Ill–Posed Inverse Problems in Image Processing

Introduction, Structured matrices, Spectral filtering, Regularization, Noise revealing

I. Hnětynková[1], M. Plešinger[2], Z. Strakoš[3]

hnetynko@karlin.mff.cuni.cz, martin.plesinger@sam.math.ethz.ch, strakos@cs.cas.cz

[1,3]Faculty of Mathematics and Phycics, Charles University, Prague
[2]Seminar of Applied Mathematics, Dept. of Math., ETH Zürich
[1,2,3]Institute of Computer Science, Academy of Sciences of the Czech Republic

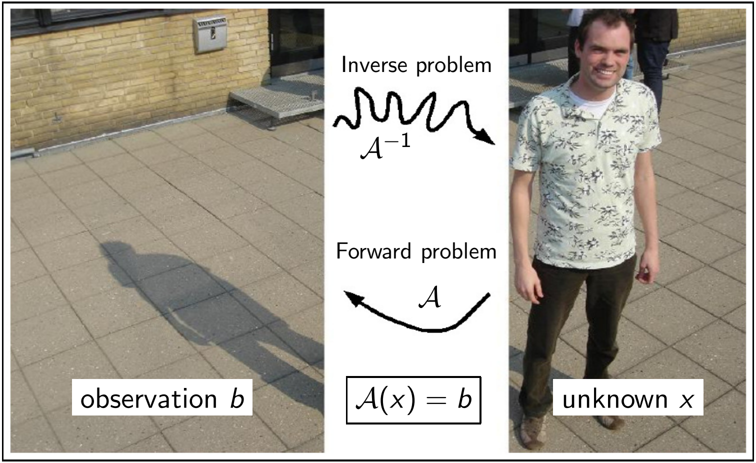
SNA '11, January 24—28

Motivation. A gentle start ...

What is it an inverse problem?

Motivation. A gentle start ...

What is it an inverse problem?



[Kjøller: M.Sc. thesis, DTU Lyngby, 2007].

More realistic examples of ill-posed inverse problems

Computer tomography in medical sciences

Computer tomograph (CT) maps a 3D object of $M \times N \times K$ voxels by ℓ X-ray measurements on ℓ pictures with $m \times n$ pixels,

$\mathcal{A}(\cdot) \equiv$  $: \mathbb{R}^{M \times N \times K} \longrightarrow \bigotimes_{j=1}^{\ell} \mathbb{R}^{m \times n}.$

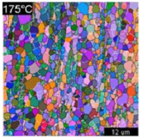
Simpler 2D tomography problem leads to the **Radon transform**. The inverse problem is ill-posed. (3D case is more complicated.)

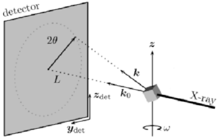
The mathematical problem is **extremely sensitive** to errors which are **always** present in the (measured) data: *discretization error* (finite ℓ, m, n); *rounding errors*; *physical sources of noise* (electronic noise in semiconductor PN-junctions in transistors, ...).

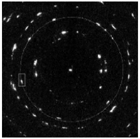
More realistic examples of ill-posed inverse problems

Transmision computer tomography in crystalographics

Reconstruction of an *unknown* orientation distribution function (ODF) of grains in a given sample of a polycrystalline matherial,

$\mathcal{A}\Bigg($ 

$\Bigg) \equiv$ 

$\longrightarrow \Bigg($  $, \dots \Bigg).$

$\underbrace{\qquad\qquad\qquad}_{\text{observation} = \text{data} + \text{noise}}$

The *right-hand side* is a set of measured difractograms.
[Hansen, Sørensen, Südkösd, Poulsen: SIIMS, 2009].

Further analogous applications also in geology, e.g.:

- Seismic tomography (cracks in tectonic plates),
- Gravimetry & magnetometry (ore mineralization).

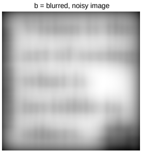
More realistic examples of ill-posed inverse problems

Image deblurring—Our pilot application

Our pilot application is the image deblurring problem

$\mathcal{A}$ (

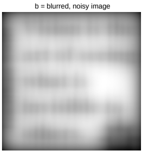
) $\longrightarrow$



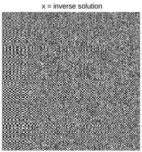
$=$ data $+$ noise.

It leads to a linear system $Ax = b$ with square nonsingular matrix.
Let us motivate our tutorial by a "naive solution" of this system

$\mathcal{A}^{-1}$ (



) $=$



.

[Nagy: Emory University].

More realistic examples of ill-posed inverse problems

General framework

In general we deal with a linear problem

$$Ax = b$$

which typically arose as a discretization of a

Fredholm integral equation of the 1st kind

$$y(\mathbf{s}) = \int K(\mathbf{s}, \mathbf{t}) x(\mathbf{t}) d\mathbf{t}.$$

The observation vector (right-hand side) is contaminated by noise

$$b = b^{\text{exact}} + b^{\text{noise}}, \quad \text{where} \quad \|b^{\text{exact}}\| \gg \|b^{\text{noise}}\|.$$

More realistic examples of ill-posed inverse problems

General framework

We want to compute (approximate)

$$x^{\text{exact}} \equiv A^{-1} b^{\text{exact}}.$$

Unfortunatelly, because the problem is inverse and ill-posed

$$\|A^{-1} b^{\text{exact}}\| \ll \|A^{-1} b^{\text{noise}}\|,$$

the data we look for are in the naive solution covered by the inverted noise. The naive solution

$$x = A^{-1} b = A^{-1} b^{\text{exact}} + A^{-1} b^{\text{noise}}$$

typically has nothing to do with the wanted x^{exact}.

Outline of the tutorial

- **Lecture I—Problem formulation:**

  Mathematical model of blurring, System of linear algebraic equations, Properties of the problem, Impact of noise.

- **Lecture II—Regularization:**

  Basic regularization techniques (TSVD, Tikhonov), Criteria for choosing regularization parameters, Iterative regularization, Hybrid methods.

- **Lecture III—Noise revealing:**

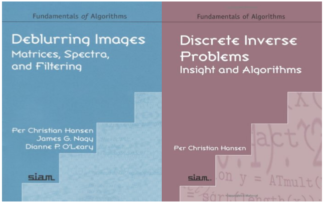
  Golub-Kahan iteratie bidiagonalization and its properties, Propagation of noise, Determination of the noise level, Noise vector approximation, Open problems.

Textbooks:

- Hansen, Nagy, O'Leary: *Deblurring Images, Spectra, Matrices, and Filtering*, SIAM, FA03, 2006.
- Hansen: *Discrete Inverse Problems, Insight and Algorithms*, SIAM, FA07, 2010.

Sofwtare (MatLab toolboxes):

- HNO package,
- Regularization tools,
- AIRtools,
- ...



(software available on the homepage of P. C. Hansen).

Outline of Lecture I

- **1. Mathematical model of blurring:**

  Blurring as an operator on the vector space of matrices, Linear and spatial invariant operator, Point-spread-function, 2D convolution, Boundary conditions.

- **2. System of linear algebraic equations:**

  Gaußian blur, Exploiting the separability, 1D Gaußian blurring operator, Boundary conditions, 2D Gaußian blurring operator, Structured matrices.

- **3. Properties of the problem:**

  Smoothing properties, Singular vectors of A, Singular values of A, The right-hand side, Discrete Picard condition (DPC), SVD and Image deblurring problem, Singular images.

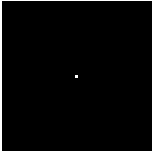
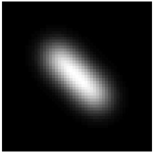
- **4. Impact of noise:**

  Violation of DPC, Naive solution, Regularization and filtering.

1. Mathematical model of blurring

The **grayscale image** can be considered as a **matrix**, consider for convenience *black* $\equiv 0$ and *white* $\equiv 1$.

Consider a, so called, **single-pixel-image (SPI)** and a blurring operator as follows

$\mathcal{A}(X) = \mathcal{A}\Bigg($  $\Bigg) =$  $= B,$

where $X = [x_1, \ldots, x_k]$, $B = [b_1, \ldots, b_k] \in \mathbb{R}^{k \times k}$.

The image (matrix) B is called **point-spread-function (PSF)**.

(In Parts 1, 2, 3 we talk about the operator, the right-hand side is noise-free.)

1. Mathematical model of blurring

Linear and spatial invariant operator

Consider $\mathcal{A}$ to be:

1. linear (additive & homogenous),
2. spatial invariant.

Linearity of $\mathcal{A}$ allows to rewrite $\mathcal{A}(X) = B$ as a system of linear algebraic equations

$$Ax = b, \qquad A \in \mathbb{R}^{N \times N}, \quad x, \; b \in \mathbb{R}^N.$$

(We do not know how, yet.)

1. Mathematical model of blurring

Linear and spatial invariant operator

The matrix X containing the SPI has only one nonzero entry (moreover equal to one).

Therefore the unfolded X

$$x = \mathrm{vec}(X) = [x_1^T, \ldots, x_k^T]^T \; = \; e_j$$

represents an Euclidean vector.

The unfolding of the corredponding B (containing the PSF) then represents jth column of A

$$A\, e_j = b = \mathrm{vec}(B) = [b_1^T, \ldots, b_k^T]^T.$$

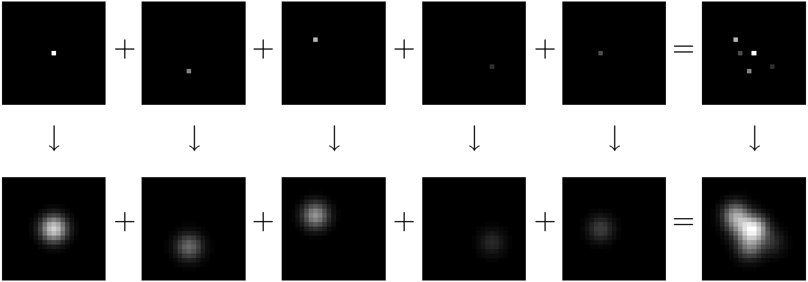
The matrix A is composed columnwise by unfolded PSFs corresponding to SPIs with different positions of the nonzero pixel.

1. Mathematical model of blurring

Linear and spatial invariant operator

Spatial invariance of $\mathcal{A}$ $\equiv$ The PSF is the same for all positions of the nonzero pixel in SPI. (What about pixels close to the border?)

Linearity + spatial invariance:



First row: Original (SPI) images (matrices X).
Second row: Blurred (PSF) images (matrices $B = \mathcal{A}(X)$).

1. Mathematical model of blurring

Point—spread—function (PSF)

Linear and spatially invariant blurring operator $\mathcal{A}$ is **fully described** by its action on one SPI, i.e. **by one PSF**. (Which one?)

Recall: Up to now the *width* and *height* of both the SPI and PSF images have been equal to some k, called the **window size**.

For correctness the window size must be properly chosen, namely:

- the window size must be sufficiently large

  (increase of k leads to extension of PSF image by black),
- the window is typically square (for simplicity),
- we use window of odd size (for simplicity), i.e.

$$k = 2\ell + 1.$$

1. Mathematical model of blurring

Point—spread—function (PSF)

The square window with sufficiently large odd size $k = 2\ell + 1$ allows to consider SPI image given by the matrix

$$SPI = e_{\ell+1} e_{\ell+1}^T \in \mathbb{R}^{k\times k}$$

(the only nonzero pixel is in the middle of SPI).

The corresponding PSF image given by the matrix

$$PSF_{\mathcal{A}} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,k} \\ \vdots & \ddots & \vdots \\ p_{k,1} & \cdots & p_{k,k} \end{bmatrix} = \begin{bmatrix} \bar{p}_{-\ell,-\ell} & \cdots & \bar{p}_{-\ell,+\ell} \\ \vdots & \ddots & \vdots \\ \bar{p}_{+\ell,-\ell} & \cdots & \bar{p}_{+\ell,+\ell} \end{bmatrix} \in \mathbb{R}^{k\times k}$$

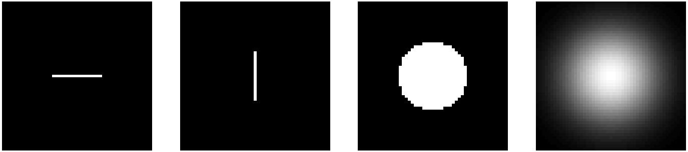
will be further used for the description of the operator $\mathcal{A}$.

1. Mathematical model of blurring

Point—spread—function (PSF)

Examples of $PSF_{\mathcal{A}}$:

horizontal motion blur	vertical motion blur	out-of-focus blur	**Gaußian blur**



1. Mathematical model of blurring

2D convolution

We have the linear, spatial invariant $\mathcal{A}$ given by $PSF_{\mathcal{A}} \in \mathbb{R}^{k \times k}$.

Consider a general grayscale image given by a matrix $X \in \mathbb{R}^{m \times n}$.

How to realize the action of $\mathcal{A}$ on X, i.e. $B = \mathcal{A}(X)$, using $PSF_{\mathcal{A}}$?

Entrywise application of PSF:

1. $X = \sum_{i=1}^{m} \sum_{j=1}^{n} X_{i,j}$, where $X_{i,j} \equiv x_{i,j}(e_i e_j^T) \in \mathbb{R}^{m \times n}$;
2. realize the action of $\mathcal{A}$ on the single-pixel-image $X_{i,j}$

$$X_{i,j} = \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ \hline 0 & x_{i,j} SPI & 0 \\ \hline 0 & 0 & 0 \end{array}\right] \longrightarrow B_{i,j} \equiv \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ \hline 0 & x_{i,j} PSF_{\mathcal{A}} & 0 \\ \hline 0 & 0 & 0 \end{array}\right];$$

3. $B = \sum_{i=1}^{m} \sum_{j=1}^{n} B_{i,j}$ due to the linearity of $\mathcal{A}$.

1. Mathematical model of blurring

2D convolution

5-by-5 example: $B = \sum_{i=1}^{m} \sum_{j=1}^{n} B_{i,j} = x_{1,1}(\,) + \ldots + x_{1,5}(\,)$

$$+x_{2,1}(\,) + x_{2,2}\begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} & 0 & 0 \\ p_{2,1} & p_{2,2} & p_{2,3} & 0 & 0 \\ p_{3,1} & p_{3,2} & p_{3,3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + x_{2,3}\begin{pmatrix} 0 & p_{1,1} & p_{1,2} & p_{1,3} & 0 \\ 0 & p_{2,1} & p_{2,2} & p_{2,3} & 0 \\ 0 & p_{3,1} & p_{3,2} & p_{3,3} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + x_{2,4}\begin{pmatrix} 0 & 0 & p_{1,1} & p_{1,2} & p_{1,3} \\ 0 & 0 & p_{2,1} & p_{2,2} & p_{2,3} \\ 0 & 0 & p_{3,1} & p_{3,2} & p_{3,3} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + x_{2,5}(\,)$$

$$+x_{3,1}(\,) + x_{3,2}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ p_{1,1} & p_{1,2} & p_{1,3} & 0 & 0 \\ p_{2,1} & p_{2,2} & p_{2,3} & 0 & 0 \\ p_{3,1} & p_{3,2} & p_{3,3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + x_{3,3}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & p_{1,1} & p_{1,2} & p_{1,3} & 0 \\ 0 & p_{2,1} & p_{2,2} & p_{3,3} & 0 \\ 0 & p_{3,1} & p_{3,2} & p_{3,3} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + x_{3,4}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{1,1} & p_{1,2} & p_{1,3} \\ 0 & 0 & p_{2,1} & p_{2,2} & p_{2,3} \\ 0 & 0 & p_{3,1} & p_{3,2} & p_{3,3} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + x_{3,5}(\,)$$

$$+x_{4,1}(\,) + x_{4,2}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ p_{1,1} & p_{1,2} & p_{1,3} & 0 & 0 \\ p_{2,1} & p_{2,2} & p_{2,3} & 0 & 0 \\ p_{3,1} & p_{3,2} & p_{3,3} & 0 & 0 \end{pmatrix} + x_{4,3}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & p_{1,1} & p_{1,2} & p_{1,3} & 0 \\ 0 & p_{2,1} & p_{2,2} & p_{2,3} & 0 \\ 0 & p_{3,1} & p_{3,2} & p_{3,3} & 0 \end{pmatrix} + x_{4,4}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{1,1} & p_{1,2} & p_{1,3} \\ 0 & 0 & p_{2,1} & p_{2,2} & p_{2,3} \\ 0 & 0 & p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix} + x_{4,5}(\,)$$

$+x_{5,1}(\,) + \ldots + x_{5,5}(\,)$, where

$$PSF_{\mathcal{A}} = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,1} & p_{3,2} & p_{3,3} \end{pmatrix}, \qquad \begin{aligned} b_{3,3} = {} & x_{2,2}\,p_{3,3} + x_{2,3}\,p_{3,2} + x_{2,4}\,p_{3,1} \\ + {} & x_{3,2}\,p_{2,3} + x_{3,3}\,p_{2,2} + x_{3,4}\,p_{2,1} \\ + {} & x_{4,2}\,p_{1,3} + x_{4,3}\,p_{1,2} + x_{4,4}\,p_{1,1} \end{aligned}.$$

1. Mathematical model of blurring

2D convolution

The entry $b_{i,j}$ of B is influenced by the entry $x_{i,j}$ and a few entries in its surroundings, depending on the support of $PSF_{\mathcal{A}}$.

In general:

$$b_{i,j} = \sum_{h=-\ell}^{\ell} \sum_{w=-\ell}^{\ell} x_{i-h,j-w} \bar{p}_{h,w}.$$

The blured image represented by matrix B is therefore the

2D convolution

of X with $PSF_{\mathcal{A}}$.

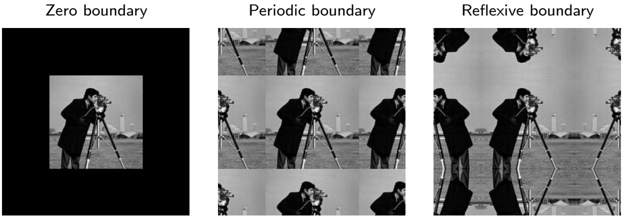
Boundary: Pixels $x_{\mu,\nu}$ for $\mu \in \mathbb{Z} \setminus [1, \ldots, m]$ or $\nu \in \mathbb{Z} \setminus [1, \ldots, n]$ ("outside" the original image X) are not given.

1. Mathematical model of blurring

Boundary conditions (BC)

Real-world blurred image B is involved by the information which is outside the scene X, i.e. by the boundary pixels $x_{\mu,\nu}$.
For the reconstruction of the real-world scene (deblurring) we do have to consider some **boundary condition**:

- Outside the scene is **nothing**, $x_{\mu,\nu} = 0$ (black), e.g., in astrononomical observations.
- The scene contains **periodic** patterns, e.g., in micro/nanoscale imaging of matherials.
- The scene can be prolongated by **reflecting**.



1. Mathematical model of blurring

Summary

Now we know "everything" about the simplest mathematical model of blurring:

- We consider linear, spatial invariant operator $\mathcal{A}$, which is represented by its point-spread-function $PSF_{\mathcal{A}}$.
- The 2D convolution of true scene with the point-spread-function represents the blurring.
- Convolution uses some information from the outside of the scene, therefore we need to consider some boundary conditions.

2. System of linear algebraic equations

2. System of linear algebraic equations

Basic concept

The problem $\mathcal{A}(X) = B$ can be rewritten (emploing the 2D convolution formula) as a system of linear algebraic equations

$$Ax = b, \qquad A \in \mathbb{R}^{mn \times mn}, \quad x = \mathrm{vec}(X), \; b = \mathrm{vec}(B) \in \mathbb{R}^{mn},$$

where the entries of A are the entries of the PSF, and

$$b_{i,j} = \sum\nolimits_{h=-\ell}^{\ell} \sum\nolimits_{w=-\ell}^{\ell} x_{i-h,j-w} \bar{p}_{h,w}.$$

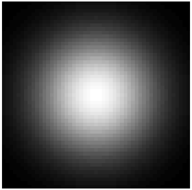
In general:

- PSF has small localized support,
- each pixel is influenced only by a few pixels in its close surroundings,
- therefore A is **sparse**.

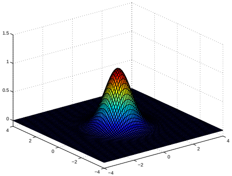
2. System of linear algebraic equations

Gaußian PSF / Gaußian blur

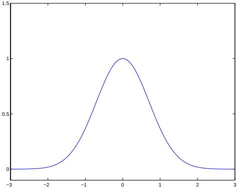
In the rest we consider **Gaußian blur**:



Gaußian PSF



$G_{2\mathrm{D}}(h, w)$



$G_{1\mathrm{D}}(\xi)$

where (in a continuous domain)

$$G_{2\mathrm{D}}(h, w) = e^{-(h^2+w^2)} = e^{-h^2} e^{-w^2}, \qquad G_{1\mathrm{D}}(\xi) = e^{-\xi^2}.$$

Gaußian blur is the simplest and in many cases sufficient model.
A big advantage is its **separability** $G_{2\mathrm{D}}(h, w) = G_{1\mathrm{D}}(h) G_{1\mathrm{D}}(w)$.

2. System of linear algebraic equations

Exploiting the separability

Consider the 2D convolution with Gaußian PSF in a continuous domain. Exploiting the separability, we get

$$
\begin{aligned}
B(i,j) &= \iint_{\mathbb{R}^2} X(i-h, j-w)\, e^{-(h^2+w^2)}\, dh\, dw \\
&= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} X(i-h, j-w)\, e^{-h^2}\, dh \right) e^{-w^2} dw \\
&= \int_{-\infty}^{\infty} Y(i, j-w) e^{-w^2} dw,
\end{aligned}
$$

where $$Y(i,j) = \int_{-\infty}^{\infty} X(i-h, j)\, e^{-h^2}\, dh.$$

The blurring in the direction h (height) is **independent** on the blurring in the direction w (width).
In the discrete setting: The blurring of columns of X is **independent** on the blurring of rows of X.

2. System of linear algebraic equations

Exploiting the separability

As a direct consequence of the separability, the PSF matrix is a **rank one** matrix of the form

$$PSF_{\mathcal{A}} = cr^T, \quad c,\, r \in \mathbb{R}^k.$$

The blurring of columns (rows) of X is realized by 1D (discrete) convolution with c (r), the discretized $G_{\mathrm{1D}}(\xi) = e^{-\xi^2}$.

Let A_C, A_R be matrices representing discete 1D Gaußian blurring operators, where

- A_C realizes blurring of columns of X,
- A_R^T realizes blurring of rows of X.

Then the problem $\mathcal{A}(X) = B$ can be rewritten as a **matrix equation**

$$A_C\, X\, A_R^T = B, \qquad A_C \in \mathbb{R}^{m\times m}, \quad A_R \in \mathbb{R}^{n\times n}.$$

2. System of linear algebraic equations

1D convolution

Consider the following example of an A_C related 1D convolution:

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{bmatrix} = \left[\begin{array}{cc|cccccc|cc} c_5 & c_4 & c_3 & c_2 & c_1 & & & & & \\ & c_5 & c_4 & c_3 & c_2 & c_1 & & & & \\ & & c_5 & c_4 & c_3 & c_2 & c_1 & & & \\ & & & c_5 & c_4 & c_3 & c_2 & c_1 & & \\ & & & & c_5 & c_4 & c_3 & c_2 & c_1 & \\ & & & & & c_5 & c_4 & c_3 & c_2 & c_1 \end{array}\right] \left[\begin{array}{c} \xi_{-1} \\ \xi_0 \\ \hline \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \hline \xi_7 \\ \xi_8 \end{array}\right],$$

where $b = [\beta_1, \ldots, \beta_6]^T$, $x = [\xi_1, \ldots, \xi_6]^T$,
and $c = [c_1, \ldots, c_5]^T$ is the 1D (Gaußian) point-spread-function.

2. System of linear algebraic equations

Boundary conditions

The vector $[\xi_{-1}, \xi_0 | \xi_1, \ldots, \xi_6 | \xi_7, \xi_8]^T$ represents the true scene. In the reconstruction we consider:

$$\begin{aligned}
[0, 0 | \xi_1, \ldots, \xi_6 | 0, 0]^T &\sim \text{ zero boundary condition,} \\
[\xi_5, \xi_6 | \xi_1, \ldots, \xi_6 | \xi_1, \xi_2]^T &\sim \text{ periodic boundary condition, or} \\
[\xi_2, \xi_1 | \xi_1, \ldots, \xi_6 | \xi_6, \xi_5]^T &\sim \text{ reflexive boundary condition.}
\end{aligned}$$

In general $A_C = M + BC$, where

$$M = \begin{bmatrix} c_3 & c_2 & c_1 & & & \\ c_4 & c_3 & c_2 & c_1 & & \\ c_5 & c_4 & c_3 & c_2 & c_1 & \\ & c_5 & c_4 & c_3 & c_2 & c_1 \\ & & c_5 & c_4 & c_3 & c_2 \\ & & & c_5 & c_4 & c_3 \end{bmatrix},$$

and BC is a correction due to the boundary conditions.

2. System of linear algebraic equations

Boundary conditions

Zero boundary condition:

$$A_C x = \left[\begin{array}{cc|cccccc|cc} c_5 & c_4 & c_3 & c_2 & c_1 & & & & & \\ & c_5 & c_4 & c_3 & c_2 & c_1 & & & & \\ & & c_5 & c_4 & c_3 & c_2 & c_1 & & & \\ & & & c_5 & c_4 & c_3 & c_2 & c_1 & & \\ & & & & c_5 & c_4 & c_3 & c_2 & c_1 & \\ & & & & & c_5 & c_4 & c_3 & c_2 & c_1 \end{array}\right] \left[\begin{array}{c} 0 \\ 0 \\ \hline \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \hline 0 \\ 0 \end{array}\right] = \begin{bmatrix} c_3 & c_2 & c_1 & & & \\ c_4 & c_3 & c_2 & c_1 & & \\ c_5 & c_4 & c_3 & c_2 & c_1 & \\ & c_5 & c_4 & c_3 & c_2 & c_1 \\ & & c_5 & c_4 & c_3 & c_2 \\ & & & c_5 & c_4 & c_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \end{bmatrix},$$

i.e. here $BC = 0$ and $A_C = M$ is a **Toeplitz matrix**.

Periodic boundary condition:

$$A_C x = \left[\begin{array}{cc|cccccc|cc} c_5 & c_4 & c_3 & c_2 & c_1 & & & & & \\ & c_5 & c_4 & c_3 & c_2 & c_1 & & & & \\ & & c_5 & c_4 & c_3 & c_2 & c_1 & & & \\ & & & c_5 & c_4 & c_3 & c_2 & c_1 & & \\ & & & & c_5 & c_4 & c_3 & c_2 & c_1 & \\ & & & & & c_5 & c_4 & c_3 & c_2 & c_1 \end{array}\right] \left[\begin{array}{c} \xi_5 \\ \xi_6 \\ \hline \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \hline \xi_1 \\ \xi_2 \end{array}\right] = \begin{bmatrix} c_3 & c_2 & c_1 & & c_5 & c_4 \\ c_4 & c_3 & c_2 & c_1 & & c_5 \\ c_5 & c_4 & c_3 & c_2 & c_1 & \\ & c_5 & c_4 & c_3 & c_2 & c_1 \\ c_1 & & c_5 & c_4 & c_3 & c_2 \\ c_2 & c_1 & & c_5 & c_4 & c_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \end{bmatrix},$$

$$\text{i.e. here} \quad BC = \begin{bmatrix} & & & & c_5 & c_4 \\ & & & & & c_5 \\ & & & & & \\ & & & & & \\ c_1 & & & & & \\ c_2 & c_1 & & & & \end{bmatrix}$$

and $A_C = M + BC$ is a **circulant matrix**.

2. System of linear algebraic equations

Boundary conditions

Reflexive boundary condition:

$$A_C x = \left[\begin{array}{cc|cccccc|cc} c_5 & c_4 & c_3 & c_2 & c_1 & & & & & \\ & c_5 & c_4 & c_3 & c_2 & c_1 & & & & \\ & & c_5 & c_4 & c_3 & c_2 & c_1 & & & \\ & & & c_5 & c_4 & c_3 & c_2 & c_1 & & \\ & & & & c_5 & c_4 & c_3 & c_2 & c_1 & \\ & & & & & c_5 & c_4 & c_3 & c_2 & c_1 \end{array}\right] \left[\begin{array}{c} \xi_2 \\ \xi_1 \\ \hline \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \hline \xi_6 \\ \xi_5 \end{array}\right] = \begin{bmatrix} c_3+c_4 & c_2+c_5 & c_1 & & & \\ c_4+c_5 & c_3 & c_2 & c_1 & & \\ c_5 & c_4 & c_3 & c_2 & c_1 & \\ & c_5 & c_4 & c_3 & c_2 & c_1 \\ & & c_5 & c_4 & c_3 & c_2+c_1 \\ & & & c_5 & c_4+c_1 & c_3+c_2 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \end{bmatrix},$$

$$\text{i.e. here} \quad BC = \begin{bmatrix} c_4 & c_5 & & \\ c_5 & & & \\ & & & c_1 \\ & & c_1 & c_2 \end{bmatrix}$$

and $A_C = M + BC$ is a **Toeplitz-plus-Hankel matrix**.

2. System of linear algebraic equations

Boundary conditions—Summary

Three types of boundary conditions:

- zero boundary condition,
- periodic boundary condition,
- reflexive boundary condition,

correspond to the three types of matrices A_C and A_R:

- Toeplitz matrix,
- circulant matrix,
- Toeplitz-plus-Hankel matrix,

in the linear system of the form

$$A_C \, X \, A_R^T = B.$$

Now we show how to rewrite the matrix equation $A_C\,X\,A_R^T = B$ as a system of linear algebraic equations in a usual form.

Consider $A_R = I_n$. The matrix equation

$$A_C\,X = B$$

can be rewritten as

$$(I_n \otimes A_C)\,\mathrm{vec}(X) = \begin{bmatrix} A_C & & \\ & \ddots & \\ & & A_C \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} = \mathrm{vec}(B),$$

where $X = [x_1, \ldots, x_n]$, $B = [b_1, \ldots, b_n]$,
and $\otimes$ denotes the **Kronecker product**.

Consider $A_C = I_m$. The matrix equation $X\,A_R^T = B$ can be rewritten as

$$(A_R \otimes I_m)\,\mathrm{vec}(X) = \begin{bmatrix} a^R_{1,1} I_m & \cdots & a^R_{1,n} I_m \\ \vdots & \ddots & \vdots \\ a^R_{n,1} I_m & \cdots & a^R_{n,n} I_m \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} = \mathrm{vec}(B).$$

Consequently $A_C\,X\,A_R^T = (A_C\,X)\,A_R^T$ gives

$$(A_R \otimes I_m)\mathrm{vec}(A_C\,X) = (A_R \otimes I_m)(I_n \otimes A_C)\mathrm{vec}(X).$$

Using properties of Kronecker product, this system is equivalent to

$$Ax = (A_R \otimes A_C)\,\mathrm{vec}(X) = \mathrm{vec}(B) = b,$$

where

$$A = \begin{bmatrix} a^R_{1,1} A_C & \cdots & a^R_{1,n} A_C \\ \vdots & \ddots & \vdots \\ a^R_{n,1} A_C & \cdots & a^R_{n,n} A_C \end{bmatrix} \in \mathbb{R}^{mn \times mn}.$$

We have

$$A = A_R \otimes A_C = \begin{bmatrix} a^R_{1,1} A_C & \cdots & a^R_{1,n} A_C \\ \vdots & \ddots & \vdots \\ a^R_{n,1} A_C & \cdots & a^R_{n,n} A_C \end{bmatrix} \in \mathbb{R}^{mn \times mn},$$

where A_C, A_R are Toeplitz, circulant, or Toeplitz-plus-Hankel.

If A_C is Toeplitz, then A is a matrix with Toeplitz blocks.

If A_R is Toeplitz, then A is a block-Toeplitz matrix.

If A_C and A_R are Toeplitz (zero BC), then A is

block—Toeplitz with Toeplitz blocks (BTTB).

Analogously, for periodic BC we get **BCCB** matrix, for reflexie BC we get a sum of four matrices **BTTB+BTHB+BHTB+BHHB**.

3. Properties of the problem

3. Properties of the problem

Smoothing properties

We have an inverse ill-posed problem $Ax = b$, a discretization of a Fredholm integral equation of the 1st kind

$$y(\mathbf{s}) = \int K(\mathbf{s}, \mathbf{t}) x(\mathbf{t}) d\mathbf{t}.$$

The matrix A is a restriction of the integral kernel $K(\mathbf{s}, \mathbf{t})$ (the convolution kernel in image deblurring)

- the kernel $K(\mathbf{s}, \mathbf{t})$ has **smoothing property**,
- therefore the vector $y(\mathbf{s})$ is smooth,

and these properties are inherited by the discretized problem. Further analysis is based on the singular value decomposition

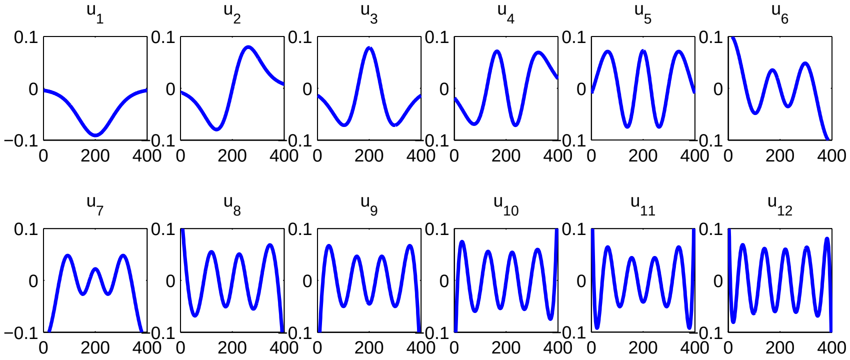
$$A = U \Sigma V^T, \qquad U \in \mathbb{R}^{N \times N}, \quad \Sigma \in \mathbb{R}^{N \times N}, \quad V \in \mathbb{R}^{N \times N},$$

(and $N = mn$ in image deblurring).

3. Properties of the problem

Singular vectors of A

Singular vectors of A represent bases with increasing frequencies:

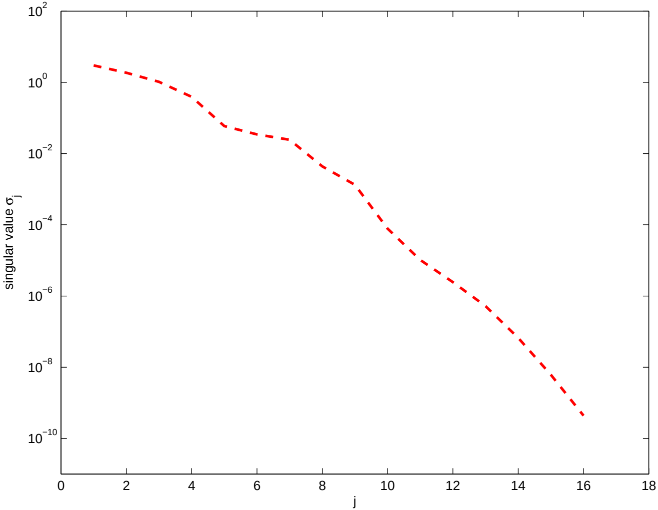


First 12 left singular vectors of 1D ill-posed problem **SHAW(400)** [Regularization Toolbox].

3. Properties of the problem

Singular values of A

Singular values decay without a noticeable gap (SHAW(400)):



First recall that b is the discretized smooth $y(\mathbf{s})$, therefore

b is smooth, i.e. dominated by low frequencies.

Thus b has large components in directions of several first vectors u_j, and $|u_j^T b|$ **on average decay with** j.

3. Properties of the problem

The Discrete Picard condition

Using the dyadic form of SVD

$$A = \sum_{j=1}^{N} u_j \sigma_j v_j^T, \quad N \text{ is the dimension of the discretized } K(\mathbf{s}, \mathbf{t}),$$

the solution of $Ax = b$ can be rewritten as a linear combination of right-singular vectors,

$$x = A^{-1} b = \sum_{j=1}^{N} \frac{u_j^T b}{\sigma_j} v_j.$$

Since x is a discretization of some real-world object $x(\mathbf{t})$ (e.g., an "true image") the sequence of these sums converges to $x(\mathbf{t})$ with $N \longrightarrow \infty$.

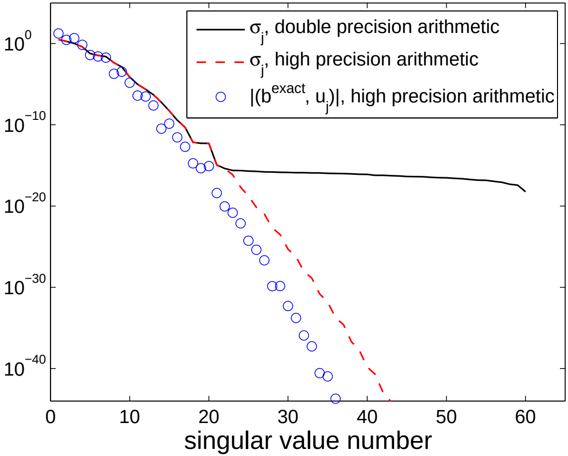
This is possible only if $|u_j^T b|$ are on average decay faster than σ_j.

This property is called the **(discrete) Picard condition (DPC)**.

3. Properties of the problem

The Discrete Picard condition

The discrete Picard condition (SHAW(400)):



3. Properties of the problem

SVD and Image deblurring problem

Back to the image deblurring problem: We have

$$A_C\,X\,A_R^T = B \quad \Longleftrightarrow \quad (A_R \otimes A_C)\,\mathrm{vec}(X) = \mathrm{vec}(B).$$

Consider SVDs of both A_C and A_R

$$A_C = U_C\,\mathrm{diag}(s_C)V_C^T, \qquad A_R = U_R\,\mathrm{diag}(s_R)V_R^T,$$
$$s_C = [\sigma_1^C, \ldots, \sigma_m^C]^T \in \mathbb{R}^m, \qquad s_R = [\sigma_1^R, \ldots, \sigma_n^R]^T \in \mathbb{R}^n.$$

Using the basic properties of the Kronecker product

$$\begin{aligned} A = A_R \otimes A_C &= (U_R\,\mathrm{diag}(s_R)V_R^T) \otimes (U_C\,\mathrm{diag}(s_C)V_C^T) \\ &= (U_R \otimes U_C)\,\mathrm{diag}(s_R \otimes s_C)(V_R \otimes V_C)^T = U\Sigma V^T, \end{aligned}$$

we get SVD of A (up to the ordering of singular values).

3. Properties of the problem

SVD and Image deblurring problem

The solution of $A_C \, X \, A_R^T = B$ can be written directly as

$$X = V_C \underbrace{(\overbrace{(U_C^T \; B \; U_R)}^{\text{projections } u_j^T b} \oslash (s_C \, s_R^T))}_{\text{fractions } (u_j^T b)/\sigma_j} V_R^T ,$$

where $K \oslash M$ denotes the Hadamard product of K with the componentwise inverse of M (using MatLab notation `K./M`).

Or using the dyadic expansion as

$$x = \sum_{j=1}^{N} \frac{u_j^T \, \mathrm{vec}(B)}{\sigma_j} \, v_j, \qquad X = \mathrm{mtx}(x), \qquad N = mn,$$

where $\mathrm{mtx}(\cdot)$ denotes an inverse mapping to $\mathrm{vec}(\cdot)$.

3. Properties of the problem
Singular images

The solution

$$x = \sum_{j=1}^{N} \underbrace{\frac{u_j^T \operatorname{vec}(B)}{\sigma_j}}_{\text{scalar}} v_j, \qquad X = \operatorname{mtx}(x), \qquad N = mn,$$

is a linear combination of right singular vectors v_j.

It can be further rewritten as

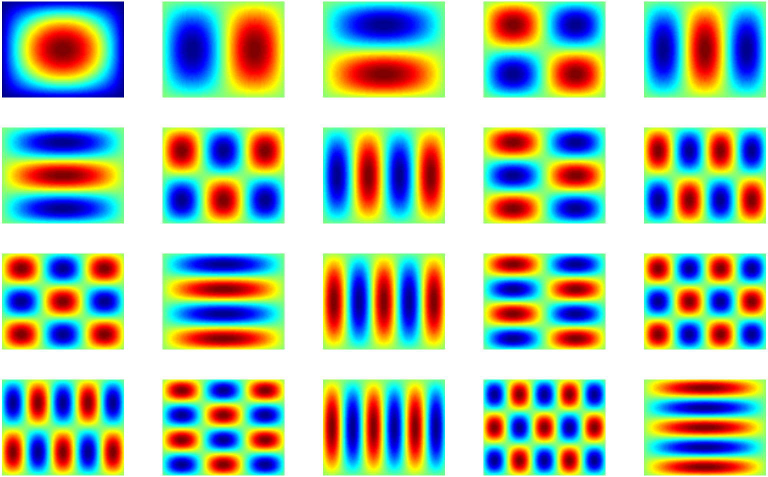
$$X = \sum_{j=1}^{N} \frac{u_j^T \operatorname{vec}(B)}{\sigma_j} V_j, \qquad V_j = \operatorname{mtx}(v_j) \in \mathbb{R}^{m \times n}$$

using **singular images** V_j (the reshaped right singular vectors).

3. Properties of the problem

Singular images

Singular images V_j (Gaußian blur, zero BC, artificial colors)



3. Properties of the problem

Note on computation of SVD

Recall that the matrices A_C, A_R are

- Toeplitz,
- circulant, or
- Toeplitz-plus-Hankel,

and often symmetric (depending on the symmetry of PSF).

Toeplitz matrix is fully determined by its first column and row,
circulant matrix by its first column (or row), and
Hankel matrix by the first column and the last row.

Eigenvalue decomposition (SVD) of such matrices can be efficiently computed using **discrete Fourier transform** (DFT/FFT algorithm), or **discrete cosine transform** (DCT algorithm).

4. Impact of noise

4. Impact of noise

Noise, Sources of noise

Consider a problem of the form

$$Ax = b, \quad b = b^{\text{exact}} + b^{\text{noise}}, \quad \|b^{\text{exact}}\| \gg \|b^{\text{noise}}\|,$$

where b^{noise} is unknown and represents, e.g.,

- rounding errors,
- discretization error,
- noise with physical sources (electronic noise on PN-junctions).

We want to approximate

$$x^{\text{exact}} \equiv A^{-1} b^{\text{exact}},$$

unfortunately

$$\|A^{-1} b^{\text{exact}}\| \ll \|A^{-1} b^{\text{noise}}\|.$$

4. Impact of noise

Violation of the discrete Picard condition

The vector b^{noise} typically resebles **white noise**, i.e. it has flat frequency characteristics.

Recall that the singular vectors of A represent frequencies.

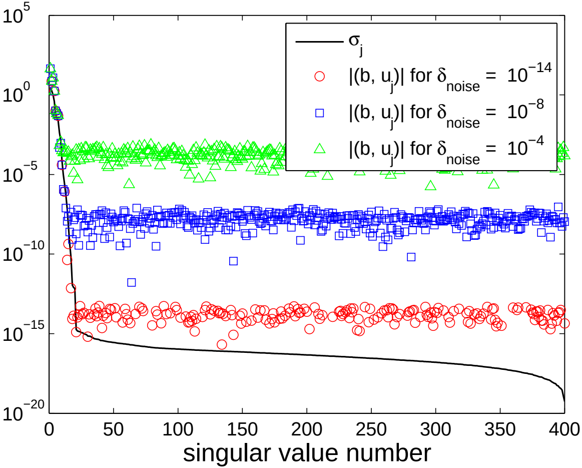
Thus the white noise components in left singular subspaces are about the same order of magnitude.
White noise

violates the discrete Picard condition.

Summarizing:

- b^{exact} has some real pre-image x^{exact}, it satifies DPC
- b^{noise} does not have any real pre-image, it violates DPC.

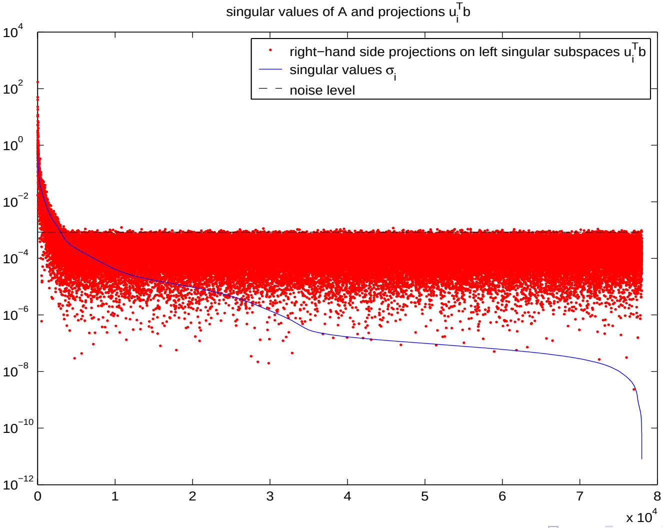
Violation of the discrete Picard condition by noise (SHAW(400)):



4. Impact of noise

Violation of the discrete Picard condition

Violation the dicrete Picard condition by noise (Image deb. pb.):



4. Impact of noise

Violation of the discrete Picard condition

Using $b = b^{\text{exact}} + b^{\text{noise}}$ we can write the expansion

$$x^{\text{naive}} \equiv A^{-1} b = \sum_{j=1}^{N} \frac{u_j^T b}{\sigma_j} v_j$$

$$= \underbrace{\sum_{j=1}^{N} \frac{u_j^T b^{\text{exact}}}{\sigma_j} v_j}_{x^{\text{exact}}} + \underbrace{\sum_{j=1}^{N} \frac{u_j^T b^{\text{noise}}}{\sigma_j} v_j}_{\text{amplified noise}} .$$

Because σ_j decay and $|u_j^T b^{\text{noise}}|$ are all about the same size, $|u_j^T b^{\text{noise}}|/\sigma_j$ grow for large j. However, $|u_j^T b^{\text{exact}}|/\sigma_j$ decay with j due to DPC. Thus the high-frequency noise covers all sensefull information in x^{naive}.

Therefore x^{naive} is called the **naive solution**.

⟨MatLab demo⟩

4. Impact of noise

Regularization and filtering

To avoid the catastrophical impact of noise we employ regularization techniques.

In general the regularization can be understood as a filtering

$$x^{\mathrm{filtered}} \equiv \sum_{j=1}^{N} \phi_j \, \frac{u_j^T b}{\sigma_j} \, v_j,$$

where the filter factors ϕ_j are given by some filter function $\phi_j = \phi(j, A, b, \ldots)$.

⟨Lecture II⟩

Summary

- We have an discrete inverse problem which is **ill-posed**. Our observation is often corrupted by (white) noise and we want to reconstruct the true pre-image of this observation.
- The whole concept was illustrated on the **image deblurring problem**, which was closely introduced and described.
- It was shown how the problem can be reformulated as a **system of linear algebraic equations**.
- We showed the typical **properties** of the corresponding matrix and the right-hand side, in particular the **discrete Picard condition**.
- Finally, we illustrated the catastrophical **impact of the noise** on the reconstruction on an example.