MATLAB PCG algorithm

```
function x = pcg(Afun,b,tol,maxit,Binvfun,x0)
x = x0; r = b - feval(Afun,x); rho = 1;
for i = 1 :  maxit
  y = feval(Binvfun,r);
  rho1 = rho; rho = r' * y;
  if (i == 1)
    p = y;
  else
    beta = rho / rho1;
    p = y + beta * p;
  end
  q = feval(Afun,p);
  alpha = rho /(p' * q);
  x = x + alpha * p;
  if (norm(b - evalf(Afun,x)) <= tolb*norm(b)), return; end
  r = r - alpha * q;
end
```

Dubious termination criterion !

△

Summary:

Advantages of Krylov methods vs. direct elimination (, **IF** they converge at all/sufficiently fast).

- They require system matrix $\mathbf{A}$ in procedural form `y=evalA(x)` $\leftrightarrow \mathbf{y} = \mathbf{A}\mathbf{x}$ only.
- They can perfectly exploit sparsity of system matrix.
- They can cash in on low accuracy requirements (, **IF** viable termination criterion available).
- They can benefit from a good initial guess.

4.4 Essential Skills Learned in Chapter 4

You should know:

- the relation between a linear system of equations with a s.p.d. matrix and a quadratic minimization problem
- how the steepest descendent method works and its convergence properties
- the idea behind the conjugate gradient method and its convergence properties
- how to use the Matlab-function pcg
- the importance of the preconditioner

4.3 p. 277

4.4 p. 278

5 Eigenvalues

Example 5.0.1 (Analytic solution of homogeneous linear ordinary differential equations). → [47, Remark 5.6.1]

Autonomous homogeneous linear ordinary differential equation (ODE):

$$\dot{\mathbf{y}} = \mathbf{A}\mathbf{y} \quad , \quad \mathbf{A} \in \mathbb{C}^{n,n} \, . \tag{5.0.1}$$

$$\mathbf{A} = \mathbf{S} \underbrace{\begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}}_{=:\mathbf{D}} \mathbf{S}^{-1} \, , \quad \mathbf{S} \in \mathbb{C}^{n,n} \text{ regular} \implies \left(\dot{\mathbf{y}} = \mathbf{A}\mathbf{y} \overset{\mathbf{z}=\mathbf{S}^{-1}\mathbf{y}}{\longleftrightarrow} \dot{\mathbf{z}} = \mathbf{D}\mathbf{z} \right) .$$

➢ solution of initial value problem:

$$\dot{\mathbf{y}} = \mathbf{A}\mathbf{y} \, , \quad \mathbf{y}(0) = \mathbf{y}_0 \in \mathbb{C}^n \quad \Rightarrow \quad \mathbf{y}(t) = \mathbf{S}\mathbf{z}(t) \, , \quad \dot{\mathbf{z}} = \mathbf{D}\mathbf{z} \, , \quad \mathbf{z}(0) = \mathbf{S}^{-1}\mathbf{y}_0 \, .$$

5.0 p. 27

The initial value problem for the *decoupled* homogeneous linear ODE $\dot{\mathbf{z}} = \mathbf{D}\mathbf{z}$ has a simple analytic solution

$$\mathbf{z}_i(t) = \exp(\lambda_i t)(\mathbf{z}_0)_i = \exp(\lambda_i t)\left((\mathbf{S}^{-1})^T_{i,:}\mathbf{y}_0\right) \, .$$

In light of Rem. **??**:

$$\mathbf{A} = \mathbf{S}\begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}\mathbf{S}^{-1} \quad \Leftrightarrow \quad \mathbf{A}\left((\mathbf{S})_{:,i}\right) = \lambda_i \left((\mathbf{S})_{:,i}\right) \quad i = 1, \ldots, n \, . \tag{5.0.2}$$

In order to find the transformation matrix $\mathbf{S}$ all non-zero solution vectors (= eigenvectors) $\mathbf{x} \in \mathbb{C}^n$ of the linear eigenvalue problem

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

have to be found.

◇

Example 5.0.2 (Normal mode analysis). → [**?**]

In the computation of the **IR spectra of a molecule** one is interested into the vibrational frequencies of a molecule which is described by n positional degrees of freedom $\mathbf{x} \in \mathbb{R}^n$ and corresponding velocities $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} \in \mathbb{R}^n$.

5.0 p. 28

Suppose all masses are equal with an effective mass m ▶ kinetic energy $K(\mathbf{x},\dot{\mathbf{x}}) = \frac{m}{2}\|\dot{\mathbf{x}}\|_2^2$

Model for total potential energy $U(\mathbf{x})$

1. find a local minimum $\mathbf{x}^*$ of the potential energy:

$$DU(\mathbf{x}^*) = 0\,, \quad \mathbf{H} = D^2U(\mathbf{x}^*) \text{ sym. pos. semi-def.}\,,$$

hence with Taylor we have near the local minimum: find a local minimum $\mathbf{x}^*$ of the potential energy:

$$U(\mathbf{x}^*+\mathbf{a}) = U(\mathbf{x}^*) + \frac{1}{2}\mathbf{a}^T\mathbf{H}\mathbf{a}\,.$$

2. Newton'mechanics near the local minimum:

$$m\ddot{\mathbf{x}} = m\ddot{\mathbf{a}} = -D_{\mathbf{a}}U(\mathbf{x}^*+\mathbf{a})\,.$$

As we are around the minimum:

$$m\ddot{\mathbf{a}} = -D_{\mathbf{a}}(U(\mathbf{x}^*) + \frac{1}{2}\mathbf{a}^T\mathbf{H}\mathbf{a}) = -\mathbf{H}\mathbf{a}\,.$$

3. As $\mathbf{H}$ is real and symmetric, its eigenvectors $\mathbf{w}^j$, with $j = 1,\ldots,n$ are orthogonal and hence they form a conveninent basis for representing any vector:

$$\mathbf{a} = c_1(t)\mathbf{w}^1 + \ldots + c_n(t)\mathbf{w}^n\,.$$

Inserting into the system of second-order ODEs, we get:

$$m(\ddot{c_1}\mathbf{w}^1 + \ldots + \ddot{c_n}\mathbf{w}^n) = -(c_1\mathbf{H}\mathbf{w}^1 + \ldots + c_n\mathbf{H}\mathbf{w}^n) = -(\lambda_1\mathbf{w}^1 + \ldots + \lambda_n\mathbf{w}^n)$$

where we denoted the associated eigenvalues by λ_j: $\mathbf{H}\mathbf{w}^j = \lambda_j\mathbf{w}^j$.

4. Taking the scalar product with the eigenvector $\mathbf{w}^k$ we obtain an uncoupled set of equations for each $c_k(t)$, $k = 1,\ldots,n$:

$$m\ddot{c_k} = -\lambda_k c_k\,.$$

5. Looking for solutions of the form $c_k = \alpha_k\sin(\omega_k t)$ and substituting it into the differential equation one get the **angular vibrational frequency of the normal mode** $\omega_k = \sqrt{\lambda_k/m}$

6. In case of different masses we end with the system

$$\mathbf{M}\ddot{\mathbf{a}} = -\mathbf{H}\mathbf{a}\,.$$

with the mass matrix $\mathbf{M}$ which is symmetric, positiv-definite, but not necessarily diagonal. We thus must perform normal mode analysis on the matrix $\mathbf{M}^{-1}\mathbf{H}$:

$$\mathbf{M}^{-1}\mathbf{H}\mathbf{w}^j = \lambda_j\mathbf{w}^j \iff \mathbf{H}\mathbf{w}^j = \lambda_j\mathbf{M}\mathbf{w}^j\,.$$

◇

5.1 Theory of eigenvalue problems

Definition 5.1.1 (Eigenvalues and eigenvectors).

- $\lambda \in \mathbb{C}$ *eigenvalue* (ger.: *Eigenwert*) *of* $\mathbf{A} \in \mathbb{K}^{n,n}$ $:\Leftrightarrow$ $\underbrace{\det(\lambda\mathbf{I}-\mathbf{A})}_{\text{characteristic polynomial } \chi(\lambda)} = 0$
- *spectrum of* $\mathbf{A} \in \mathbb{K}^{n,n}$: $\sigma(\mathbf{A}) := \{\lambda \in \mathbb{C}\text{: } \lambda \text{ eigenvalue of } \mathbf{A}\}$

Two simple facts:

$$\lambda \in \sigma(\mathbf{A}) \quad \Rightarrow \quad \dim \mathrm{Eig}_{\mathbf{A}}(\lambda) > 0\,, \tag{5.1.1}$$

$$\det(\mathbf{A}) = \det(\mathbf{A}^T) \quad \forall \mathbf{A} \in \mathbb{K}^{n,n} \quad \Rightarrow \quad \sigma(\mathbf{A}) = \sigma(\mathbf{A}^T)\,. \tag{5.1.2}$$

✎ notation: $\rho(\mathbf{A}) := \max\{|\lambda|\text{: } \lambda \in \sigma(\mathbf{A})\} \hat{=}$ spectral radius of $\mathbf{A} \in \mathbb{K}^{n,n}$

Theorem 5.1.2 (Bound for spectral radius).
For any matrix norm $\|\cdot\|$ induced by a vector norm (→ Def. 2.4.2)

$$\rho(\mathbf{A}) \le \|\mathbf{A}\|\,.$$

Lemma 5.1.3 (Gershgorin circle theorem). *For any $\mathbf{A} \in \mathbb{K}^{n,n}$ holds true*

$$\sigma(\mathbf{A}) \subset \bigcup_{j=1}^{n}\{z \in \mathbb{C}\text{: } |z-a_{jj}| \le \sum_{i\neq j}|a_{ji}|\}\,.$$

Lemma 5.1.4 (Similarity and spectrum)**.**
The spectrum of a matrix is invariant with respect to similarity transformations:

$$\forall \mathbf{A} \in \mathbb{K}^{n,n}: \quad \sigma(\mathbf{S}^{-1}\mathbf{A}\mathbf{S}) = \sigma(\mathbf{A}) \quad \forall \text{ regular } \mathbf{S} \in \mathbb{K}^{n,n} .$$

Lemma 5.1.5. *Existence of a one-dimensional invariant subspace*

$$\forall \mathbf{C} \in \mathbb{C}^{n,n}: \quad \exists \mathbf{u} \in \mathbb{C}^n: \quad \mathbf{C}\left(\operatorname{Span}\{\mathbf{u}\}\right) \subset \operatorname{Span}\{\mathbf{u}\} .$$

Theorem 5.1.6 (Schur normal form)**.**

$$\forall \mathbf{A} \in \mathbb{K}^{n,n}: \quad \exists \mathbf{U} \in \mathbb{C}^{n,n} \text{ unitary}: \quad \mathbf{U}^H\mathbf{A}\mathbf{U} = \mathbf{T} \quad \text{with } \mathbf{T} \in \mathbb{C}^{n,n} \text{ upper triangular} .$$

Corollary 5.1.7 (Principal axis transformation)**.**

$$\mathbf{A} \in \mathbb{K}^{n,n},\ \mathbf{A}\mathbf{A}^H = \mathbf{A}^H\mathbf{A}: \quad \exists \mathbf{U} \in \mathbb{C}^{n,n} \text{ unitary}: \quad \mathbf{U}^H\mathbf{A}\mathbf{U} = \operatorname{diag}(\lambda_1, \ldots, \lambda_n), \quad \lambda_i \in \mathbb{C} .$$

A matrix $\mathbf{A} \in \mathbb{K}^{n,n}$ with $\mathbf{A}\mathbf{A}^H = \mathbf{A}^H\mathbf{A}$ is called normal.

Examples of normal matrices are
- Hermitian matrices: $\mathbf{A}^H = \mathbf{A}$ ➤ $\sigma(\mathbf{A}) \subset \mathbb{R}$
- unitary matrices: $\mathbf{A}^H = \mathbf{A}^{-1}$ ➤ $|\sigma(\mathbf{A})| = 1$
- skew-Hermitian matrices: $\mathbf{A} = -\mathbf{A}^H$ ➤ $\sigma(\mathbf{A}) \subset i\mathbb{R}$

➤

Normal matrices can be diagonalized by *unitary* similarity transformations

Symmetric real matrices can be diagonalized by *orthogonal* similarity transformations

In Thm. 5.1.7:
- $\lambda_1, \ldots, \lambda_n$ = eigenvalues of $\mathbf{A}$
- Columns of $\mathbf{U}$ = orthonormal basis of eigenvectors of $\mathbf{A}$

Eigenvalue problems: (EVPs)
- ❶ Given $\mathbf{A} \in \mathbb{K}^{n,n}$ find **all** eigenvalues (= spectrum of $\mathbf{A}$).
- ❷ Given $\mathbf{A} \in \mathbb{K}^{n,n}$ find $\sigma(\mathbf{A})$ plus **all** eigenvectors.
- ❸ Given $\mathbf{A} \in \mathbb{K}^{n,n}$ find **a few** eigenvalues and associated eigenvectors

(Linear) generalized eigenvalue problem:

Given $\mathbf{A} \in \mathbb{C}^{n,n}$, regular $\mathbf{B} \in \mathbb{C}^{n,n}$, seek $\mathbf{x} \neq 0$, $\lambda \in \mathbb{C}$

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{B}\mathbf{x} \quad \Leftrightarrow \quad \mathbf{B}^{-1}\mathbf{A}\mathbf{x} = \lambda\mathbf{x} . \tag{5.1.3}$$

$\mathbf{x} \hat{=}$ generalized eigenvector, $\lambda \hat{=}$ generalized eigenvalue

Obviously every generalized eigenvalue problem is equivalent to a standard eigenvalue problem

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{B}\mathbf{x} \quad \Leftrightarrow \quad \mathbf{B}^{-1}\mathbf{A} = \lambda\mathbf{x} .$$

However, usually it is not advisable to use this equivalence for numerical purposes!

Remark 5.1.1 (Generalized eigenvalue problems and Cholesky factorization).

If $\mathbf{B} = \mathbf{B}^H$ s.p.d. (→ Def. 2.6.2) with Cholesky factorization $\mathbf{B} = \mathbf{R}^H\mathbf{R}$

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{B}\mathbf{x} \quad \Leftrightarrow \quad \widetilde{\mathbf{A}}\mathbf{y} = \lambda\mathbf{y} \quad \text{where } \widetilde{\mathbf{A}} := \mathbf{R}^{-H}\mathbf{A}\mathbf{R}^{-1},\ \mathbf{y} := \mathbf{R}\mathbf{x} .$$

➞ This transformation can be used for efficient computations.

5.2 “Direct” Eigensolvers

Purpose: solution of eigenvalue problems ❶, ❷ for dense matrices “up to machine precision”

MATLAB-function: `eig`

`d = eig(A)` : computes spectrum $\sigma(\mathbf{A}) = \{d_1, \ldots, d_n\}$ of $\mathbf{A} \in \mathbb{C}^{n,n}$
`[V,D] = eig(A)` : computes $\mathbf{V} \in \mathbb{C}^{n,n}$, *diagonal* $\mathbf{D} \in \mathbb{C}^{n,n}$ such that $\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{D}$

Remark 5.2.1 (QR-Algorithm). → [20, Sect. 7.5]

Note: All “direct” eigensolvers are iterative methods

Idea: Iteration based on successive **unitary** similarity transformations

$\mathbf{A} = \mathbf{A}^{(0)} \longrightarrow \mathbf{A}^{(1)} \longrightarrow \ldots \longrightarrow$ { diagonal matrix , if $\mathbf{A} = \mathbf{A}^H$; upper triangular matrix , else. (→ Thm. 5.1.6)

(superior stability of unitary transformations, see Rem. **??**)

Code 5.2.2: QR-algorithm with shift

```
function d = eigqr(A,tol)
n = size(A,1);
while (norm(tril(A,-1)) > tol*norm(A))
  shift = A(n,n);
  [Q,R] = qr( A - shift * eye(n));
  A = Q'*A*Q;
end
d = diag(A);
```

- QR-algorithm (with shift)
  - in general: quadratic convergence
  - cubic convergence for normal matrices

  (→ [20, Sect. 7.5,8.2])

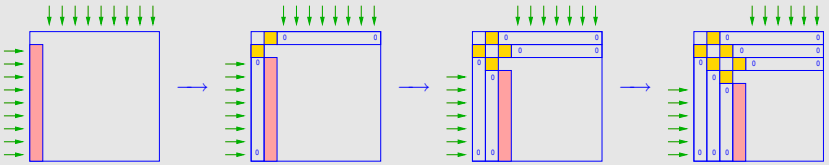
Computational cost: $O(n^3)$ operations per step of the QR-algorithm

Library implementations of the QR-algorithm provide *numerically stable* eigensolvers (→ Def.**??**)

Remark 5.2.3 (Unitary similarity transformation to tridiagonal form).

Successive Householder similarity transformations of $\mathbf{A} = \mathbf{A}^H$:

(→ ≙ affected rows/columns, ▭ ≙ targeted vector)



- transformation to tridiagonal form **!** (for general matrices a similar strategy can achieve a similarity transformation to upper Hessenberg form)

- this transformation is used as a preprocessing step for QR-algorithm ➢ `eig`.

Similar functionality for generalized EVP $\mathbf{Ax} = \lambda\mathbf{Bx}$, $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{n,n}$

`d = eig(A,B)` : computes all generalized eigenvalues

`[V,D] = eig(A,B)` : computes $\mathbf{V} \in \mathbb{C}^{n,n}$, *diagonal* $\mathbf{D} \in \mathbb{C}^{n,n}$ such that $\mathbf{AV} = \mathbf{BVD}$

Note: (Generalized) eigenvectors can be recovered as columns of $\mathbf{V}$:

$$\mathbf{AV} = \mathbf{VD} \quad \Leftrightarrow \quad \mathbf{A}(\mathbf{V})_{:,i} = (\mathbf{D})_{i,i}\mathbf{V}_{:,i} \,,$$

if $\mathbf{D} = \operatorname{diag}(d_1, \ldots, d_n)$.

Remark 5.2.4 (Computational effort for eigenvalue computations).

Computational effort (#elementary operations) for `eig()`:

eigenvalues & eigenvectors of $\mathbf{A} \in \mathbb{K}^{n,n}$	$\sim 25n^3 + O(n^2)$
only eigenvalues of $\mathbf{A} \in \mathbb{K}^{n,n}$	$\sim 10n^3 + O(n^2)$
eigenvalues and eigenvectors $\mathbf{A} = \mathbf{A}^H \in \mathbb{K}^{n,n}$	$\sim 9n^3 + O(n^2)$
only eigenvalues of $\mathbf{A} = \mathbf{A}^H \in \mathbb{K}^{n,n}$	$\sim \frac{4}{3}n^3 + O(n^2)$
only eigenvalues of tridiagonal $\mathbf{A} = \mathbf{A}^H \in \mathbb{K}^{n,n}$	$\sim 30n^2 + O(n)$

} $O(n^3)$!

Note: `eig` not available for sparse matrix arguments

Exception: `d=eig(A)` for sparse *Hermitian* matrices

△

Example 5.2.5 (Runtimes of `eig`).

Code 5.2.6: measuring runtimes of `eig`

```
function eigtiming

A = rand(500,500); B = A'*A;
C = gallery('tridiag',500,1,3,1);
times = [];
for n=5:5:500
  An = A(1:n,1:n); Bn = B(1:n,1:n); Cn = C(1:n,1:n);
  t1 = 1000; for k=1:3, tic; d = eig(An); t1 = min(t1,toc); end
  t2 = 1000; for k=1:3, tic; [V,D] = eig(An); t2 = min(t2,toc); end
  t3 = 1000; for k=1:3, tic; d = eig(Bn); t3 = min(t3,toc); end
  t4 = 1000; for k=1:3, tic; [V,D] = eig(Bn); t4 = min(t4,toc); end
  t5 = 1000; for k=1:3, tic; d = eig(Cn); t5 = min(t5,toc); end
  times = [times; n t1 t2 t3 t4 t5];
end

figure;
loglog(times(:,1),times(:,2),'r+', times(:,1),times(:,3),'m*',...
       times(:,1),times(:,4),'cp', times(:,1),times(:,5),'b^',...
       times(:,1),times(:,6),'k.');
xlabel('{\bf matrix size n}','fontsize',14);
ylabel('{\bf time [s]}','fontsize',14);
```

```
title('eig runtimes');
legend('d = eig(A)','[V,D] = eig(A)','d = eig(B)','[V,D] = eig(B)','d = eig(C)',...
       'location','northwest');

print -depsc2 '../PICTURES/eigtimingall.eps'

figure;
loglog(times(:,1),times(:,2),'r+', times(:,1),times(:,3),'m*',...
         times(:,1),(times(:,1).^3)/(times(1,1)^3)*times(1,2),'k-');
xlabel('{\bf matrix size n}','fontsize',14);
ylabel('{\bf time [s]}','fontsize',14);
title('nxn random matrix');
legend('d = eig(A)','[V,D] = eig(A)','O(n^3)','location','northwest');

print -depsc2 '../PICTURES/eigtimingA.eps'

figure;
loglog(times(:,1),times(:,4),'r+', times(:,1),times(:,5),'m*',...
         times(:,1),(times(:,1).^3)/(times(1,1)^3)*times(1,2),'k-');
xlabel('{\bf matrix size n}','fontsize',14);
ylabel('{\bf time [s]}','fontsize',14);
title('nxn random Hermitian matrix');
```



```
legend('d = eig(A)','[V,D] = eig(A)','O(n^3)','location','northwest');

print -depsc2 '../PICTURES/eigtimingB.eps'

figure;
loglog(times(:,1),times(:,6),'r*',...
         times(:,1),(times(:,1).^2)/(times(1,1)^2)*times(1,2),'k-');
xlabel('{\bf matrix size n}','fontsize',14);
ylabel('{\bf time [s]}','fontsize',14);
title('nxn tridiagonel Hermitian matrix');
legend('d = eig(A)','O(n^2)','location','northwest');

print -depsc2 '../PICTURES/eigtimingC.eps'
```

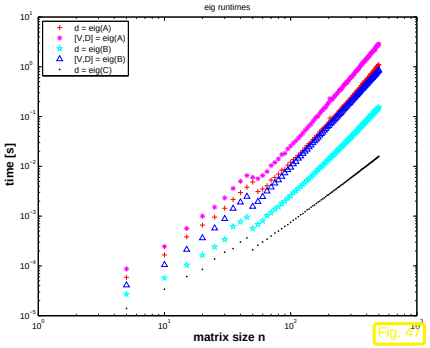



Fig. 47

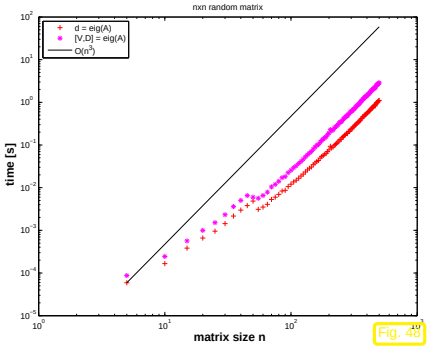


Fig. 48

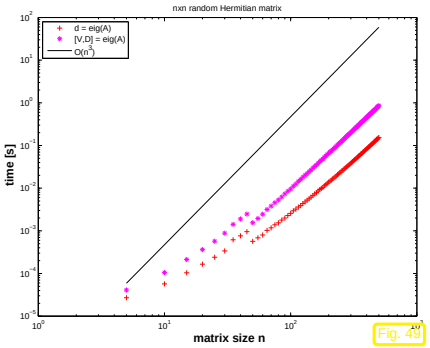


Fig. 49

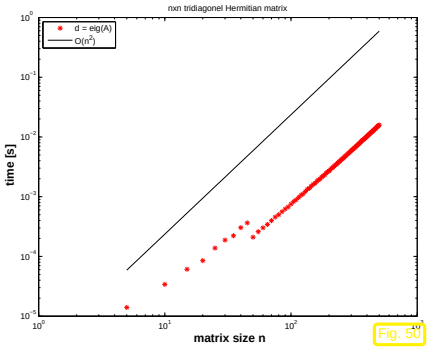


Fig. 50

☛ For the sake of efficiency: think which information you really need when computing eigenvalues/eigenvectors of dense matrices

Potentially more efficient methods for *sparse matrices* will be introduced below in Sects. **??**, 5.3.

◇

5.3 Krylov Subspace Methods

Recall from Sect. 4.2.2 , Lemma 4.2.5:

Residuals $\mathbf{r}_0, \ldots, \mathbf{r}_{m-1}$ generated in CG iteration, Alg. 4.2.1 applied to $\mathbf{A}\mathbf{x} = \mathbf{z}$ with $\mathbf{x}^{(0)} = 0$, provide *orthogonal basis* for $\mathcal{K}_m(\mathbf{A}, \mathbf{z})$ (, if $\mathbf{r}_k \neq 0$).

Inexpensive Ritz projection of $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ onto $\mathcal{K}_m(\mathbf{A}, \mathbf{z})$: orthogonal matrix

$$\mathbf{V}_m^T \mathbf{A} \mathbf{V}_m \mathbf{x} = \lambda \mathbf{x} \ , \quad \mathbf{V}_m := \left(\frac{\mathbf{r}_0}{\|\mathbf{r}_0\|}, \ldots, \frac{\mathbf{r}_{m-1}}{\|\mathbf{r}_{m-1}\|} \right) \in \mathbb{R}^{n,m} \ . \tag{5.3.1}$$

recall: residuals generated by *short recursions*, see Alg. 4.2.1

Lemma 5.3.1 (Tridiagonal Ritz projection from CG residuals)**.**
$\mathbf{V}_m^T \mathbf{A} \mathbf{V}_m$ *is a tridiagonal matrix.*

Proof. Lemma 4.2.5: $\{\mathbf{r}_0, \ldots, \mathbf{r}_{\ell-1}\}$ is an orthogonal basis of $\mathcal{K}_\ell(\mathbf{A}, \mathbf{r}_0)$, if all the residuals are non-zero. As $\mathbf{A}\mathcal{K}_{\ell-1}(\mathbf{A}, \mathbf{r}_0) \subset \mathcal{K}_\ell(\mathbf{A}, \mathbf{r}_0)$, we conclude the orthogonality $\mathbf{r}_m^T \mathbf{A} \mathbf{r}_j$ for all $j = 0, \ldots, m-2$. Since

$$\left(\mathbf{V}_m^T \mathbf{A} \mathbf{V}_m \right)_{ij} = \mathbf{r}_{i-1}^T \mathbf{A} \mathbf{r}_{j-1} \ , \quad 1 \leq i, j \leq m \ ,$$

the assertion of the theorem follows. □

$$\mathbf{V}_l^H \mathbf{A} \mathbf{V}_l = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \\ & & & & \ddots \\ & & & \ddots & \ddots & \beta_{k-1} \\ & & & & \beta_{k-1} & \alpha_k \end{pmatrix} =: \mathbf{T}_l \in \mathbb{K}^{k,k} \quad \text{[tridiagonal matrix]}$$

Algorithm for computing $\mathbf{V}_l$ and $\mathbf{T}_l$:

Lanczos process

Computational effort/step:

- 1× $\mathbf{A}$×vector
- 2 dot products
- 2 AXPY-operations
- 1 division

Code 5.3.1: Lanczos process

```
function [V,alph,bet] = lanczos(A,k,z0)
V = z0/norm(z0);
alph=zeros(k,1);
bet = zeros(k,1);
for j=1:k
  p = A*V(:,j); alph(j) = dot(p,V(:,j));
  w = p - alph(j)*V(:,j);
  if (j > 1), w = w - bet(j-1)*V(:,j-1); end;
  bet(j) = norm(w); V = [V,w/bet(j)];
end
bet = bet(1:end-1);
```

Total computational effort for l steps of Lanczos process, if $\mathbf{A}$ has at most k non-zero entries per row: $O(nkl)$

Note: Code 5.3.0 assumes that no residual vanishes. This could happen, if $\mathbf{z}_0$ exactly belonged to the span of a few eigenvectors. However, in practical computations inevitable round-off errors will always ensure that the iterates do not stay in an invariant subspace of $\mathbf{A}$, *cf.* Rem. **??**.

Convergence (what we expect from the above considerations) → [14, Sect. 8.5])

In l-th step: $\lambda_n \approx \mu_l^{(l)}$, $\lambda_{n-1} \approx \mu_{l-1}^{(l)}, \ldots, \lambda_1 \approx \mu_1^{(l)}$,

$$\sigma(\mathbf{T}_l) = \{\mu_1^{(l)}, \ldots, \mu_l^{(l)}\}, \quad \mu_1^{(l)} \leq \mu_2^{(l)} \leq \cdots \leq \mu_l^{(l)} \ .$$

Example 5.3.2 (Lanczos process for eigenvalue computation)*.*

$\mathbf{A}$ from Ex. **??**

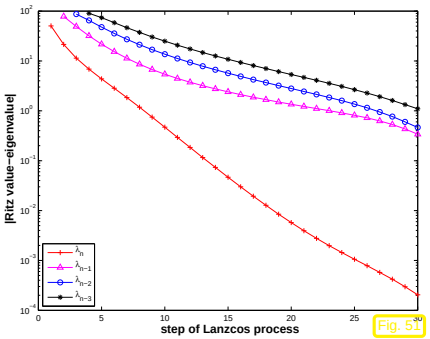


Fig. 51

`A = gallery('minij',100);`

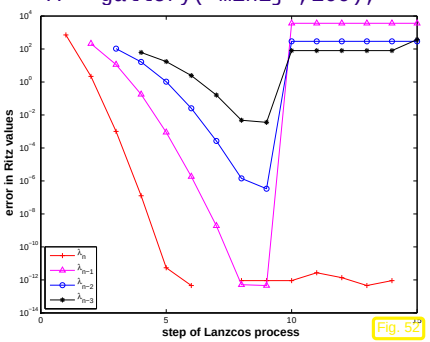


Fig. 52

Observation: same as in Ex. **??**, linear convergence of Ritz values to eigenvalues.

However for $\mathbf{A} \in \mathbb{R}^{10,10}$, $a_{ij} = \min\{i, j\}$ good initial convergence, but sudden "jump" of Ritz values off eigenvalues!

Conjecture: Impact of roundoff errors, *cf.* Ex. 4.2.4

◇

Example 5.3.3 (Impact of roundoff on Lanczos process).

$\mathbf{A} \in \mathbb{R}^{10,10}$, $a_{ij} = \min\{i,j\}$. `A = gallery('minij',10);`

Computed by `[V,alpha,beta] = lanczos(A,n,ones(n,1));`, see Code 5.3.0:

$$\mathbf{T} = \begin{pmatrix}
38.500000 & 14.813845 & & & & & & & & \\
14.813845 & 9.642857 & 2.062955 & & & & & & & \\
 & 2.062955 & 2.720779 & 0.776284 & & & & & & \\
 & & 0.776284 & 1.336364 & 0.385013 & & & & & \\
 & & & 0.385013 & 0.826316 & 0.215431 & & & & \\
 & & & & 0.215431 & 0.582380 & 0.126781 & & & \\
 & & & & & 0.126781 & 0.446860 & 0.074650 & & \\
 & & & & & & 0.074650 & 0.363803 & 0.043121 & \\
 & & & & & & & 0.043121 & 3.820888 & 11.991094 \\
 & & & & & & & & 11.991094 & 41.254286
\end{pmatrix}$$

$\sigma(\mathbf{A}) = \{0.255680, 0.273787, 0.307979, 0.366209, 0.465233, 0.643104, 1.000000, 1.873023, 5.048917, 44.766069\}$

$\sigma(\mathbf{T}) = \{0.263867, 0.303001, 0.365376, 0.465199, 0.643104, 1.000000, 1.873023, 5.048917, 44.765976, 44.766069\}$

► Uncanny cluster of computed eigenvalues of $\mathbf{T}$ ("ghost eigenvalues", [20, Sect. 9.2.5])

$$\mathbf{V}^H\mathbf{V} = \begin{pmatrix}
1.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000251 & 0.258801 & 0.883711 \\
0.000000 & 1.000000 & -0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000106 & 0.109470 & 0.373799 \\
0.000000 & -0.000000 & 1.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000005 & 0.005373 & 0.018347 \\
0.000000 & 0.000000 & 0.000000 & 1.000000 & -0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000096 & 0.000328 \\
0.000000 & 0.000000 & 0.000000 & -0.000000 & 1.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000001 & 0.000003 \\
0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 & 1.000000 & -0.000000 & 0.000000 & 0.000000 & 0.000000 \\
0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 & -0.000000 & 1.000000 & -0.000000 & 0.000000 & 0.000000 \\
0.000251 & 0.000106 & 0.000005 & 0.000000 & 0.000000 & 0.000000 & -0.000000 & 1.000000 & -0.000000 & 0.000000 \\
0.258801 & 0.109470 & 0.005373 & 0.000096 & 0.000001 & 0.000000 & 0.000000 & -0.000000 & 1.000000 & 0.000000 \\
0.883711 & 0.373799 & 0.018347 & 0.000328 & 0.000003 & 0.000000 & 0.000000 & 0.000000 & 0.000000 & 1.000000
\end{pmatrix}$$

► Loss of orthogonality of residual vectors due to roundoff
(compare: impact of roundoff on CG iteration, Ex. 4.2.4

l	$\sigma(\mathbf{T}_l)$									
1										38.500000
2									3.392123	44.750734
3								1.117692	4.979881	44.766064
4							0.597664	1.788008	5.048259	44.766069
5						0.415715	0.925441	1.870175	5.048916	44.766069
6					0.336507	0.588906	0.995299	1.872997	5.048917	44.766069
7				0.297303	0.431779	0.638542	0.999922	1.873023	5.048917	44.766069
8			0.276160	0.349724	0.462449	0.643016	1.000000	1.873023	5.048917	44.766069
9		0.276035	0.349451	0.462320	0.643006	1.000000	1.873023	3.821426	5.048917	44.766069
10	0.263867	0.303001	0.365376	0.465199	0.643104	1.000000	1.873023	5.048917	44.765976	44.766069

◇

Idea:
- do not rely on orthogonality relations of Lemma 4.2.5
- use explicit Gram-Schmidt orthogonalization

Details: inductive approach: given $\{\mathbf{v}_1, \ldots, \mathbf{v}_l\}$ ONB of $\mathcal{K}_l(\mathbf{A}, \mathbf{z})$

► $\widetilde{\mathbf{v}}_{l+1} := \mathbf{A}\mathbf{v}_l - \sum_{j=1}^{l} (\mathbf{v}_j^H \mathbf{A}\mathbf{v}_l)\, \mathbf{v}_j$, $\mathbf{v}_{l+1} := \frac{\widetilde{\mathbf{v}}_{l+1}}{\|\widetilde{\mathbf{v}}_{l+1}\|_2}$ $\Rightarrow$ $\mathbf{v}_{l+1} \perp \mathcal{K}_l(\mathbf{A}, \mathbf{z})$. (5.3.2)

(Gram-Schmidt, *cf.* (4.2.6)) orthogonal

► Arnoldi process

In step l:

$1\times$	$\mathbf{A}\times$vector
$l+1$	dot products
l	AXPY-operations
n	divisions

➢ Computational cost for l steps, if at most k non-zero entries in each row of $\mathbf{A}$: $O(nkl^2)$

`H(l+1,l) = 0` → STOP !

Code 5.3.4: Arnoldi process

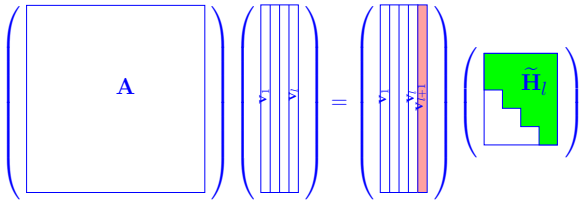
```
function [V,H] = arnoldi(A,k,v0)
V = [v0/norm(v0)];
H = zeros(k+1,k);
for l=1:k
  vt = A*V(:,l);
  for j=1:l
    H(j,l) = dot(V(:,j),vt);
    vt = vt - H(j,l)*V(:,j);
  end
  H(l+1,l) = norm(vt);
  if (H(l+1,l) == 0), break; end
  V = [V, vt/H(l+1,l)];
end
```

If it does not stop prematurely, the Arnoldi process of Code 5.3.3 will yield an *orthonormal basis* (OBN) of $\mathcal{K}_{k+1}(\mathbf{A}, \mathbf{v}_0)$ for a **general** $\mathbf{A} \in \mathbb{C}^{n,n}$.

Algebraic view of the Arnoldi process of Code 5.3.3, meaning of output H:

$$\mathbf{V}_l = [\mathbf{v}_1, \ldots, \mathbf{v}_l] \ : \ \mathbf{A}\mathbf{V}_l = \mathbf{V}_{l+1}\widetilde{\mathbf{H}}_l \quad , \quad \widetilde{\mathbf{H}}_l \in \mathbb{K}^{l+1,l} \text{ mit } \widetilde{h}_{ij} = \begin{cases} \mathbf{v}_i^H \mathbf{A}\mathbf{v}_j & \text{, if } i \le j \text{ ,} \\ \|\widetilde{\mathbf{v}}_i\|_2 & \text{, if } i = j+1 \text{ ,} \\ 0 & \text{else.} \end{cases}$$

➡ $\widetilde{\mathbf{H}}_l$ = non-square upper Hessenberg matrices



Translate Code 5.3.3 to matrix calculus:

Lemma 5.3.2 (Theory of Arnoldi process)**.**
For the matrices $\mathbf{V}_l \in \mathbb{K}^{n,l}$, $\widetilde{\mathbf{H}}_l \in \mathbb{K}^{l+1,l}$ arising in the l-th step, $l \le n$, of the Arnoldi process holds

(i) $\mathbf{V}_l^H \mathbf{V}_l = \mathbf{I}$ *(unitary matrix),*

(ii) $\mathbf{A}\mathbf{V}_l = \mathbf{V}_{l+1}\widetilde{\mathbf{H}}_l$, $\widetilde{\mathbf{H}}_l$ *is non-square upper Hessenberg matrix,*

(iii) $\mathbf{V}_l^H \mathbf{A}\mathbf{V}_l = \mathbf{H}_l \in \mathbb{K}^{l,l}$, $h_{ij} = \widetilde{h}_{ij}$ *for* $1 \le i, j \le l$,

(iv) If $\mathbf{A} = \mathbf{A}^H$ *then* $\mathbf{H}_l$ *is tridiagonal (➢ Lanczos process)*

Proof. Direct from Gram-Schmidt orthogonalization and inspection of Code 5.3.3. □

Remark 5.3.5 (Arnoldi process and Ritz projection).

Interpretation of Lemma 5.3.2 (iii) & (i):

$\mathbf{H}_l\mathbf{x} = \lambda\mathbf{x}$ is a (generalized) Ritz projection of EVP $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$

△

Eigenvalue approximation for general EVP $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ by Arnoldi process:

In l-th step: $\lambda_n \approx \mu_l^{(l)}$, $\lambda_{n-1} \approx \mu_{l-1}^{(l)}, \ldots, \lambda_1 \approx \mu_1^{(l)}$,
$\sigma(\mathbf{H}_l) = \{\mu_1^{(l)}, \ldots, \mu_l^{(l)}\}$, $|\mu_1^{(l)}| \le |\mu_2^{(l)}| \le \cdots \le |\mu_l^{(l)}|$.

Code 5.3.6: Arnoldi eigenvalue approximation

```
function [dn,V,Ht] = arnoldieig(A,v0,k,tol)
n = size(A,1); V = [v0/norm(v0)];
H = zeros(1,0); dn = zeros(k,1);
for l=1:n
  d = dn;
  Ht = [Ht, zeros(l,1); zeros(1,l)];
  vt = A*V(:,l);
  for j=1:l
    Ht(j,l) = dot(V(:,j),vt);
    vt = vt - Ht(j,l)*V(:,j);
  end
  ev = sort(eig(Ht(1:l,1:l)));
  dn(1:min(l,k)) = ev(end:-1:end-min(l,k)+1);
  if (norm(d-dn) < tol*norm(dn)), break; end;
  Ht(l+1,l) = norm(vt);
  V = [V, vt/Ht(l+1,l)];
end
```

Arnoldi process for computing the k largest (in modulus) eigenvalues of $\mathbf{A} \in \mathbb{C}^{n,n}$

1 $\mathbf{A}\times$vector per step (➢ attractive for sparse matrices)

However: required storage increases with number of steps, *cf.* situation with GMRES, Sect. **??**.

Heuristic termination criterion

Example 5.3.7 (Stabilty of Arnoldi process).

$\mathbf{A} \in \mathbb{R}^{100,100}$, $a_{ij} = \min\{i, j\}$.

```
A = gallery('minij',100);
```

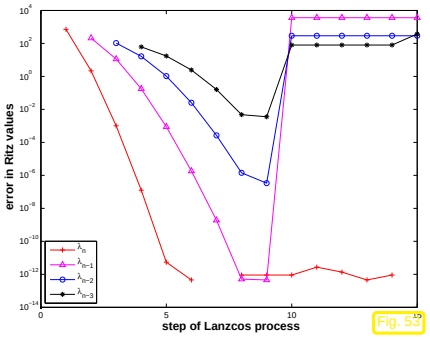


Fig. 53

Lanczos process: Ritz values

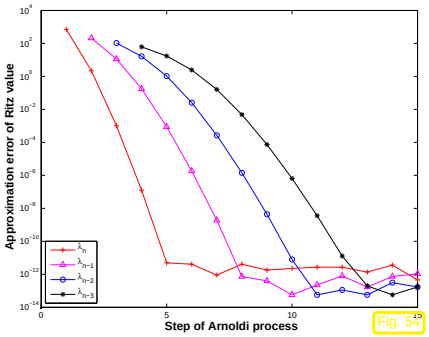


Fig. 54

Arnoldi process: Ritz values

Ritz values during Arnoldi process for `A = gallery('minij',10);` ↔ Ex. 5.3.2

l	$\sigma(\mathbf{H}_l)$									
1										38.500000
2									3.392123	44.750734
3								1.117692	4.979881	44.766064
4							0.597664	1.788008	5.048259	44.766069
5						0.415715	0.925441	1.870175	5.048916	44.766069
6					0.336507	0.588906	0.995299	1.872997	5.048917	44.766069
7				0.297303	0.431779	0.638542	0.999922	1.873023	5.048917	44.766069
8			0.276159	0.349722	0.462449	0.643016	1.000000	1.873023	5.048917	44.766069
9		0.263872	0.303009	0.365379	0.465199	0.643104	1.000000	1.873023	5.048917	44.766069
10	0.255680	0.273787	0.307979	0.366209	0.465233	0.643104	1.000000	1.873023	5.048917	44.766069

Observation: (almost perfect approximation of spectrum of $\mathbf{A}$)

For the above examples both the Arnoldi process and the Lanczos process are *algebraically equivalent*, because they are applied to a symmetric matrix $\mathbf{A} = \mathbf{A}^T$. However, they behave strikingly differently, which indicates that they are *not numerically equivalent*.

The Arnoldi process is much less affected by roundoff than the Lanczos process, because it does not take for granted orthogonality of the "residual vector sequence". Hence, the Arnoldi process enjoys superior numerical stability (→ Sect. **??**, Def. **??**) compared to the Lanczos process. ◇

Example 5.3.8 (Eigenvalue computation with Arnoldi process).

Eigenvalue approximation from Arnoldi process for *non-symmetric* $\mathbf{A}$, initial vector `ones(100,1);`

```
n=100;
M=full(gallery('tridiag',-0.5*ones(n-1,1),2*ones(n,1),-1.5*ones(n-1,1)));
A=M*diag(1:n)*inv(M);
```

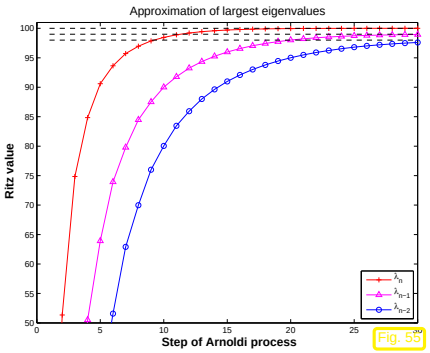


Fig. 55

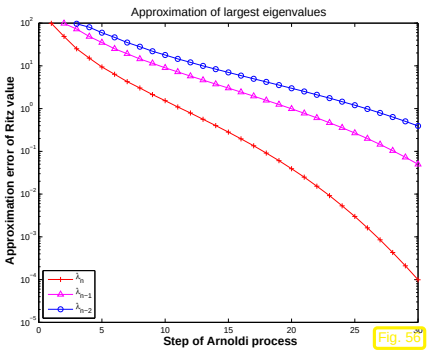


Fig. 56

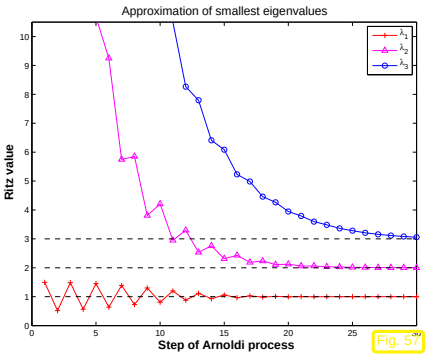


Fig. 57

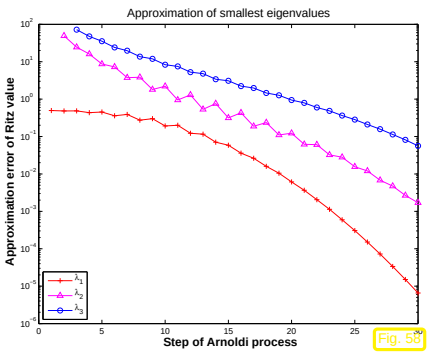


Fig. 58

Observation: "vaguely linear" convergence of largest and smallest eigenvalues, *cf.* Ex. **??**. ◇

Krylov subspace iteration methods (= Arnoldi process, Lanczos process) attractive for computing *a few* of the largest/smallest eigenvalues and associated eigenvectors of *large sparse matrices*.

Remark 5.3.9 (Krylov subspace methods for generalized EVP).

Adaptation of Krylov subspace iterative eigensolvers to generalized EVP: $\mathbf{Ax} = \lambda\mathbf{Bx}$, $\mathbf{B}$ s.p.d.: replace Euclidean inner product with "$\mathbf{B}$-inner product" $(\mathbf{x},\mathbf{y}) \mapsto \mathbf{x}^H\mathbf{By}$.

△

MATLAB-functions:

`d = eigs(A,k,sigma)` : k largest/smallest eigenvalues of $\mathbf{A}$

`d = eigs(A,B,k,sigma)`: k largest/smallest eigenvalues for generalized EVP $\mathbf{Ax} = \lambda\mathbf{Bx}$,$\mathbf{B}$ s.p.d.

`d = eigs(Afun,n,k)` : `Afun` = handle to function providing matrix×vector for $\mathbf{A}$/$\mathbf{A}^{-1}$/$\mathbf{A}-\alpha\mathbf{I}$/$(\mathbf{A}-\alpha\mathbf{B})^{-1}$. (Use flags to tell `eigs` about special properties of matrix behind `Afun`.)

`eigs` just calls routines of the open source ARPACK numerical library.

5.4 Singular Value Decomposition

Remark 5.4.1 (Principal component analysis (PCA)).

Given: n data points $\mathbf{a}_j \in \mathbb{R}^m$, $j = 1,\ldots,n$, in m-dimensional (feature) space

Conjectured: "linear dependence": $\mathbf{a}_j \in V$, $V \subset \mathbb{R}^m$ p-dimensional subspace, $p < \min\{m,n\}$ *unknown*
(≻ possibility of dimensional reduction)

Task (PCA): determine (minimal) p and (orthonormal basis of) V

Perspective of linear algebra:

$$\text{Conjecture} \;\Leftrightarrow\; \operatorname{rank}(\mathbf{A}) = p \text{ for } \mathbf{A} := (\mathbf{a}_1,\ldots,\mathbf{a}_n) \in \mathbb{R}^{m,n},\quad \operatorname{Im}(\mathbf{A}) = V$$

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

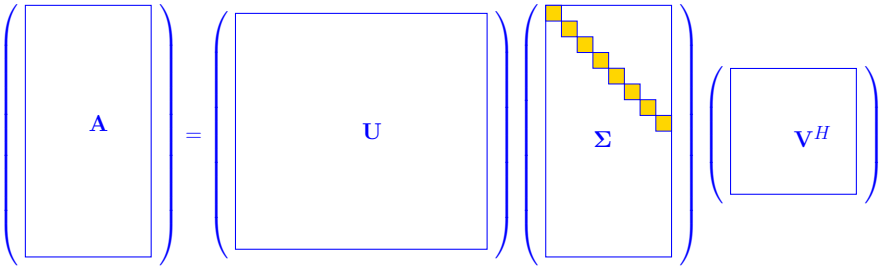
Application: ◣Chemometrics (multivariate calibration methods for the analysis of chemical mixtures)

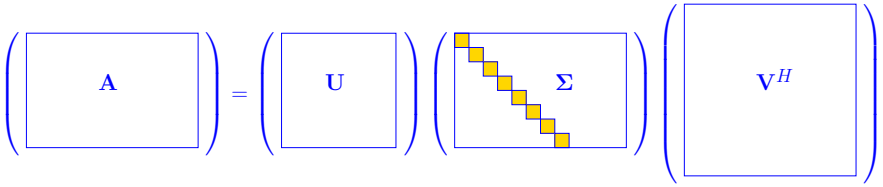
△

Theorem 5.4.1. *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 $\mathbf{\Sigma} = \operatorname{diag}(\sigma_1,\ldots,\sigma_p) \in \mathbb{R}^{m,n}$, $p := \min\{m,n\}$, $\sigma_1 \geq \sigma_2 \geq \sigma_p \geq 0$ such that*

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

$(*)$: $\mathbf{\Sigma}$ (generalized) diagonal matrix :⇔ $(\mathbf{\Sigma})_{i,j} = 0$, if $i \neq j$, $1 \leq i \leq m$, $1 \leq j \leq n$.





Proof. (of Thm. 5.4.1, by induction)

[47, Thm. 4.2.3]: Continuous functions attain extremal values on compact sets (here the unit ball $\{\mathbf{x} \in \mathbb{R}^n\colon \|\mathbf{x}\|_2 \leq 1\}$)

$$\blacktriangleright\quad \exists \mathbf{x} \in \mathbb{K}^n, \mathbf{y} \in \mathbb{K}^m\,,\quad \|\mathbf{x}\| = \|\mathbf{y}\|_2 = 1:\quad \mathbf{Ax} = \sigma\mathbf{y}\,,\quad \sigma = \|\mathbf{A}\|_2\,,$$

where we used the definition of the matrix 2-norm, see Def. 2.4.2. By Gram-Schmidt orthogonalization: $\exists \widetilde{\mathbf{V}} \in \mathbb{K}^{n,n-1}$, $\widetilde{\mathbf{U}} \in \mathbb{K}^{m,m-1}$ such that

$$\mathbf{V} = (\mathbf{x}\ \widetilde{\mathbf{V}}) \in \mathbb{K}^{n,n}\quad,\quad \mathbf{U} = (\mathbf{y}\ \widetilde{\mathbf{U}}) \in \mathbb{K}^{m,m} \text{ are unitary.}$$

$$\blacktriangleright\quad \mathbf{U}^H\mathbf{AV} = (\mathbf{y}\ \widetilde{\mathbf{U}})^H\mathbf{A}(\mathbf{x}\ \widetilde{\mathbf{V}}) = \left(\begin{array}{c|c} \mathbf{y}^H\mathbf{Ax} & \mathbf{y}^H\mathbf{A}\widetilde{\mathbf{V}} \\ \hline \widetilde{\mathbf{U}}^H\mathbf{Ax} & \widetilde{\mathbf{U}}^H\mathbf{A}\widetilde{\mathbf{V}} \end{array}\right) = \left(\begin{array}{c|c} \sigma & \mathbf{w}^H \\ \hline \mathbf{0} & \mathbf{B} \end{array}\right) =: \mathbf{A}_1\,.$$

$$\left\| \mathbf{A}_1 \begin{pmatrix} \sigma \\ \mathbf{w} \end{pmatrix} \right\|_2^2 = \left\| \begin{pmatrix} \sigma^2 + \mathbf{w}^H \mathbf{w} \\ \mathbf{B}\mathbf{w} \end{pmatrix} \right\|_2^2 = (\sigma^2 + \mathbf{w}^H \mathbf{w})^2 + \|\mathbf{B}\mathbf{w}\|_2^2 \geq (\sigma^2 + \mathbf{w}^H \mathbf{w})^2 ,$$

$$\|\mathbf{A}_1\|_2^2 = \sup_{0 \neq \mathbf{x} \in \mathbb{K}^n} \frac{\|\mathbf{A}_1 \mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} \geq \frac{\left\|\mathbf{A}_1 \begin{pmatrix} \sigma \\ \mathbf{w} \end{pmatrix}\right\|_2^2}{\left\| \begin{pmatrix} \sigma \\ \mathbf{w} \end{pmatrix} \right\|_2^2} \geq \frac{(\sigma^2 + \mathbf{w}^H \mathbf{w})^2}{\sigma^2 + \mathbf{w}^H \mathbf{w}} = \sigma^2 + \mathbf{w}^H \mathbf{w} . \tag{5.4.1}$$

$$\sigma^2 = \|\mathbf{A}\|_2^2 = \left\| \mathbf{U}^H \mathbf{A} \mathbf{V} \right\|_2^2 = \|\mathbf{A}_1\|_2^2 \overset{(5.4.1)}{\Longrightarrow} \|\mathbf{A}_1\|_2^2 = \|\mathbf{A}_1\|_2^2 + \|\mathbf{w}\|_2^2 \quad \Rightarrow \quad \mathbf{w} = 0 .$$

$$\blacktriangleright \quad \mathbf{A}_1 = \left(\begin{array}{c|c} \sigma & 0 \\ \hline 0 & \mathbf{B} \end{array} \right) .$$

Then apply induction argument to $\mathbf{B}$ □.

Definition 5.4.2 (Singular value decomposition (SVD))**.**
The decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ of Thm. 5.4.1 is called singular value decomposition (SVD) of $\mathbf{A}$. The diagonal entries σ_i of $\mathbf{\Sigma}$ are the singular values of $\mathbf{A}$.

Lemma 5.4.3. *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.*

Proof. $\mathbf{A}\mathbf{A}^H$ and $\mathbf{A}^H\mathbf{A}$ are similar (→ Lemma 5.1.4) to diagonal matrices with non-zero diagonal entries σ_i^2 ($\sigma_i \neq 0$), e.g.,

$$\mathbf{A}\mathbf{A}^H = \mathbf{U}\mathbf{\Sigma}\mathbf{H}^H\mathbf{V}\mathbf{\Sigma}^H\mathbf{U}^H = \mathbf{U} \underbrace{\mathbf{\Sigma}\mathbf{\Sigma}^H}_{\text{diagonal matrix}} \mathbf{U}^H .$$

□

Remark 5.4.2 (SVD and additive rank-1 decomposition).

Recall from linear algebra: rank-1 matrices are tensor products of vectors

$$\mathbf{A} \in \mathbb{K}^{m,n} \quad \text{and} \quad \operatorname{rank}(\mathbf{A}) = 1 \quad \Leftrightarrow \quad \exists \mathbf{u} \in \mathbb{K}^m, \mathbf{v} \in \mathbb{K}^n : \quad \mathbf{A} = \mathbf{u}\mathbf{v}^H , \tag{5.4.2}$$

because $\operatorname{rank}(\mathbf{A}) = 1$ means that $\mathbf{a}\mathbf{x} = \mu(\mathbf{x})\mathbf{u}$ for some $\mathbf{u} \in \mathbb{K}^m$ and linear form $\mathbf{x} \mapsto \mu(\mathbf{x})$. By the Riesz representation theorem the latter can be written as $\mu(\mathbf{x}) = \mathbf{v}^H\mathbf{x}$.

$\blacktriangleright$ Singular value decomposition provides additive decomposition into rank-1 matrices:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H = \sum_{j=1}^{p} \sigma_j (\mathbf{U})_{:,j} (\mathbf{V})_{:,j}^H . \tag{5.4.3}$$

Remark 5.4.3 (Uniqueness of SVD).
SVD of Def. 5.4.2 is not (necessarily) unique, but the singular values are.

Assume that $\mathbf{A}$ has two singular value decompositions

$$\mathbf{A} = \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^H = \mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2^H \quad \Rightarrow \quad \mathbf{U}_1 \underbrace{\mathbf{\Sigma}_1\mathbf{\Sigma}_1^H}_{=\operatorname{diag}(s_1^1,\ldots,s_m^1)} \mathbf{U}_1^H = \mathbf{A}\mathbf{A}^H = \mathbf{U}_2 \underbrace{\mathbf{\Sigma}_2\mathbf{\Sigma}_2^H}_{=\operatorname{diag}(s_1^2,\ldots,s_m^2)} \mathbf{U}_2^H .$$

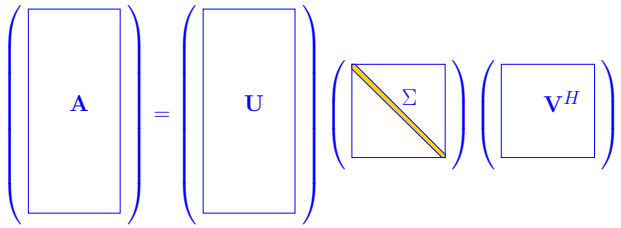
Two similar diagonal matrices are equal ! □

△

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

```
s = svd(A)          : computes singular values of matrix A
[U,S,V] = svd(A)    : computes singular value decomposition according to Thm. 5.4.1
[U,S,V] = svd(A,0)  : "economical" singular value decomposition for m > n: : U ∈
                      K^{m,n}, Σ ∈ R^{n,n}, V ∈ K^{n,n}
s = svds(A,k)       : k largest singular values (important for sparse A → Def. 2.5.1)
[U,S,V] = svds(A,k): partial singular value decomposition: U ∈ K^{m,k}, V ∈ K^{n,k},
                      Σ ∈ R^{k,k} diagonal with k largest singular values of A.
```

Explanation: "economical" singular value decomposition:



(MATLAB) algorithm for computing SVD is (numerically) stable → Def. **??**

Complexity:

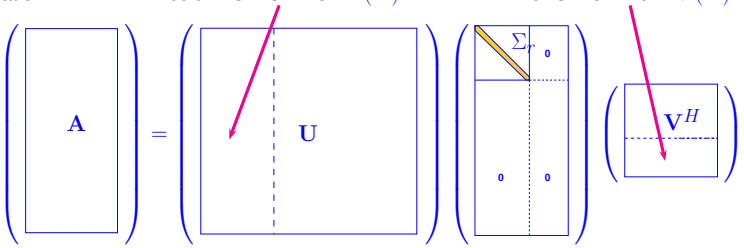
$2mn^2 + 2n^3 + O(n^2) + O(mn)$ for `s = svd(A)`,
$4m^2n + 22n^3 + O(mn) + O(n^2)$ for `[U,S,V] = svd(A)`,
$O(mn^2) + O(n^3)$ for `[U,S,V]=svd(A,0)`, $m \gg n$.

• Application of SVD: computation of rank (→ Def. 2.0.2), kernel and range of a matrix

Lemma 5.4.4 (SVD and rank of a matrix)**.**
If the singular values of $\mathbf{A}$ *satisfy* $\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots \sigma_p = 0$*, then*

- $\operatorname{rank}(\mathbf{A}) = r$,
- $\operatorname{Ker}(\mathbf{A}) = \operatorname{Span}\{(\mathbf{V})_{:,r+1}, \ldots, (\mathbf{V})_{:,n}\}$,
- $\operatorname{Im}(\mathbf{A}) = \operatorname{Span}\{(\mathbf{U})_{:,1}, \ldots, (\mathbf{U})_{:,r}\}$.

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



(5.4.4)

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

Lemma 5.4.4 ➤ 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 5.4.4: 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);
9  figure; plot3(A(1,:),A(2,:),A(3,:),'r*'); grid on; hold on;
10 M = U(:,1:2)*[reshape(X,1,prod(size(X)));reshape(Y,1,prod(size(Y)))];
11 mesh(reshape(M(1,:),n,m),reshape(M(2,:),n,m),reshape(M(3,:),n,m));
12 colormap(cool); view(35,10);
13
14 print -depsc2 '../PICTURES/svdpca.eps';
```


singular values:

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

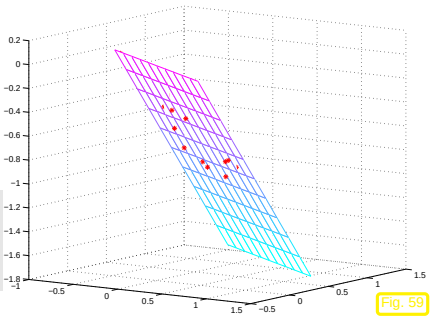


Fig. 59

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

Example 5.4.5 (Principal component analysis for data analysis).

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

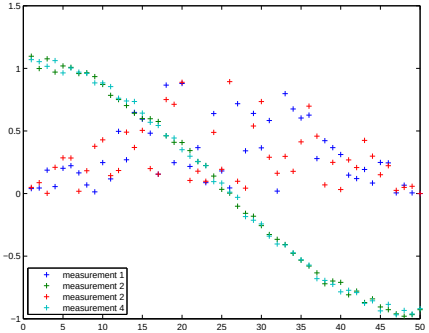
Columns of $\mathbf{A}$ → series of measurements at different times/locations etc.
Rows of $\mathbf{A}$ → measured values corresponding to one time/location etc.

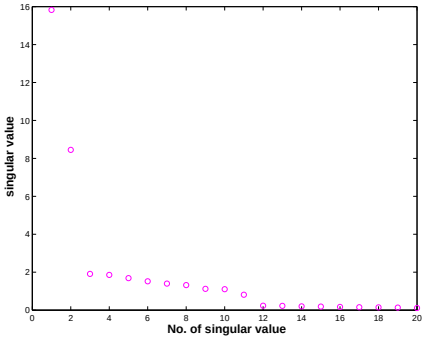
Goal: detect linear correlations

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

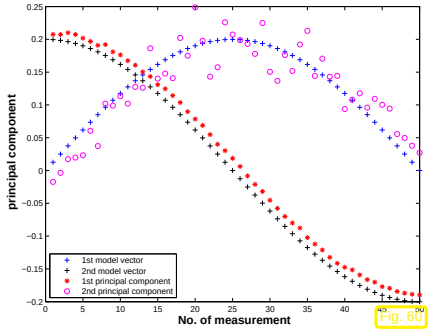


Fig. 60

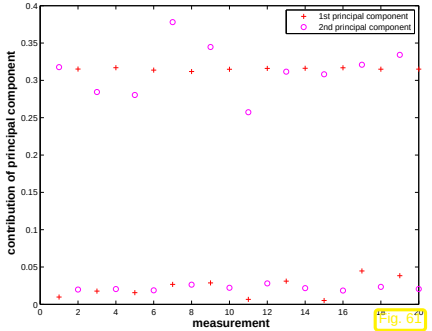


Fig. 61 ◇

• Application of SVD: extrema of quadratic forms on the unit sphere

A minimization problem on the Euclidean unit sphere $\{\mathbf{x} \in \mathbb{K}^n\colon \|\mathbf{x}\|_2 = 1\}$:

given $\mathbf{A} \in \mathbb{K}^{m,n}$, $m > n$, find $\mathbf{x} \in \mathbb{K}^n$, $\|\mathbf{x}\|_2 = 1$, $\|\mathbf{A}\mathbf{x}\|_2 \to \min$. (5.4.5)

Use that multiplication with unitary matrices preserves the 2-norm (→ Thm. **??**) and the singular value decomposition $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^H$ (→ Def. 5.4.2):

$$\min_{\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2^2 = \min_{\|\mathbf{x}\|_2=1} \left\|\mathbf{U}\Sigma\mathbf{V}^H\mathbf{x}\right\|_2^2 = \min_{\|\mathbf{V}^H\mathbf{x}\|_2=1} \left\|\mathbf{U}\Sigma(\mathbf{V}^H\mathbf{x})\right\|_2^2$$
$$= \min_{\|\mathbf{y}\|_2=1} \|\Sigma\mathbf{y}\|_2^2 = \min_{\|\mathbf{y}\|_2=1} (\sigma_1^2 y_1^2 + \cdots + \sigma_n^2 y_n^2) \geq \sigma_n^2 .$$

The minimum σ_n^2 is attained for $\mathbf{y} = \mathbf{e}_n$ ⇒ minimizer $\mathbf{x} = \mathbf{V}\mathbf{e}_n = (\mathbf{V})_{:,n}$.

By similar arguments:

$$\sigma_1 = \max_{\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2 \quad , \quad (\mathbf{V})_{:,1} = \operatorname{argmax}_{\|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2 . \tag{5.4.6}$$

Recall: 2-norm of the matrix $\mathbf{A}$ (→ Def. 2.4.2) is defined as the maximum in (5.4.6). Thus we have proved the following theorem:

Lemma 5.4.5 (SVD and Euclidean matrix norm)**.**

- $\forall \mathbf{A} \in \mathbb{K}^{m,n}\colon \|\mathbf{A}\|_2 = \sigma_1(\mathbf{A})$,
- $\forall \mathbf{A} \in \mathbb{K}^{n,n}$ *regular*: $\operatorname{cond}_2(\mathbf{A}) = \sigma_1/\sigma_n$.

Remark: MATLAB functions `norm(A)` and `cond(A)` rely on svd(A)

• Application of SVD: *best low rank approximation*

Definition 5.4.6 (Frobenius norm).
The Frobenius norm of $\mathbf{A} \in \mathbb{K}^{m,n}$ is defined as

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

Obvious: $\|\mathbf{A}\|_F$ invariant under unitary transformations of $\mathbf{A}$

Frobenius norm and SVD: $\boxed{\|\mathbf{A}\|_F^2 = \sum_{j=1}^{p} \sigma_j^2}$ (5.4.7)

✎ notation: $\mathcal{R}_k(m,n) := \{\mathbf{A} \in \mathbb{K}^{m,n}: \operatorname{rank}(\mathbf{A}) \le k\}$, $m,n,k \in \mathbb{N}$

Theorem 5.4.7 (best low rank approximation).
Let $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^H$ be the SVD of $\mathbf{A} \in \mathbb{K}^{m,n}$ (→ Thm. 5.4.1). For $1 \le k \le \operatorname{rank}(\mathbf{A})$ set $\mathbf{U}_k := [\mathbf{u}_{:,1}, \ldots, \mathbf{u}_{:,k}] \in \mathbb{K}^{m,k}$, $\mathbf{V}_k := [\mathbf{v}_{:,1}, \ldots, \mathbf{v}_{:,k}] \in \mathbb{K}^{n,k}$, $\mathbf{\Sigma}_k := \operatorname{diag}(\sigma_1, \ldots, \sigma_k) \in \mathbb{K}^{k,k}$. Then, for $\|\cdot\| = \|\cdot\|_F$ and $\|\cdot\| = \|\cdot\|_2$, holds true

$$\left\|\mathbf{A} - \mathbf{U}_k\Sigma_k\mathbf{V}_k^H\right\| \le \|\mathbf{A} - \mathbf{F}\| \quad \forall \mathbf{F} \in \mathcal{R}_k(m,n) .$$

Proof. Write $\mathbf{A}_k = \mathbf{U}_k\Sigma_k\mathbf{V}_k^H$. Obviously, with $r = \operatorname{rank}\mathbf{A}$,

$$\operatorname{rank}\mathbf{A}_k = k \quad \text{and} \quad \|\mathbf{A} - \mathbf{A}_k\| = \|\mathbf{\Sigma} - \mathbf{\Sigma}_k\| = \begin{cases} \sigma_{k+1} & \text{, for } \|\cdot\| = \|\cdot\|_2 , \\ \sqrt{\sigma_{k+1}^2 + \cdots + \sigma_r^2} & \text{, for } \|\cdot\| = \|\cdot\|_F . \end{cases}$$

❶ Pick $\mathbf{B} \in \mathbb{K}^{n,n}$, $\operatorname{rank}\mathbf{B} = k$.

$$\dim \operatorname{Ker}(\mathbf{B}) = n - k \;\Rightarrow\; \operatorname{Ker}(\mathbf{B}) \cap \operatorname{Span}\{\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}\} \neq \{0\} ,$$

where $\mathbf{v}_i$, $i = 1, \ldots, n$ are the columns of $\mathbf{V}$. For $\mathbf{x} \in \operatorname{Ker}(\mathbf{B}) \cap \operatorname{Span}\{\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}\}$, $\|\mathbf{x}\|_2 = 1$

$$\|\mathbf{A} - \mathbf{B}\|_2^2 \ge \|(\mathbf{A} - \mathbf{B})\mathbf{x}\|_2^2 = \|\mathbf{A}\mathbf{x}\|_2^2 = \left\|\sum_{j=1}^{k+1} \sigma_j(\mathbf{v}_j^H\mathbf{x})\mathbf{u}_j\right\|_2^2 = \sum_{j=1}^{k+1} \sigma_j^2(\mathbf{v}_j^H\mathbf{x})^2 \ge \sigma_{j+1}^2 ,$$

because $\sum_{j=1}^{k+1} (\mathbf{v}_j^H\mathbf{x})^2 = 1$.

❷ Find ONB $\{\mathbf{z}_1, \ldots, \mathbf{z}_{n-k}\}$ of $\operatorname{Ker}(\mathbf{B})$ and assemble it into a matrix $\mathbf{Z} = [\mathbf{z}_1 \ldots \mathbf{z}_{n-k}] \in \mathbb{K}^{n,n-k}$

$$\|\mathbf{A} - \mathbf{B}\|_F^2 \ge \|(\mathbf{A} - \mathbf{B})\mathbf{Z}\|_F^2 = \|\mathbf{A}\mathbf{Z}\|_F^2 = \sum_{i=1}^{n-k} \|\mathbf{A}\mathbf{z}_i\|_2^2 = \sum_{i=1}^{n-k}\sum_{j=1}^{r} \sigma_j^2(\mathbf{v}_j^H\mathbf{z}_i)^2 \qquad \square$$

Note: information content of a rank-k matrix $\mathbf{M} \in \mathbb{K}^{m,n}$ is equivalent to $k(m+n)$ numbers!

Approximation by low-rank matrices ↔ matrix compression

5.5 Essential Skills Learned in Chapter 5

You should know:

- complexity of the direct eigensolver eig of Matlab
- what are Krylov methods, when and how to use them
- what is the singular value decomposition and how to use it
- applications of the svd: principal component analysis, extrema of quadratic forms on the unit sphere, best low rank approximation