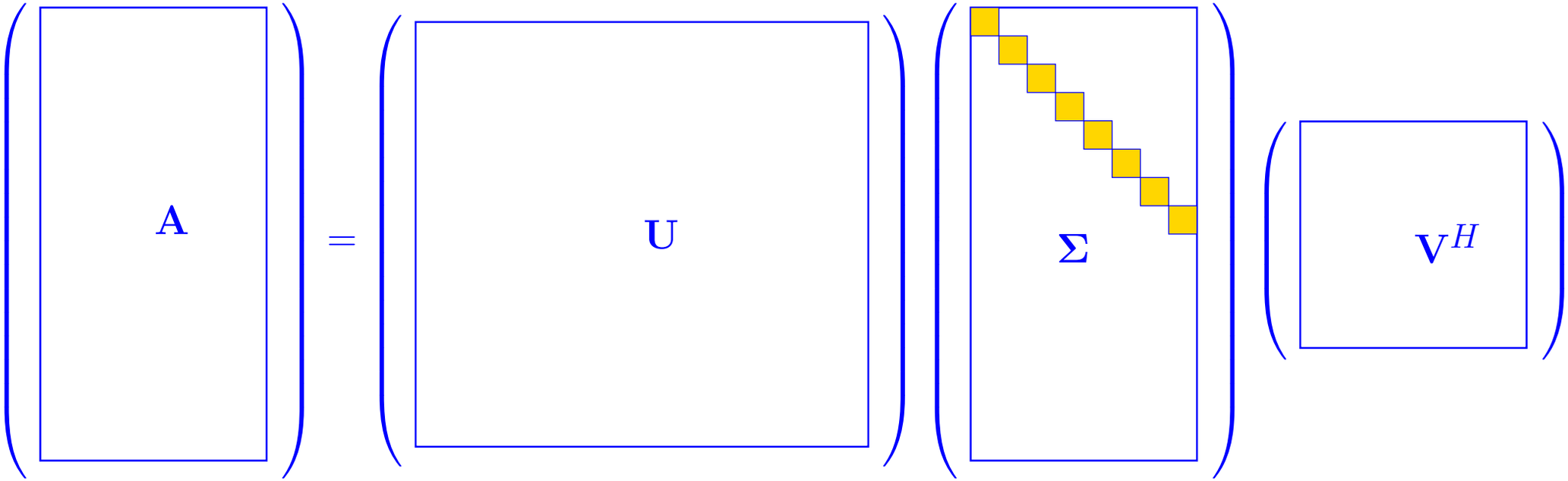
The next theorem shows that each $m \times n$-matrix $\mathbf{A} \in \mathbb{C}^{m,n}$ admits a singular value decomposition and that the singular values $s_1(\mathbf{A})$, $\ldots$, $s_{\min(m,n)}(\mathbf{A})$ are unique and thereby well defined.

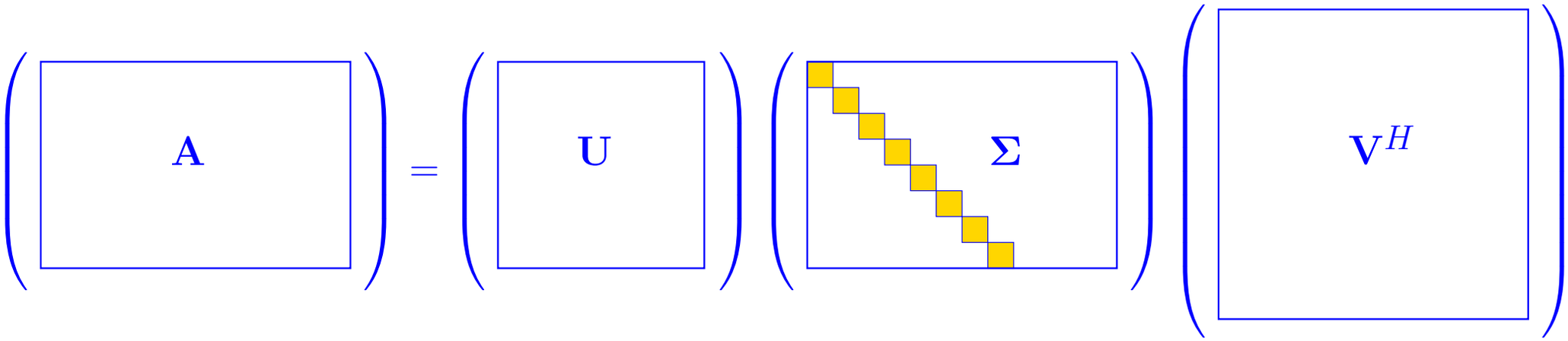
Theorem 2.3.11 (Existence of a SVD and uniqueness of singular values)**.** *Let* $\mathbf{A} \in \mathbb{K}^{m,n}$*. Then there exist real numbers* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)} \geq 0$ *and unitary matrices* $\mathbf{U} \in \mathbb{K}^{m,m}$ *and* $\mathbf{V} \in \mathbb{K}^{n,n}$ *such that*

$$\mathbf{A} = \mathbf{U}\,\mathrm{diag}_{m,n}(\sigma_1, \ldots, \sigma_{\min(m,n)})\,\mathbf{V}^H\,.$$

The triple $(\mathbf{U}, \mathrm{diag}_{m,n}(\sigma_1, \ldots, \sigma_{\min(m,n)}), \mathbf{V})$ *is thus a SVD of* $\mathbf{A}$*. Moreover, if* $(\tilde{\mathbf{U}}, \tilde{\mathbf{\Sigma}}, \tilde{\mathbf{V}})$ *is a SVD of* $\mathbf{A}$*, then*

$$\tilde{\mathbf{\Sigma}} = \mathrm{diag}_{m,n}(\sigma_1, \ldots, \sigma_{\min(m,n)}). \tag{2.3.14}$$





MATLAB-functions (for algorithms see [20, Sect. 8.3]):

`s = svd(A)` : computes singular values of $\mathbf{A} \in \mathbb{K}^{m,n}$

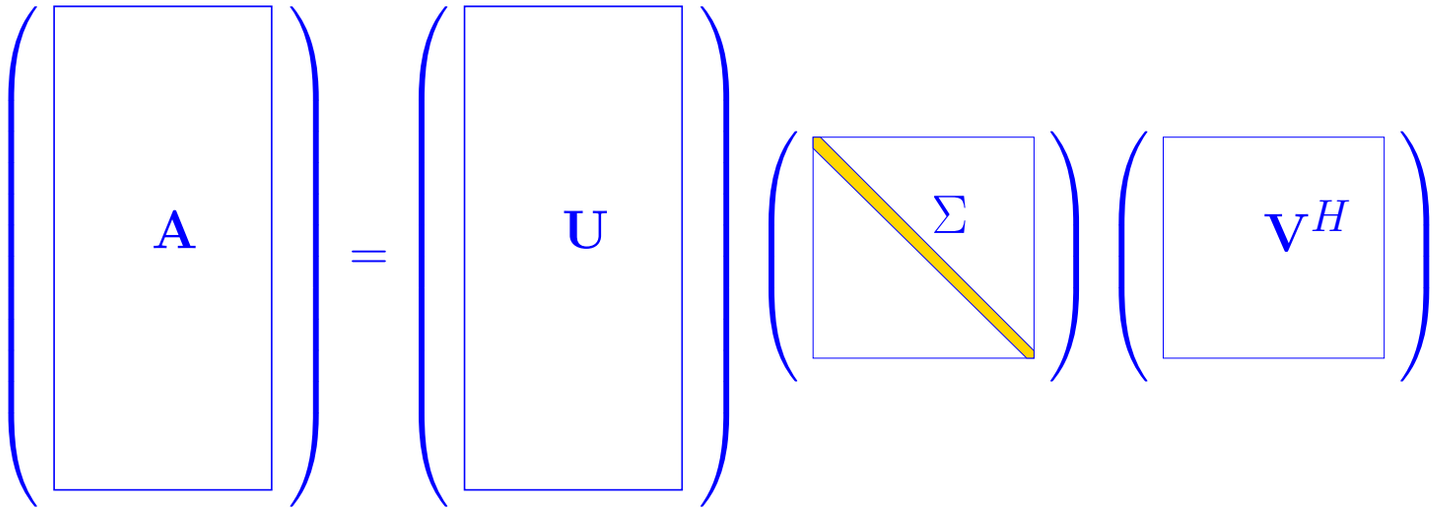
`[U,S,V] = svd(A)` : computes singular value decomposition of $\mathbf{A} \in \mathbb{K}^{m,n}$ according to Thm. 2.3.11

`[U,S,V] = svd(A,0)` : computes "economical" singular value decomposition of $\mathbf{A} \in \mathbb{K}^{m,n}$: $\mathbf{U} \in \mathbb{K}^{m,\min(m,n)}$, $\Sigma \in \mathbb{R}^{\min(m,n),n}$, $\mathbf{V} \in \mathbb{K}^{n,n}$

`s = svds(A,k)` : computes $s_1(\mathbf{A}), \ldots, s_{\min(k,n,m)}(\mathbf{A})$, i.e., the first $\min(k,n,m)$ singular values of $\mathbf{A} \in \mathbb{K}^{m,n}$ (important for sparse $\mathbf{A} \in \mathbb{K}^{m,n}$)

`[U,S,V] = svds(A,k)` : computes partial singular value decomposition of $\mathbf{A} \in \mathbb{K}^{m,n}$: $\mathbf{U} \in \mathbb{K}^{m,\min(k,m,n)}$, $\Sigma = \mathrm{diag}_{\min(k,m,n),\min(k,m,n)}(s_1(\mathbf{A}), \ldots, s_{\min(k,m,n)}(\mathbf{A})) \in \mathbb{R}^{\min(k,m,n),\min(k,m,n)}$, $\mathbf{V} \in \mathbb{K}^{n,\min(k,m,n)}$

Explanation: "economical" singular value decomposition:



(MATLAB) algorithm for computing SVD is (numerically) stable

Complexity:

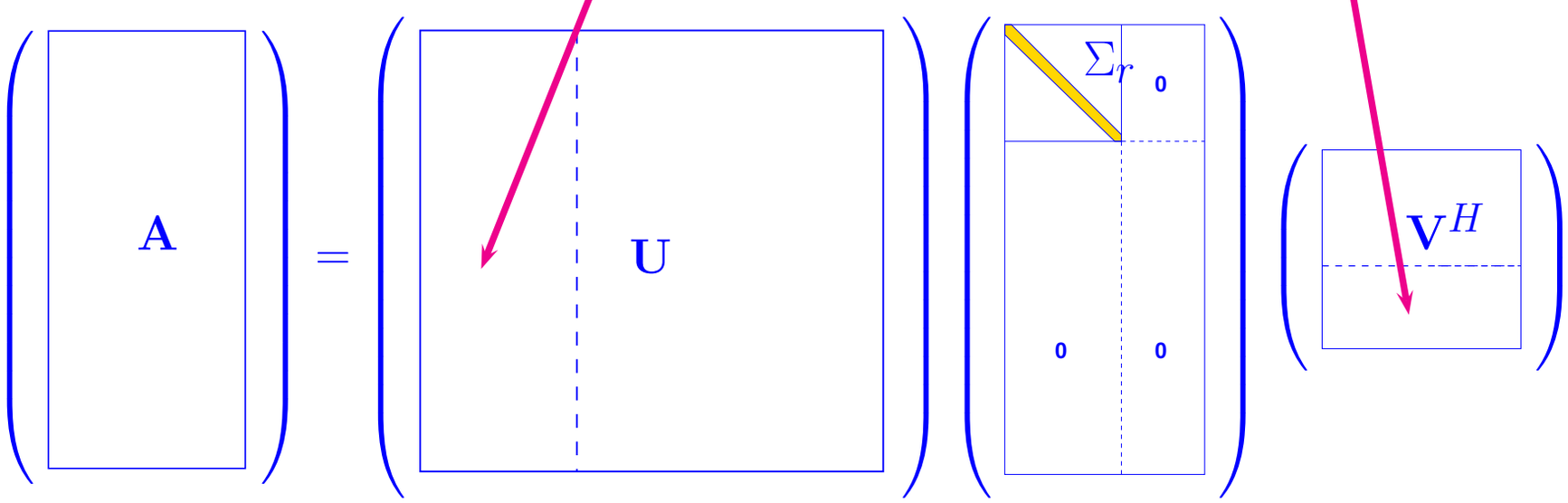
$2mn^2 + 2n^3 + O(n^2) + O(mn)$ for `s = svd(A)`,

$4m^2n + 22n^3 + O(mn) + O(n^2)$ for `[U,S,V] = svd(A)`,

$O(mn^2) + O(n^3)$ for `[U,S,V]=svd(A,0)`, $m \gg n$.

- Application of SVD: computation of rank, kernel and range of a matrix

Illustration: columns = ONB of $\mathrm{Im}(\mathbf{A})$ rows = ONB of $\mathrm{Ker}(\mathbf{A})$



(2.3.18)

Lemma 2.3.14 (Properties of singular values)**.** *For every* $\mathbf{A} \in \mathbb{C}^{m,n}$ *it holds that*

$$s_1(\mathbf{A}) = \|A\|_2 \quad \textit{and} \quad \mathrm{rank}(\mathbf{A}) = \#\big(\{s_1(\mathbf{A}), \ldots, s_{\min(m,n)}(\mathbf{A})\} \cap (0, \infty)\big). \tag{2.3.19}$$

Remark: MATLAB function r=rank(A) relies on svd(A)

2.3.3.1 Application of SVD: Best low rank approximation

Definition 2.3.15 (Frobenius norm)**.**

For every $\mathbf{A} = (a_{i,j})_{i\in\{1,\ldots,m\},j\in\{1,\ldots,n\}} \in \mathbb{K}^{m,n}$ *the Frobenius norm of* $\mathbf{A}$ *is defined as*

$$\|\mathbf{A}\|_F^2 := \sum_{i=1}^{m}\sum_{j=1}^{n} |a_{i,j}|^2 \,.$$

Theorem 2.3.17 (Best low rank approximation)**.**

Let $\mathbf{A} \in \mathbb{K}^{m,n}$*, let* $(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V})$ *be a SVD of* $\mathbf{A}$ *with* $\mathbf{U} \in \mathbb{K}^{m,m}$ *and* $\mathbf{V} \in \mathbb{K}^{n,n}$ *(→ Thm. 2.3.11) and let* $k \in \{1, \ldots, \operatorname{rank}(\mathbf{A})\}$*. Then*

$$\left\|\mathbf{A} - \tilde{\mathbf{A}}\right\|_2 \geq \left\|\mathbf{A} - \mathbf{U}\, \operatorname{diag}_{m,n}(s_1(\mathbf{A}), \ldots, s_k(\mathbf{A}))\, \mathbf{V}^H\right\|_2 = \begin{cases} s_{k+1}(\mathbf{A}) & : k \neq \operatorname{rank}(\mathbf{A}) \\ 0 & : k = \operatorname{rank}(\mathbf{A}) \end{cases}$$

and

$$\left\|\mathbf{A} - \tilde{\mathbf{A}}\right\|_F \geq \left\|\mathbf{A} - \mathbf{U}\, \operatorname{diag}_{m,n}(s_1(\mathbf{A}), \ldots, s_k(\mathbf{A}))\, \mathbf{V}^H\right\|_F = \sqrt{\sum_{j=k+1}^{\operatorname{rank}(\mathbf{A})} |s_j(\mathbf{A})|^2}$$

for all $\tilde{\mathbf{A}} \in \mathbb{K}^{m,n}$ *with* $\operatorname{rank}(\tilde{\mathbf{A}}) \leq k$*.*

Note: A matrix $\tilde{\mathbf{A}} \in \mathbb{K}^{m,n}$ can be uniquely described by $\operatorname{rank}(\mathbf{A}) \cdot (m + n + 1)$ numbers from $\mathbb{K}$!

Approximation by low-rank matrices $\leftrightarrow$ matrix compression

2.3.3.2 Application of SVD: Principal component analysis (PCA)

Given: n data points $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^m$ in m-dimensional (feature) space

Notation: Define $V := \mathrm{Span}\{\mathbf{a}_1, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^m$ and $p := \dim(V)$ (*unknown*).

Conjectured: "linear dependence" of $\mathbf{a}_1, \ldots, \mathbf{a}_n \in V$ and $p < \min(m, n)$
($\succ$ possibility of dimensional reduction)

Task (PCA): determine p and (orthonormal basis of) V
(detect linear dependence)

Perspective of linear algebra: Define $\mathbf{A} := (\mathbf{a}_1, \ldots, \mathbf{a}_n) \in \mathbb{R}^{m,n}$ and
observe that $V = \mathrm{Im}(\mathbf{A})$ and $p = \mathrm{rank}(\mathbf{A})$.

Extension: Data affected by measurement errors
(but conjecture upheld for unperturbed data)

Application: ◣Chemometrics (multivariate calibration methods for the analysis of chemical mixtures)

PCA by SVD

❶ no perturbations:

SVD: $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ satisfies $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_p > \sigma_{p+1} = \cdots = \sigma_{\min\{m,n\}} = 0$,

$$V = \operatorname{Span} \underbrace{\{(\mathbf{U})_{:,1}, \ldots, (\mathbf{U})_{:,p}\}}_{\text{ONB of } V} .$$

❷ with perturbations:

SVD: $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ satisfies $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_p \gg \sigma_{p+1} \approx \cdots \approx \sigma_{\min\{m,n\}} \approx 0$,

$$V = \operatorname{Span} \underbrace{\{(\mathbf{U})_{:,1}, \ldots, (\mathbf{U})_{:,p}\}}_{\text{ONB of } V} .$$

If there is a pronounced gap in distribution of the singular values, which separates p large from $\min\{m,n\} - p$ relatively small singular values, this hints that $\operatorname{Im}(\mathbf{A})$ has "essentially dimension" p.

It depends on the application what one accepts as a "pronounced gap".

Code 2.4: PCA in three dimensions via SVD

```
1 % Use of SVD for PCA with perturbations
2
3 V = [1 , -1; 0 , 0.5; -1 , 0]; A = V*rand(2,20)+0.1*rand(3,20);
4 [U,S,V] = svd(A,0);
5
6 figure; sv = diag(S(1:3,1:3))
7
8 [X,Y] = meshgrid(-2:0.2:0,-1:0.2:1); n = size(X,1); m = size(X,2);
```

```
figure; plot3(A(1,:),A(2,:),A(3,:),'r*'); grid on; hold on;
M = U(:,1:2)*[reshape(X,1,prod(size(X)));reshape(Y,1,prod(size(Y)))];
mesh(reshape(M(1,:),n,m),reshape(M(2,:),n,m),reshape(M(3,:),n,m));
colormap(cool); view(35,10);

print -depsc2 '../PICTURES/svdpca.eps';
```

singular values:

$$\begin{matrix} 3.1378 \\ 1.8092 \\ \hline 0.1792 \end{matrix}$$

We observe a gap between the second and third singular value ➢ the data points essentially lie in a 2D subspace.

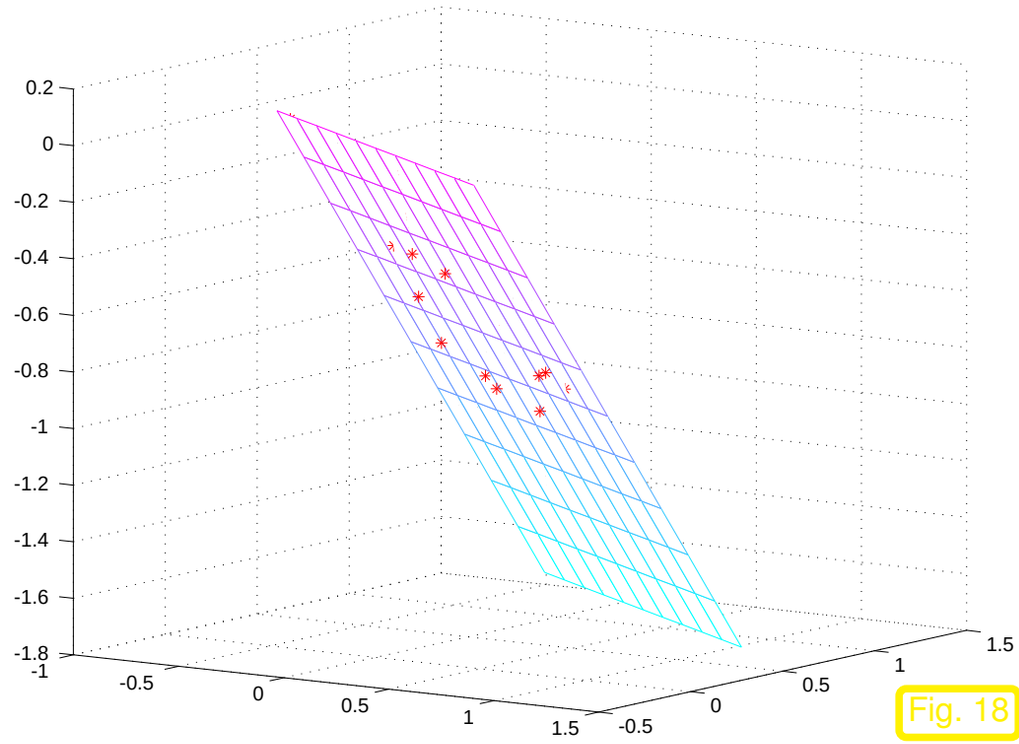


Fig. 18

Principal component analysis for data analysis:

$\mathbf{A} \in \mathbb{R}^{m,n}$, $m \gg n$:

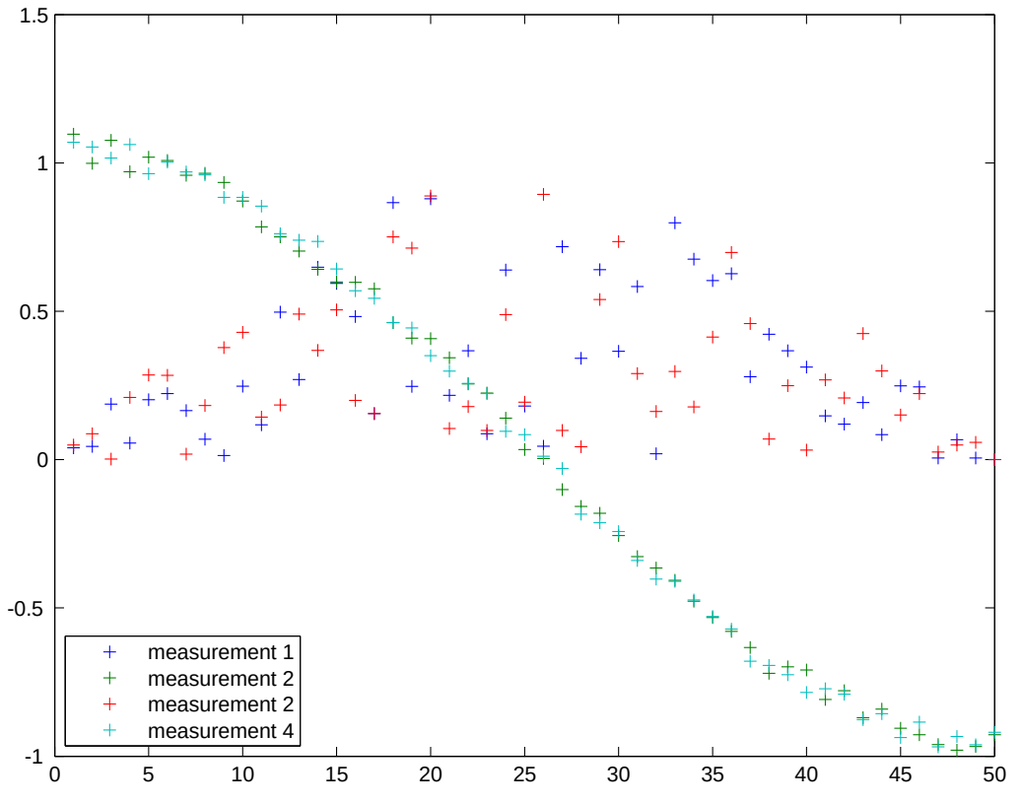
Columns $\mathbf{A}$ $\rightarrow$ series of measurements at different times/locations etc.
Rows of $\mathbf{A}$ $\rightarrow$ measured values corresponding to one time/location etc.

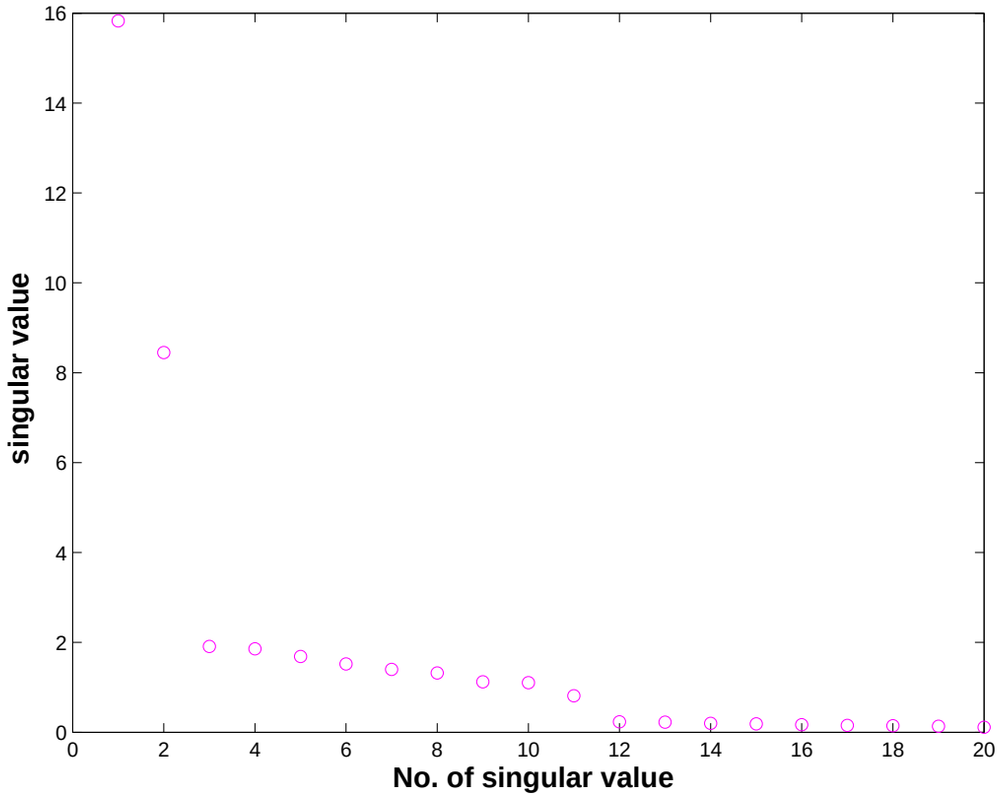
Goal: detect linear correlations

Concrete: two quantities measured over one year at 10 different sites

(Of course, measurements affected by errors/fluctuations)

```
n = 10;
m = 50;
x = sin(pi*(1:m)'/m);
y = cos(pi*(1:m)'/m);
A = [];
for i = 1:n
  A = [A, x.*rand(m,1),...
       y+0.1*rand(m,1)];
end
```





← distribution of singular values of matrix

two dominant singular values !

measurements display linear correlation with **two** principal components

principal components = $\mathbf{U}_{:,1}$, $\mathbf{U}_{:,2}$ (leftmost columns of $\mathbf{U}$-matrix of SVD)

their relative weights = $\mathbf{V}_{:,1}$, $\mathbf{V}_{:,2}$ (leftmost columns of $\mathbf{V}$-matrix of SVD)

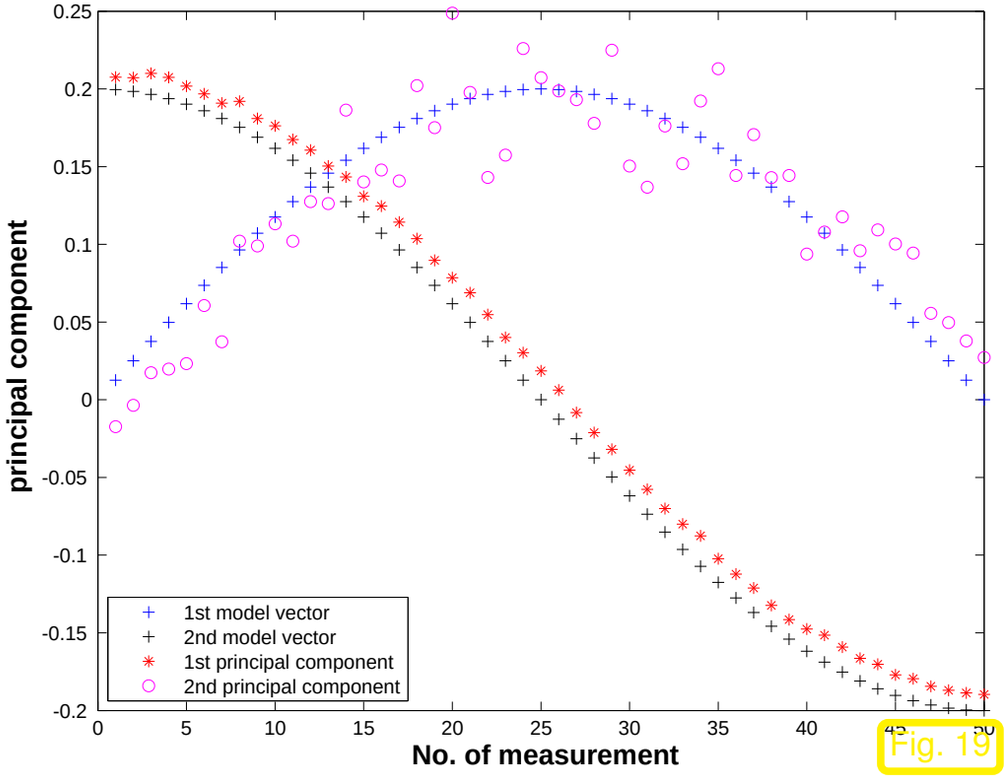

principal component
No. of measurement
1st model vector
2nd model vector
1st principal component
2nd principal component

Fig. 19

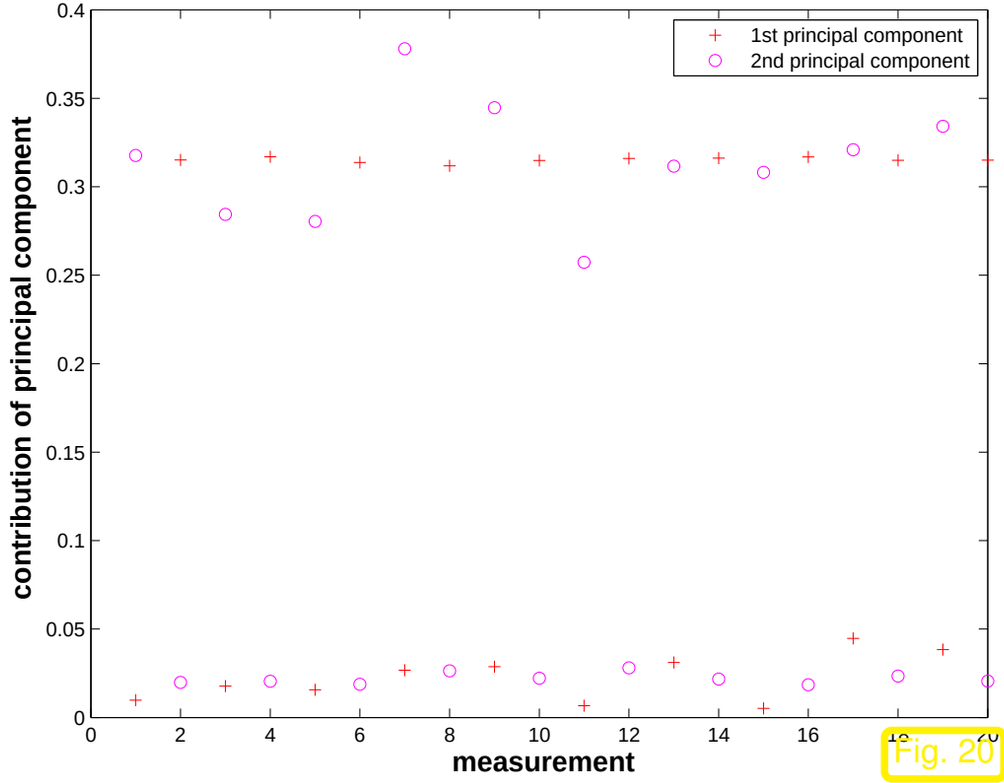

contribution of principal component
measurement
1st principal component
2nd principal component

Fig. 20