Numerical Methods for Computational Science and Engineering

Fall Semester 2017 (HS17)

Prof. Rima Alaifari, SAM, ETH Zurich

3.4.4. SVD-based Optimization & Approximation

3.4.4.1 Norm-constrained Extrema

Consider the following minimization problem:

Given $A \in \mathbb{K}^{m,n}$ $\quad m \geq n$

find $x \in \mathbb{K}^n$, $\|x\|_2 = 1$ s.t. $\|Ax\|_2 \to \min$

Use SVD: $A = U \Sigma V^H$

$$\min_{\|x\|_2=1} \|Ax\|_2^2 = \min_{\|x\|_2=1} \|U\Sigma V^H x\|_2^2 = \min_{\|x\|_2=1} \|\Sigma(V^H x)\|_2^2$$

$$= \min_{\|y\|_2=1} \|\Sigma y\|_2^2 = \min_{\|y\|_2=1} (\sigma_1^2 y_1^2 + \sigma_2^2 y_2^2 + \dots + \sigma_n^2 y_n^2)$$

$\uparrow$ $y = V^H x$

$$\geq \sigma_n^2$$

$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$

minimal value is attained with $y = e_n$

$$\Rightarrow \quad V^H x = e_n$$

$$x = V e_n = (V)_{:,n}$$

Example (computational geometry): fitting a hyperplane

cf. Example 3.4.35

Given m points y_j in $\mathbb{R}^d$, find a hyperplane $\mathcal{H}$ s.t.

$$\sum_{i=1}^{m} \operatorname{dist}(\mathcal{H}, y_j)^2 \to \min$$

3.4.4.2 Best low-rank approximation

For a given matrix A, find a low-rank matrix that is the closest to A:

Given a matrix $A \in \mathbb{K}^{m,n}$

Find a matrix $\tilde{A} \in \mathbb{K}^{m,n}$, $\operatorname{rank}(\tilde{A}) \leq k$ s.t.

$$\|A - \tilde{A}\|_{2/F} \longrightarrow \min \quad \text{over rank-}k \text{ matrices}$$

$\|\cdot\|_F$ Frobenius norm of $A \in \mathbb{K}^{m,n}$:

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

can be computed from the singular values:

$$\|A\|_F^2 = \sum_{j=1}^{r} \sigma_j^2 \qquad r = \operatorname{rank}(A) \quad (\#\text{ of nonzero sing. vals})$$

Typically: $k \ll \min\{m,n\}$

Applications: • data compression

• mathematical modelling

Use SVD: $A = U \Sigma V^H$

$$A = \sum_{i=1}^{r} \sigma_i \underbrace{(U)_{:,i} (V)_{:,i}^H}_{\text{outer product}}$$

Take $\tilde{A} = \sum_{i=1}^{k} \sigma_i (U)_{:,i} (V)_{:,i}^H$

This is the best k-rank approximation to A

I.e. for $R_k(m,n) := \{ F \in \mathbb{K}^{m,n} : \operatorname{rank}(F) \leq k \}$

$\tilde{A}$ is closest to A in $R_k(m,n)$!

↑ in 2-norm & in Frobenius norm

[Eckart-Young – Mirsky]

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

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

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

Best low-rank compression useful for compression

(Dense matrix cannot be well approximated by sparse matrix)

But best low-rank approximation: reduce required memory from $m \cdot n$ to $k(m+n-1)$

How good is the approximation:

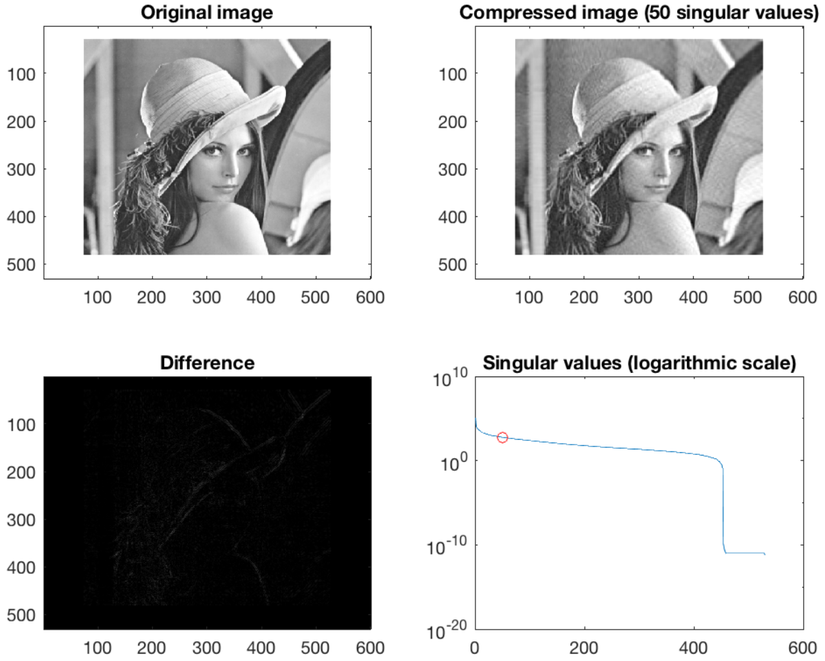
$$A - \tilde{A} = \sum_{i=k+1}^{r} \sigma_i (U)_{:,i} (V)_{:,i}^{H}$$

Singular values of $A - \tilde{A}$: $\sigma_{k+1} \ge \ldots \ge \sigma_r$

largest sing. value of $A - \tilde{A}$: σ_{k+1}

$$\|A - \tilde{A}\|_2 = \sigma_{k+1} \qquad \|A - \tilde{A}\|_F^2 = \sum_{i=k+1}^{r} \sigma_i^2$$

Example: Image Compression



Original image: $530 \times 600 \approx 3 \cdot 10^5$

Compressed image: $50 \cdot \underbrace{1129}_{m+n-1} \approx 5 \cdot 10^4$

Note: In practice, other methods are used for image compression (e.g. wavelet-based)

3.4.4.3 Principle Component Analysis (PCA)

Important tool:
- dimensionality reduction
- trend analysis
- data classification

Example:

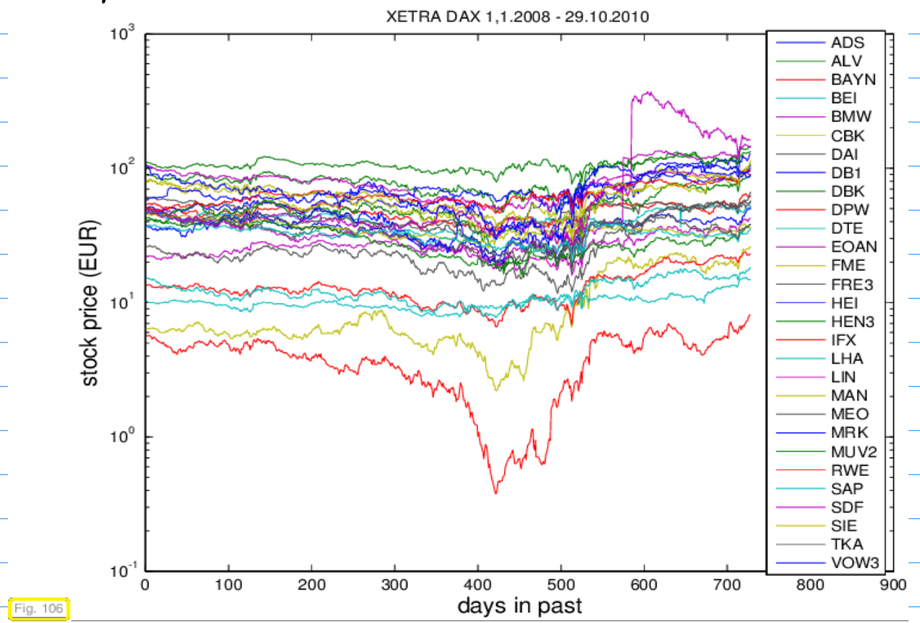


Fig. 106

Trend in stock price development?

Each stock price as vector $a_j \in \mathbb{R}^m$ for n stocks

Trend ≙ Are there few vectors $\tilde{u}_1, \dots, \tilde{u}_p$, $p \ll n$

s.t. $a_j \in \text{span}\{\tilde{u}_1, \dots, \tilde{u}_p\}$ for all $j \in \{1, \dots, n\}$

[and $\tilde{u}_1, \dots, \tilde{u}_p$ orthonormal] ?

This would mean:

$A = [a_1 \; a_2 \; \dots \; a_n] \in \mathbb{R}^{m,n}$ $\quad \text{rank}(A) = p$

and $\quad \mathcal{R}(A) = \text{span}\{\tilde{u}_1, \dots, \tilde{u}_p\}$

Unrealistic scenario

Because of small random fluctuations in stock prices

Instead: Can we find orthonormal vectors

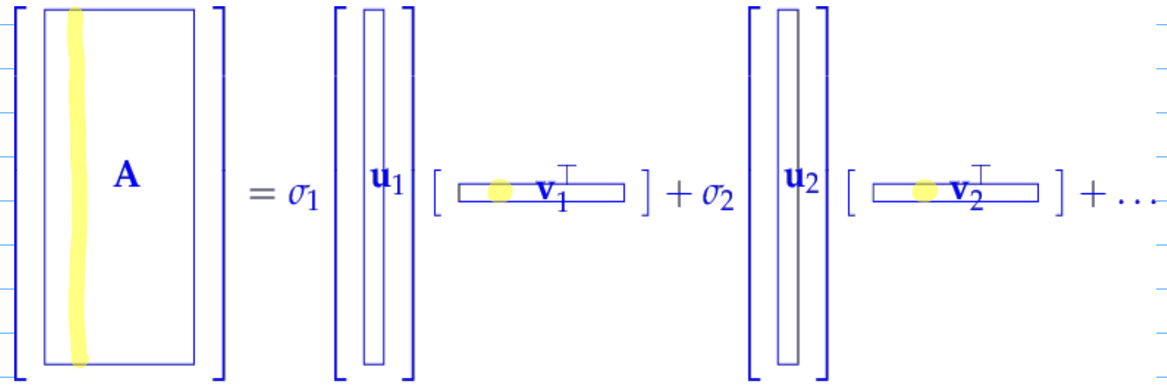
$\tilde{u}_1, \dots, \tilde{u}_p$ s.t.

$\forall j \in \{1, \dots, n\}$: $a_j \in \text{span}\{\tilde{u}_1, \dots, \tilde{u}_p\}$ + "small perturb."

i.e. stock prices approximately follow a trend

How? Exploit SVD of A

$A = U \Sigma V^T$



Each a_j is a linear combination of $u_1, u_2, \dots, u_n$

↑ columns of U

Vectors v_i (and hence matrix V) carry weights of the lin. comb.

$$a_j = (\sigma_1 (v_1)_j)\, u_1 + (\sigma_2 (v_2)_j)\, u_2 + \dots + (\sigma_n (v_n)_j)\, u_n$$

No perturbations (A is exactly low-rank):

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

orthonormal trend vectors $(\mathbf{U})_{:,1}, \dots, (\mathbf{U})_{:,p}$.

(unrealistic)

With perturbations (A is approximately low-rank):

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

orthonormal trend vectors $(\mathbf{U})_{:,1}, \dots, (\mathbf{U})_{:,p}$.

$\sigma_p \gg \sigma_{p+1}$ "pronounced gap"

if it exists: "$\mathcal{R}(A)$ is almost p-dimensional"

Then: $A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_p u_p v_p^T +$

$$+ \underbrace{\sigma_{p+1} u_{p+1} v_{p+1}^T + \dots + \sigma_n u_n v_n^T}_{\text{small}}$$

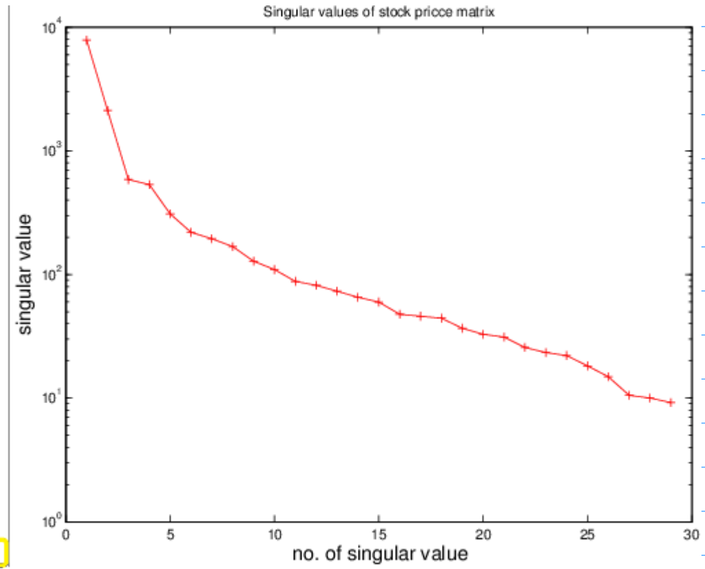


Fig. 121

Example:

Sing. values of the stock price matrix $A = [a_1 \dots a_n]$ decay $\approx$ exponentially

Common criterion:

$$p = \min\left\{ q : \sum_{j=1}^{q} \sigma_j^2 \geq (1-\tau) \sum_{j=1}^{n} \sigma_j^2 \right\} \quad \tau \ll 1$$

i.e. sum of the p first σ_j^2's is almost as large as the sum over all σ_j^2's

Example (3.4.55): Classification of measured data

measured data voltage vs. current of 20 diodes

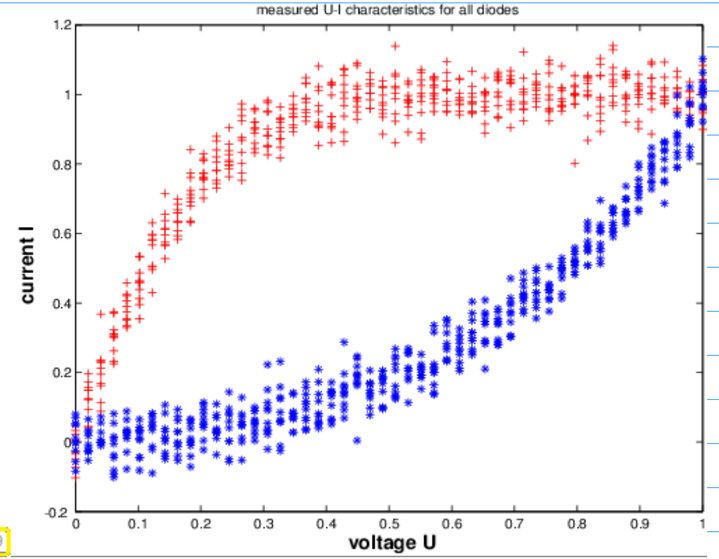


Fig. 109

How many different types of diodes are there?

U - I characteristics of these types?

Write data vectors in matrix & compute SVD

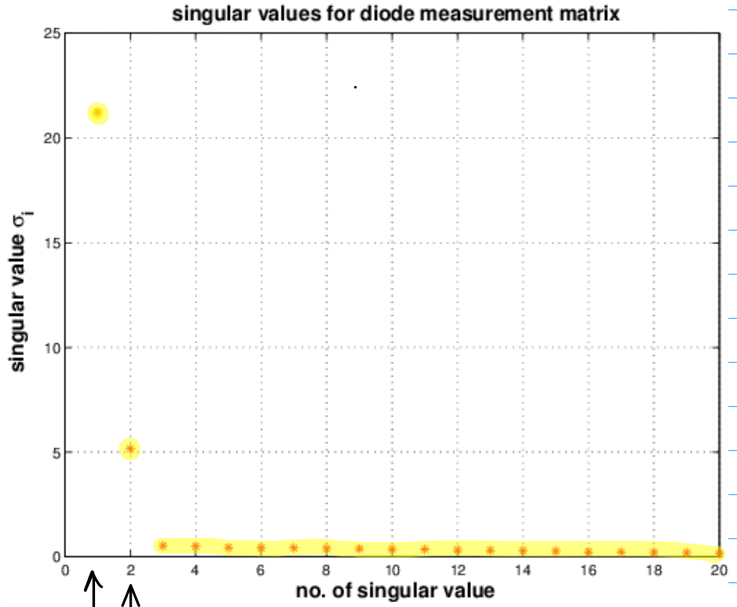


Fig. 113

2 dominant singular values

⇒ 2 principle components, i.e. two types of diodes

3.5. Total Least Squares

So far: $Ax = b$ overdetermined $(m > n)$

$A \in \mathbb{K}^{m,n}$

and b is possibly perturbed

→ find least-squares solution

$\|Ax - b\|_2 \longrightarrow \min$

Now: More generally: also system matrix can be perturbed (e.g. consisting of measurements)

Total least-squares problem:

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

Find $\hat{A} \in \mathbb{R}^{m,n}$, $\hat{b} \in \mathbb{R}^m$ with

$$\| [A \; b] - [\hat{A} \; \hat{b}] \|_F \to \min, \quad \hat{b} \in \mathcal{R}(\hat{A})$$

$\hat{b} \in \mathcal{R}(\hat{A})$ $\quad \operatorname{rank}\left([\hat{A} \; \hat{b}]\right) = n$

$$\Rightarrow \quad [\hat{A} \; \hat{b}] = \underset{\operatorname{rank}(\hat{X})=n}{\operatorname{argmin}} \| [A \; b] - \hat{X} \|_F$$

$\uparrow$

best n-rank approximation to $[A \; b]$

$\rightarrow$ Use SVD of $[A \; b] \in \mathbb{R}^{m,n+1}$:

$$[A \; b] = U \Sigma V^T = \sum_{j=1}^{n+1} \sigma_j (U)_{:,j} (V)_{:,j}^T$$

best n-rank approximation:

$$[\hat{A} \; \hat{b}] = \sum_{j=1}^{n} \sigma_j (U)_{:,j} (V)_{:,j}^T$$

$$\Rightarrow \quad [\hat{A} \; \hat{b}] (V)_{:,n+1} = 0 \qquad [V \text{ orthogonal}]$$

$\uparrow$

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

$$\Rightarrow \quad \hat{A} (V)_{1:n,n+1} + \hat{b} (V)_{n+1,n+1} = 0$$

$$\Rightarrow \quad \frac{1}{(V)_{n+1,n+1}} (V)_{1:n,n+1} = -\hat{A}^{-1} \hat{b}$$

$$x := \hat{A}^{-1} \hat{b} = -\frac{1}{(V)_{n+1,n+1}} (V)_{1:n,n+1}$$

$x = \hat{A}^{-1}\hat{b}$ solution to the total least-sq. problem

3.6 Constrained Least-Squares

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

$C \in \mathbb{R}^{p,n}$, $\operatorname{rank}(C) = p$, $p < n$, $d \in \mathbb{R}^p$

find $x \in \mathbb{R}^n$ s.t. $\|Ax-b\|_2 \to \min$

and $Cx = d$

I.e. $\begin{bmatrix} A \\ C \end{bmatrix} x = \begin{bmatrix} b \\ d \end{bmatrix}$

$Ax = b$ least-sq. sense

$Cx = d$ fulfilled exactly

$\|Ax-b\|_2 \to \min$ with $Cx = d$

constrained optimization problem

Solve via Lagrange multiplier formulation:

$$L(y,m) := \frac{1}{2}\|Ay-b\|_2^2 + m^T(Cy-d), \quad m \in \mathbb{R}^p$$

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

saddle point problem

Intuition: $\sup_{m \in \mathbb{R}^p} L(y,m) = \begin{cases} \infty & \text{if } Cy \neq d \\ \frac{1}{2}\|Ay-b\|_2^2 & \text{if } Cy = d \end{cases}$

⇒ solution x to min-max problem has to satisfy $Cx = d$ exactly

Visualization

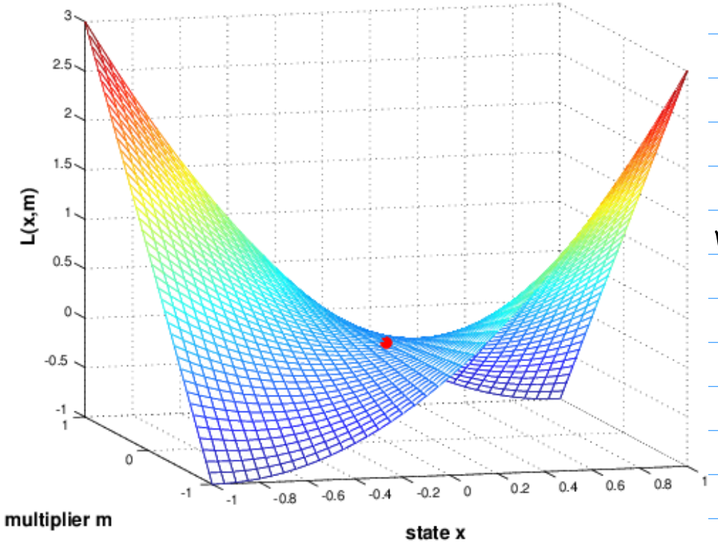


Fig. 126

At saddle point the partial derivatives necessarily vanish

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

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

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

Augmented normal equations
(matrix saddle point problem)

3.6.2. Solution via SVD

$\operatorname{rank}(C) = p$, $C \in \mathbb{R}^{p,n}$ $\quad p < n$ "fat matrix"

$$C = U\begin{bmatrix} \Sigma_p & 0 \end{bmatrix}\begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \qquad U, \Sigma_p \in \mathbb{R}^{p,p}$$

$$V_1 \in \mathbb{R}^{n,p}\,,\ V_2 \in \mathbb{R}^{n,n-p}$$

$Cx = d$ underdetermined

x_0 particular solution to $Cx = d$

$x = x_0 + \mathcal{N}(C)$ is also a solution

$$x_0 = V_1 \Sigma_p^{-1} U^T d$$

Recall $\mathcal{N}(C) = \mathcal{R}(V_2)$ [$\mathcal{R}$ range]

V_2 forms ONB of $\mathcal{N}(C)$

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

Now solve

$$\| Ax - b \|_2 \to \min$$

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

$$\| AV_2 y - (b - Ax_0) \|_2 \to \min$$

unconstrained least-sq. problem

with $AV_2 \in \mathbb{R}^{m, n-p}$, $b - Ax_0 \in \mathbb{R}^m$

4. Filtering Algorithms

signal processing: time-discrete signals as vectors/sequences through sampling of time-continuous signals

time-continuous signal $X(t)$ $\quad t \in [0, T]$

time-discrete signal $x_j = X(j \cdot \Delta t)$

$$x = [x_0, x_1, \dots, x_{n-1}]^T \in \mathbb{R}^n$$

$n \cdot \underbrace{\Delta t}_{\text{time between 2 samples}} \leq T$

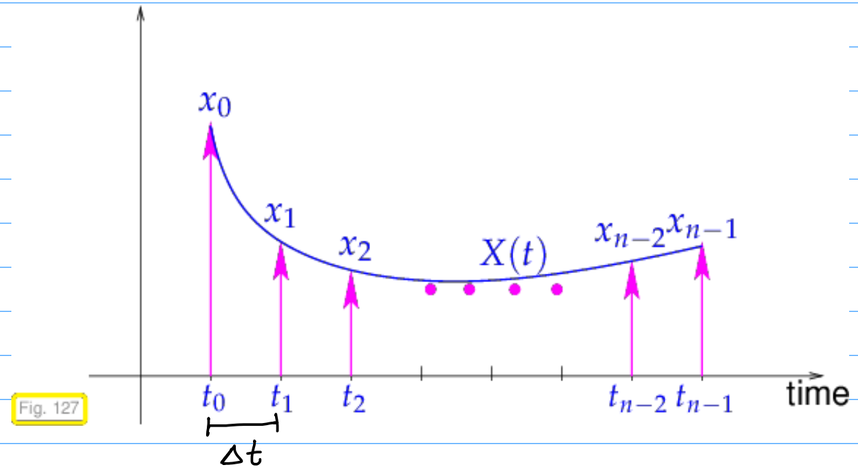


Fig. 127

More generally: $l^\infty(\mathbb{Z})$ as vector space of bounded signals

4.1 Discrete Convolutions

Filters / channels e.g. in wireless communication

We will consider finite linear time-invariant causal filters (general

model for many digital communication channels)

Mathematically: channel/filter is a mapping

$$F: \ell^\infty(\mathbb{Z}) \to \ell^\infty(\mathbb{Z})$$

$$F((x_j)_{j\in\mathbb{Z}}) = (y_j)_{j\in\mathbb{Z}}$$

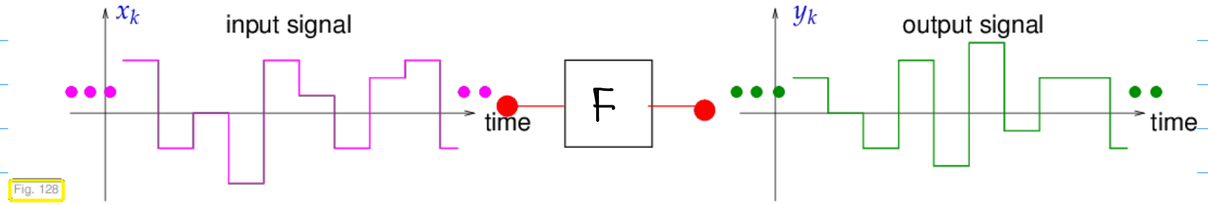


Fig. 128

Properties of F:

- finite: Every finite-length signal $(x_j)_{j\in\mathbb{Z}}$ produces finite-length output $F((x_j)_{j\in\mathbb{Z}})$

- time-invariant: F commutes with shift operator

  time-shifted input + apply filter
  $\hat{=}$ apply filter + time-shifted output

  time-shift operator $S_m: \ell^\infty(\mathbb{Z}) \to \ell^\infty(\mathbb{Z})$

  $$S_m((x_j)_{j\in\mathbb{Z}}) = (x_{j-m})_{j\in\mathbb{Z}}$$

  $m \in \mathbb{Z}$

  time-invariant:

  $$F(S_m((x_j)_{j\in\mathbb{Z}})) = S_m(F((x_j)_{j\in\mathbb{Z}}))$$

- linear: for all $(x_j)_{j\in\mathbb{Z}}, (y_j)_{j\in\mathbb{Z}} \in \ell^\infty(\mathbb{Z})$

  $\alpha, \beta \in \mathbb{R}$

  $$F(\alpha(x_j)_{j\in\mathbb{Z}} + \beta(y_j)_{j\in\mathbb{Z}}) = \alpha F((x_j)_{j\in\mathbb{Z}}) + \beta F((y_j)_{j\in\mathbb{Z}})$$

[linear comb. of input has output equal to lin. comb. of individual outputs]

- causal: output only depends on past & present inputs, not on the future

if $x_j = 0 \quad \forall\, j \le M \Rightarrow F((x_j)_{j\in\mathbb{Z}})_k = 0 \quad \forall\, k \le M$

Impulse response:

Matrix $\hat{=}$ describing the action of a linear mapping $\mathbb{R}^n \to \mathbb{R}^m$ through its action on unit vectors

Now: describe filters through their action on "impulses"

Definition 4.1.3. Impulse response

The impulse response of a channel/filter is the output for a single unit impulse at $t = 0$ as input, that is, the input signal is $x_j = \delta_{j,0} := \begin{cases} 1 & \text{, if } j = 0 \\ 0 & \text{else} \end{cases}$ (Kronecker symbol).

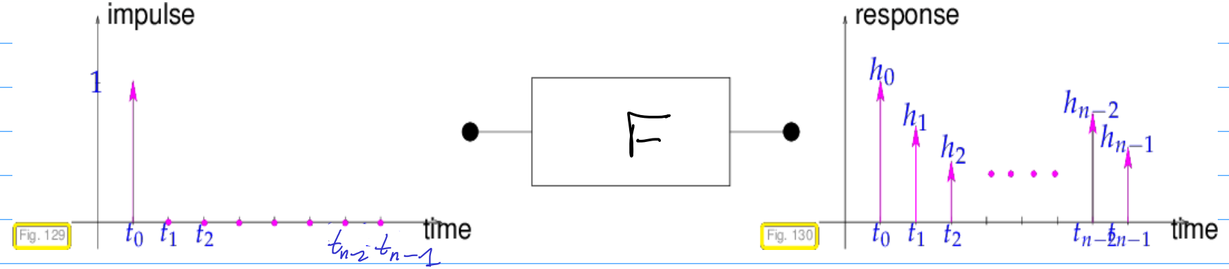


Fig. 129 Fig. 130

$$h_k = F((\delta_{j,0})_{j\in\mathbb{Z}})_k$$

FIR filters: finite impulse response (only finitely many nonzero h_k's)

Finite time-invariant linear causal filters: LTI-FIR

Impulse response: $(\dots, 0, h_0, h_1, \dots, h_{n-1}, 0, \dots)$

$n \in \mathbb{N}$ "n-th order filter"

$$F((\delta_{j,0})_{j\in\mathbb{Z}}) = (\dots, 0, h_0, h_1, \dots, h_{n-1}, 0, \dots)$$

$$F((\delta_{j,k})_{j\in\mathbb{Z}}) = (\dots, 0, h_0, h_1, \dots, h_{n-1}, 0, \dots)$$

↑ shifted unit pulse ↑ at time $t = k \cdot \Delta t$

Any finite signal $(x_j)_{j\in\mathbb{Z}} = (\dots, 0, 0, x_0, x_1, \dots, x_{m-1}, 0, \dots)$

linear combination of shifted pulses

$$(x_j)_{j\in\mathbb{Z}} = \sum_{k=0}^{m-1} x_k \cdot (\delta_{j,k})_{j\in\mathbb{Z}}$$

$$= \sum_{k=0}^{m-1} x_k \cdot S_k((\delta_{j,0})_{j\in\mathbb{Z}})$$

$$F((x_j)_{j\in\mathbb{Z}}) = F\left(\sum_{k=0}^{m-1} x_k \cdot S_k((\delta_{j,0})_{j\in\mathbb{Z}})\right)$$

$$= \sum_{k=0}^{m-1} x_k \, S_k\left(\underbrace{F((\delta_{j,0})_{j\in\mathbb{Z}})}\right)$$

↑ linearity + time-invariance $= (\dots, 0, h_0, h_1, \dots, h_{n-1}, 0, \dots)$

$$(y_j)_{j\in\mathbb{Z}} = F((x_j)_{j\in\mathbb{Z}})$$

$$\Rightarrow \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{m+n-2} \end{bmatrix} = x_0 \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{n-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + x_1 \begin{bmatrix} 0 \\ h_0 \\ h_1 \\ \vdots \\ h_{n-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \dots + x_{m-1} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ h_0 \\ \vdots \\ h_{n-1} \end{bmatrix}$$

$$\Rightarrow F((x_j)_{j\in\mathbb{Z}})_k = y_k = \sum_{j=0}^{m-1} x_j h_{k-j}$$

$k = 0, \dots, m+n-2$, $h_j := 0$ for $j < 0$ & $j \geq n$

Maximal duration of output: $(m+u-2)\cdot\Delta t$

↑ length of signal ↑ length of filter

Example:

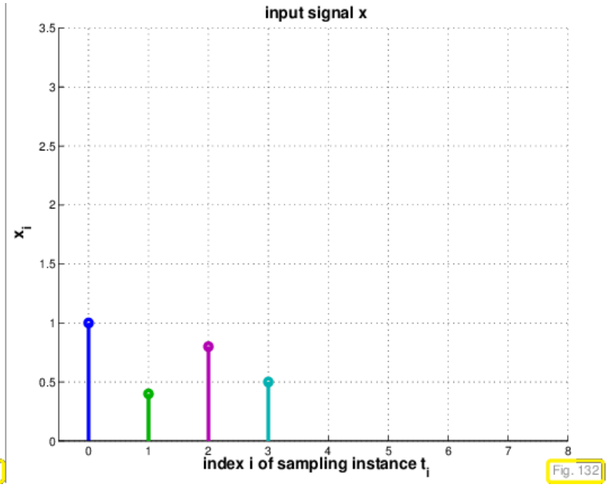


Fig. 131

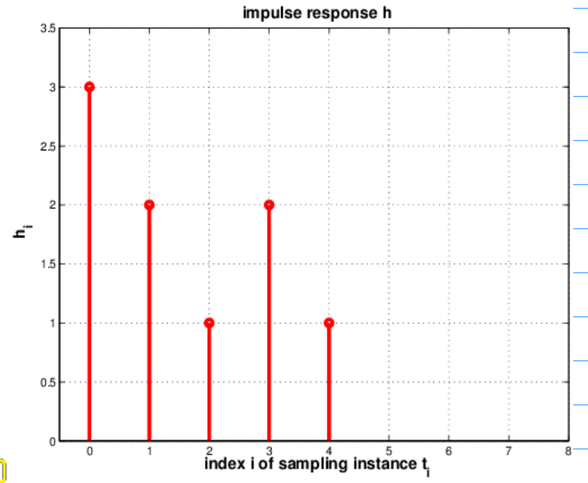


Fig. 132

signal: $(\ldots, 0, x_0, \ldots, x_3, 0, \ldots)$ $m=4$

filter: $(\ldots, 0, h_0, \ldots, h_4, 0, \ldots)$ $n=5$

Output:

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_7 \end{bmatrix} = x_0 \begin{bmatrix} h_0 \\ \vdots \\ h_4 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + x_1 \begin{bmatrix} 0 \\ h_1 \\ \vdots \\ h_4 \\ 0 \\ 0 \end{bmatrix} + \ldots + x_3 \begin{bmatrix} 0 \\ 0 \\ h_0 \\ \vdots \\ h_4 \end{bmatrix}$$

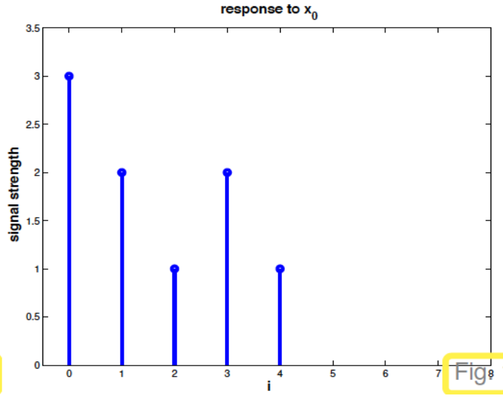


Fig. 134

+

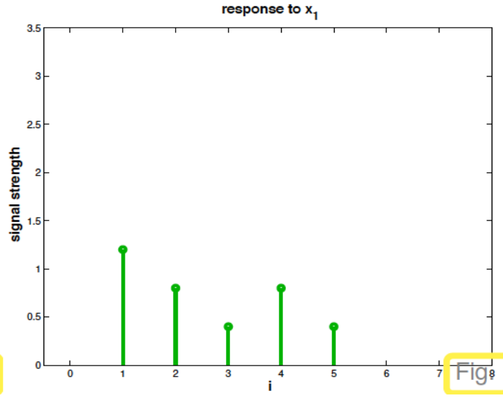


Fig. 135

+

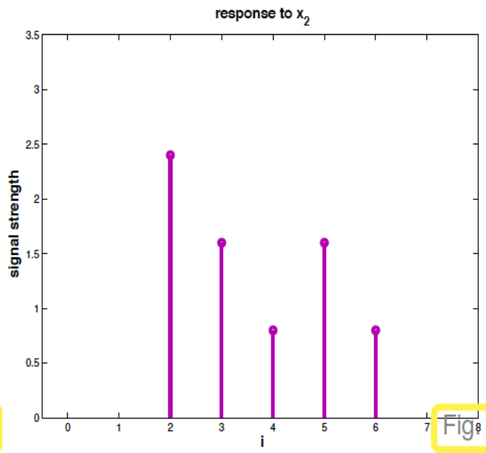
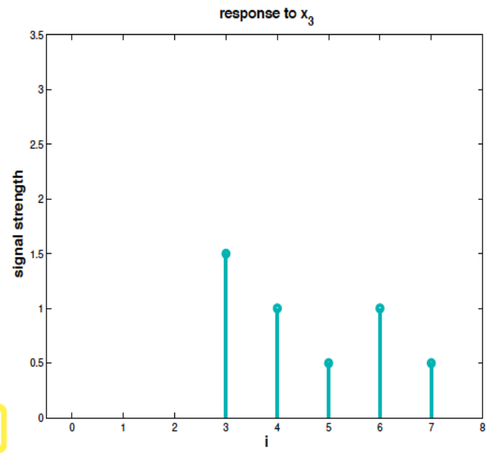


Fig. 136

+



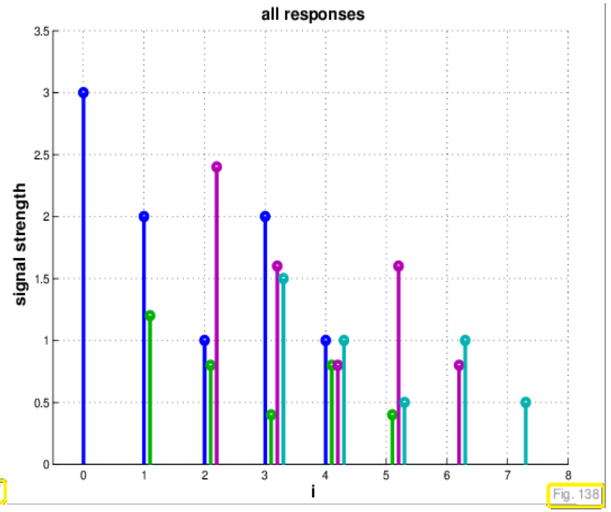


Fig. 137

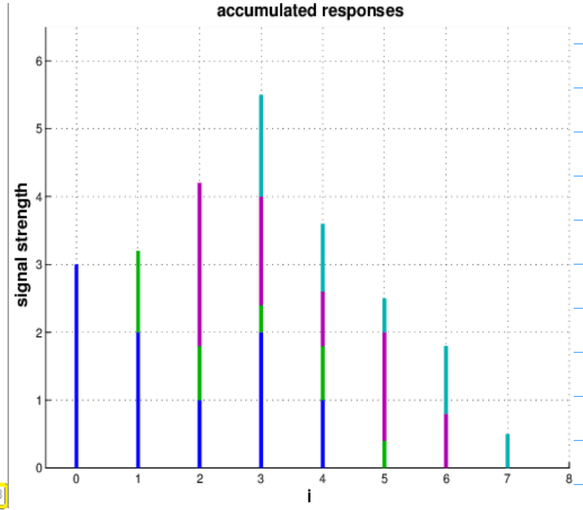


Fig. 138

So for finite-length signals:

$(\dots, 0, x_0, \dots, x_{m-1}, 0, \dots)$ can be represented by a vector

$\underline{x} = [x_0, \dots, x_{m-1}]^T \in \mathbb{R}^m$ and filter is a linear mapping $F: \mathbb{R}^m \to \mathbb{R}^{m+n-1}$ (for filter length n, i.e. impulse response of duration $\Delta t \cdot (n-1)$)

$$\begin{bmatrix} y_0 \\ \vdots \\ y_{m+n-2} \end{bmatrix} = \begin{bmatrix} h_0 & 0 & \cdots & 0 \\ \vdots & h_0 & & \vdots \\ h_{n-1} & & \ddots & h_0 \\ 0 & h_{n-1} & & \vdots \\ \vdots & 0 & \ddots & h_{n-1} \\ 0 & \vdots & & \\ & 0 & & \end{bmatrix} \begin{bmatrix} x_0 \\ \vdots \\ x_{m-1} \end{bmatrix}$$

Representation of LT-FIR filter as a matrix

Recall: $F((x_i)_{i\in\mathbb{Z}})_k = \sum_{j\in\mathbb{Z}} x_j h_{k-j}$

$\Rightarrow F((x_i)_{i\in\mathbb{Z}})$ is the discrete convolution of

$x = (\dots, 0, x_0, \dots, x_{m-1}, 0, \dots)$

$h = (\dots, 0, h_0, \dots, h_{n-1}, 0, \dots)$

Definition (Discrete convolution)

For two sequences $f, g \in \ell^\infty(\mathbb{Z})$ their discrete convolution $u := f * g \; (= g * f) \in \ell^\infty(\mathbb{Z})$ is defined by

$$u_k = \sum_{j \in \mathbb{Z}} f_j \, g_{k-j}$$

$$= \sum_{j \in \mathbb{Z}} f_{k-j} \, g_j$$

$$y = F((x_i)_{i \in \mathbb{Z}}) = x * h \quad \in \ell^\infty(\mathbb{Z})$$

Equivalently: discrete convolution for vectors (finite length sequences)

$$\underline{y} := \underline{x} * \underline{h} \qquad y_k := \sum_{j=0}^{m-1} x_j \, h_{k-j}$$

$\underline{y} \in \mathbb{R}^{m+n-1}$

where $h_j = 0$ for $j < 0$ & $j \geq n$