10. Multi-objective least squares

- multi-objective least squares
- regularized data fitting
- control
- estimation and inversion

Multi-objective least squares

we have several objectives

$$J_1 = \|A_1 x - b_1\|^2, \qquad \ldots, \qquad J_k = \|A_k x - b_k\|^2$$

- A_i is an $m_i \times n$ matrix, b_i is an m_i-vector
- we seek *one* x that makes all k objectives small
- usually there is a trade-off: no single x minimizes all objectives simultaneously

Weighted least squares formulation: find x that minimizes

$$\lambda_1 \|A_1 x - b_1\|^2 + \cdots + \lambda_k \|A_k x - b_k\|^2$$

- coefficients $\lambda_1, \ldots, \lambda_k$ are positive weights
- weights λ_i express relative importance of different objectives
- without loss of generality, we can choose $\lambda_1 = 1$

Solution of weighted least squares

- weighted least squares is equivalent to a standard least squares problem

$$\text{minimize} \quad \left\| \begin{bmatrix} \sqrt{\lambda_1} A_1 \\ \sqrt{\lambda_2} A_2 \\ \vdots \\ \sqrt{\lambda_k} A_k \end{bmatrix} x - \begin{bmatrix} \sqrt{\lambda_1} b_1 \\ \sqrt{\lambda_2} b_2 \\ \vdots \\ \sqrt{\lambda_k} b_k \end{bmatrix} \right\|^2$$

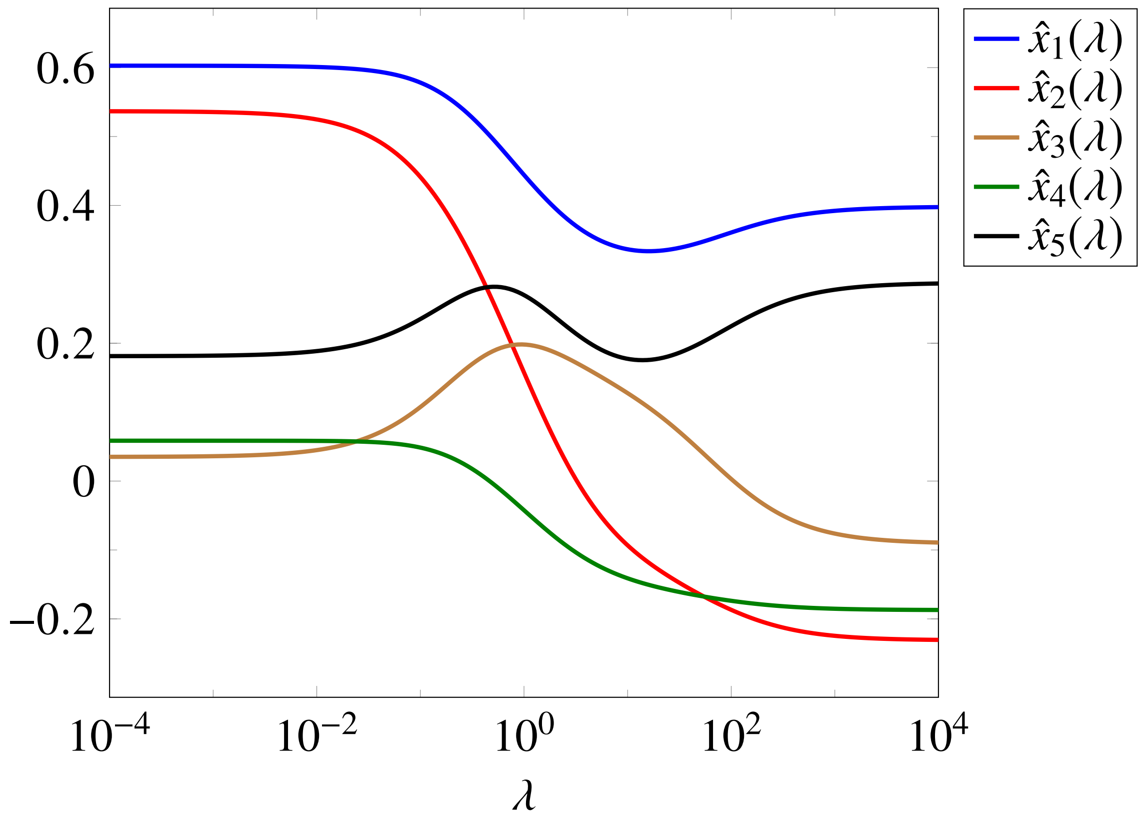
- solution is unique if the *stacked matrix* has linearly independent columns
- each matrix A_i may have linearly dependent columns (or be a wide matrix)
- it the stacked matrix has linearly independent columns, the solution is

$$\hat{x} = \left(\lambda_1 A_1^T A_1 + \cdots + \lambda_k A_k^T A_k\right)^{-1} \left(\lambda_1 A_1^T b_1 + \cdots + \lambda_k A_k^T b_k\right)$$

Example with two objectives

$$\text{minimize} \quad \|A_1x - b_1\|^2 + \lambda\|A_2x - b_2\|^2$$

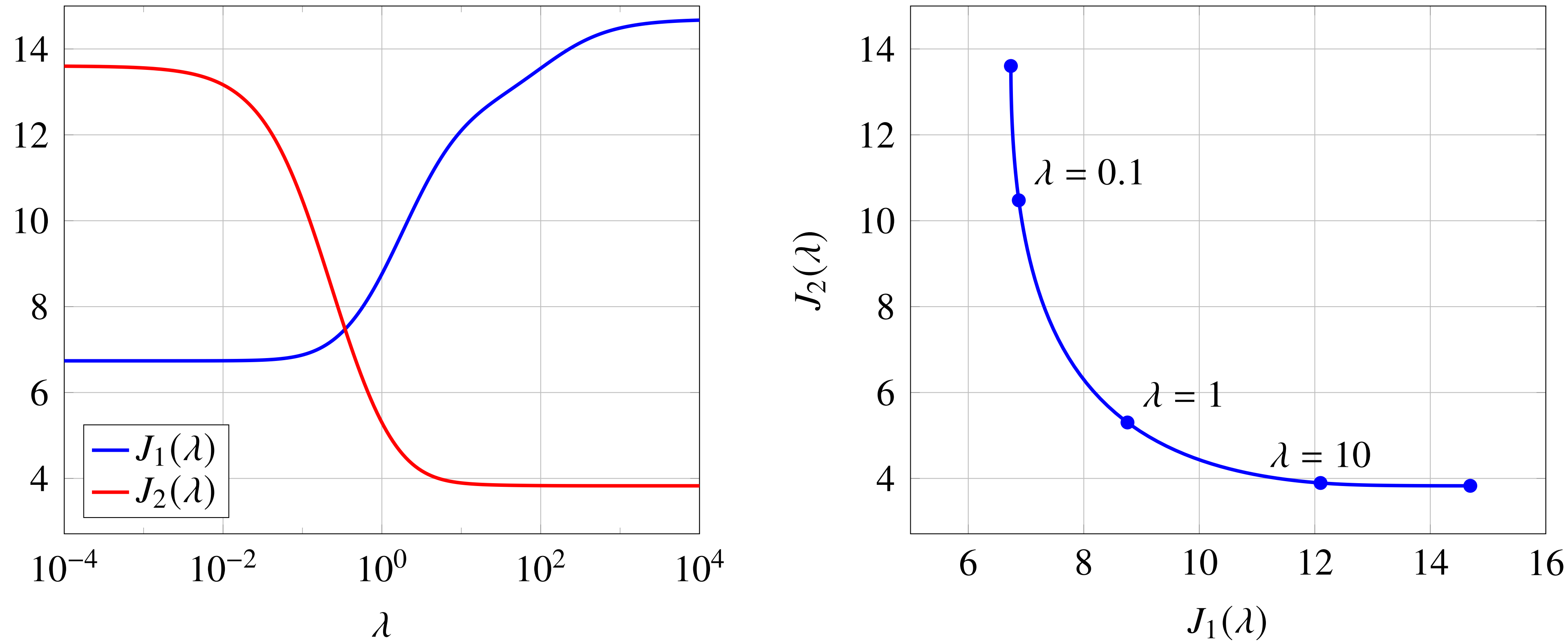
A_1 and A_2 are 10×5



plot shows weighted least squares solution $\hat{x}(\lambda)$ as function of weight λ

Example with two objectives

$$\text{minimize} \quad \|A_1 x - b_1\|^2 + \lambda \|A_2 x - b_2\|^2$$



- left figure shows $J_1(\lambda) = \|A_1\hat{x}(\lambda) - b_1\|^2$ and $J_2(\lambda) = \|A_2\hat{x}(\lambda) - b_2\|^2$
- right figure shows optimal trade-off curve of $J_2(\lambda)$ versus $J_1(\lambda)$

Outline

- multi-objective least squares
- **regularized data fitting**
- control
- estimation and inversion

Motivation

- consider linear-in-parameters model

$$\hat{f}(x) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x)$$

  we assume $f_1(x)$ is the constant function 1

- we fit the model $\hat{f}(x)$ to examples $(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})$
- large coefficient θ_i makes model more sensitive to changes in $f_i(x)$
- keeping $\theta_2, \ldots, \theta_p$ small helps avoid over-fitting
- this leads to two objectives:

$$J_1(\theta) = \sum_{k=1}^{N} (\hat{f}(x^{(k)}) - y^{(k)})^2, \qquad J_2(\theta) = \sum_{j=2}^{p} \theta_j^2$$

  primary objective $J_1(\theta)$ is sum of squares of prediction errors

Weighted least squares formulation

$$\text{minimize} \quad J_1(\theta) + \lambda J_2(\theta) = \sum_{k=1}^{N} (\hat{f}(x^{(k)}) - y^{(k)})^2 + \lambda \sum_{j=2}^{p} \theta_j^2$$

- λ is positive *regularization parameter*
- equivalent to least squares problem: minimize

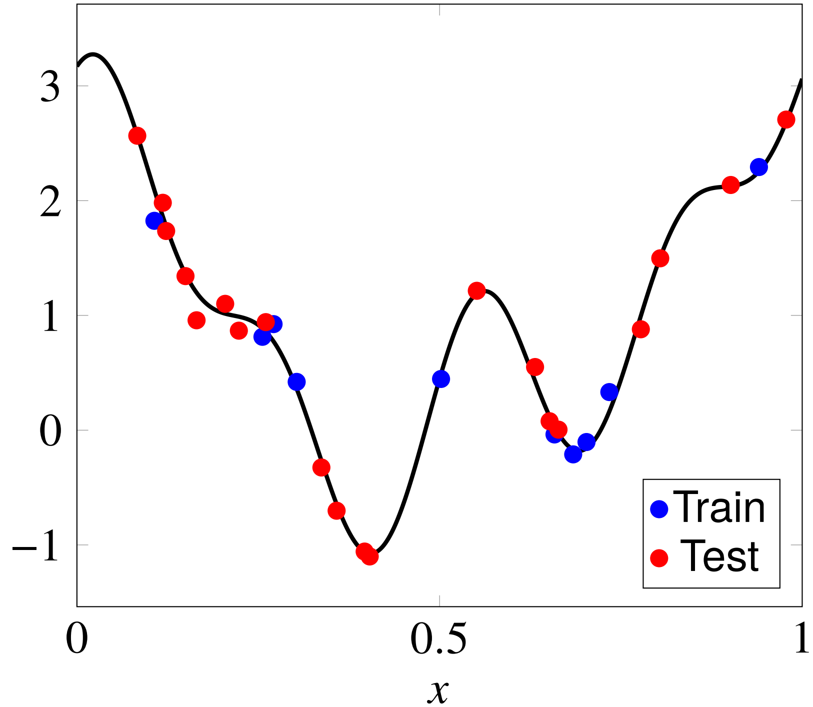
$$\left\| \begin{bmatrix} A_1 \\ \sqrt{\lambda} A_2 \end{bmatrix} \theta - \begin{bmatrix} y^{\mathrm{d}} \\ 0 \end{bmatrix} \right\|^2$$

with $y^{\mathrm{d}} = (y^{(1)}, \ldots, y^{(N)})$,

$$A_1 = \begin{bmatrix} 1 & f_2(x^{(1)}) & \cdots & f_p(x^{(1)}) \\ 1 & f_2(x^{(2)}) & \cdots & f_p(x^{(2)}) \\ \vdots & \vdots & & \vdots \\ 1 & f_2(x^{(N)}) & \cdots & f_p(x^{(N)}) \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

- stacked matrix has linearly independent columns (for positive λ)
- value of λ can be chosen by out-of-sample validation or cross-validation

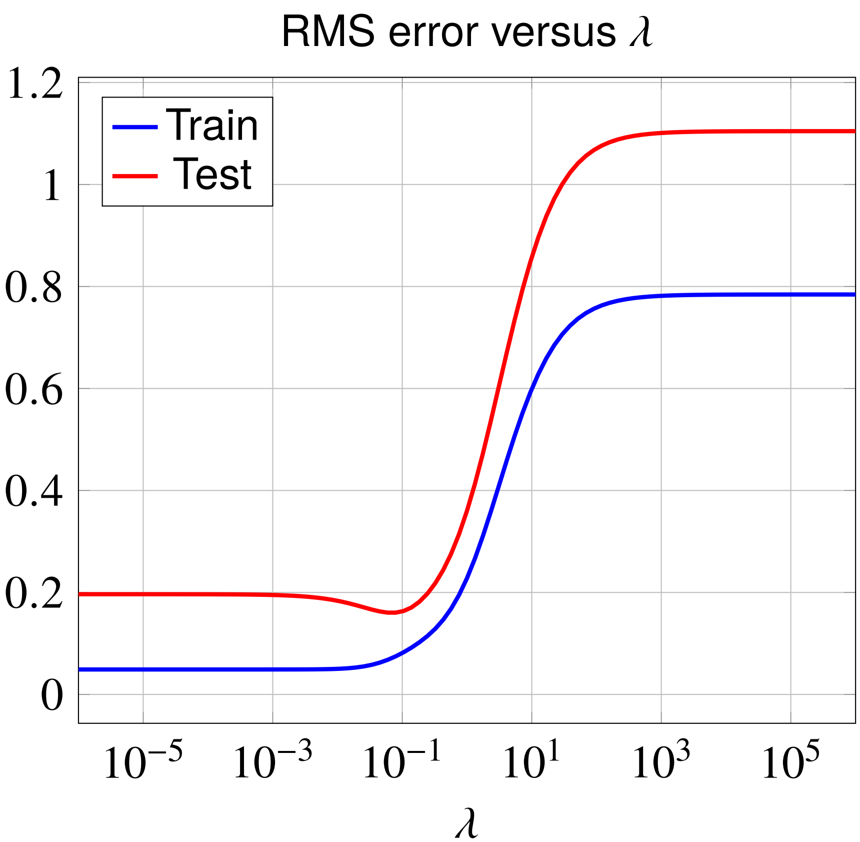
Example

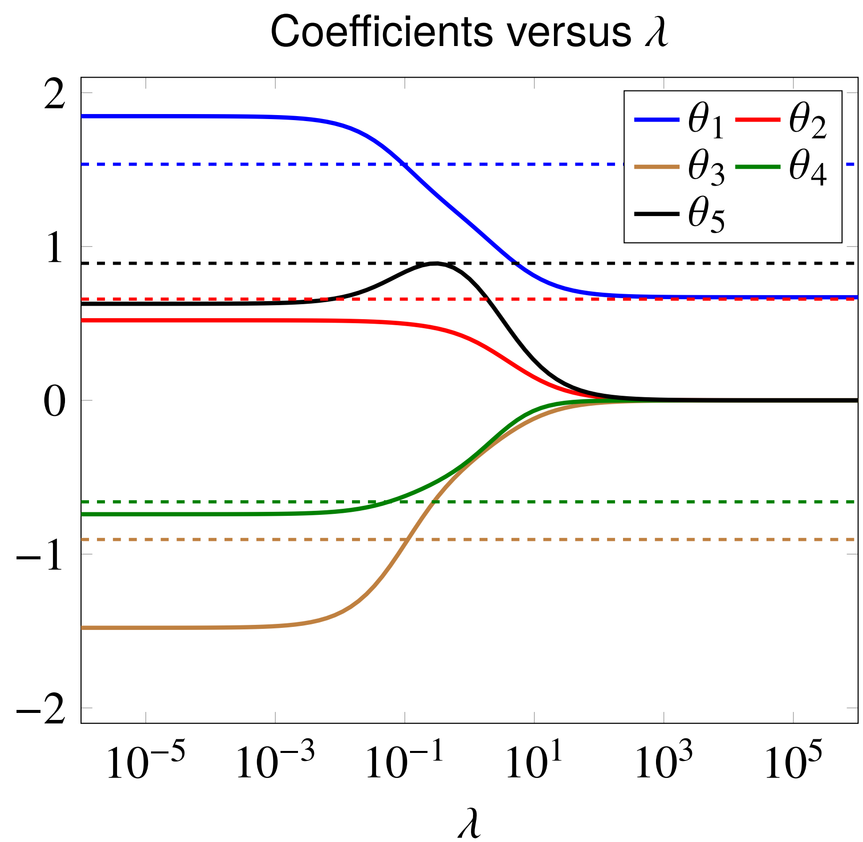


- solid line is signal used to generate synthetic (simulated) data
- 10 blue points are used as training set; 20 red points are used as test set
- we fit a model with five parameters $\theta_1, \ldots, \theta_5$:

$$\hat{f}(x) = \theta_1 + \sum_{k=1}^{4} \theta_{k+1} \sin(\omega_k x + \phi_k) \qquad \text{(with given } \omega_k, \phi_k\text{)}$$

Result of regularized least squares fit





- minimum test RMS error is for λ around 0.08
- increasing λ "shrinks" the coefficients $\theta_2, \ldots, \theta_5$
- dashed lines show coefficients used to generate the data
- for λ near 0.08, estimated coefficients are close to these "true" values

Outline

- multi-objective least squares
- regularized data fitting
- **control**
- estimation and inversion

Control

$$y = Ax + b$$

- x is n-vector of *actions* or *inputs*
- y is m-vector of *results* or *outputs*
- relation between inputs and outputs is a known affine function

the goal is to choose inputs x to optimize different objectives on x and y

Optimal input design

Linear dynamical system

$$y(t) = h_0 u(t) + h_1 u(t-1) + h_2 u(t-2) + \cdots + h_t u(0)$$

- output $y(t)$ and input $u(t)$ are scalar
- we assume input $u(t)$ is zero for $t < 0$
- coefficients $h_0, h_1, \ldots$ are the *impulse response coefficients*
- output is convolution of input with impulse response

Optimal input design

- optimization variable is the input sequence $x = (u(0), u(1), \ldots, u(N))$
- goal is to track a desired output using a small and slowly varying input

Input design objectives

$$\text{minimize} \quad J_{\text{t}}(x) + \lambda_{\text{v}} J_{\text{v}}(x) + \lambda_{\text{m}} J_{\text{m}}(x)$$

- primary objective: track desired output y_{des} over an interval $[0, N]$:

$$J_{\text{t}}(x) = \sum_{t=0}^{N} (y(t) - y_{\text{des}}(t))^2$$

- secondary objectives: use a small and slowly varying input signal:

$$J_{\text{m}}(x) = \sum_{t=0}^{N} u(t)^2, \qquad J_{\text{v}}(x) = \sum_{t=0}^{N-1} (u(t+1) - u(t))^2$$

Tracking error

$$
\begin{aligned}
J_{\mathrm{t}}(x) &= \sum_{t=0}^{N} (y(t) - y_{\mathrm{des}}(t))^2 \\
&= \|A_{\mathrm{t}} x - b_{\mathrm{t}}\|^2
\end{aligned}
$$

with

$$
A_{\mathrm{t}} = \begin{bmatrix}
h_0 & 0 & 0 & \cdots & 0 & 0 \\
h_1 & h_0 & 0 & \cdots & 0 & 0 \\
h_2 & h_1 & h_0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
h_{N-1} & h_{N-2} & h_{N-3} & \cdots & h_0 & 0 \\
h_N & h_{N-1} & h_{N-2} & \cdots & h_1 & h_0
\end{bmatrix}, \qquad
b_{\mathrm{t}} = \begin{bmatrix}
y_{\mathrm{des}}(0) \\
y_{\mathrm{des}}(1) \\
y_{\mathrm{des}}(2) \\
\vdots \\
y_{\mathrm{des}}(N-1) \\
y_{\mathrm{des}}(N)
\end{bmatrix}
$$

Input variation and magnitude

Input variation

$$J_{\mathrm{v}}(x) = \sum_{t=0}^{N-1} (u(t+1) - u(t))^2 = \|Dx\|^2$$

with D the $N \times (N+1)$ matrix

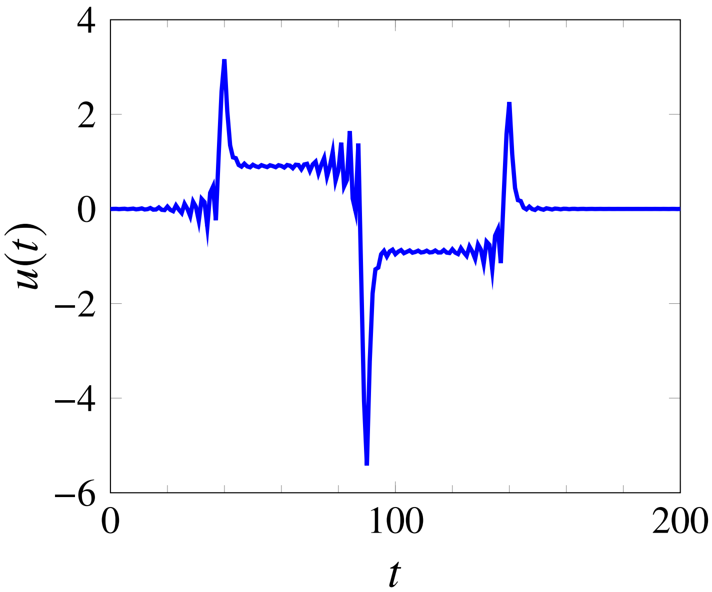
$$D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}$$

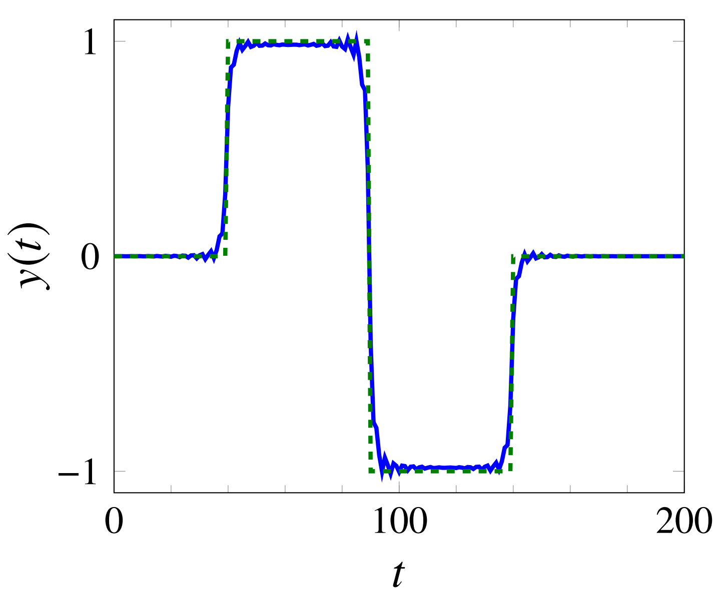
Input magnitude

$$J_{\mathrm{m}}(x) = \sum_{t=0}^{N} u(t)^2 = \|x\|^2$$

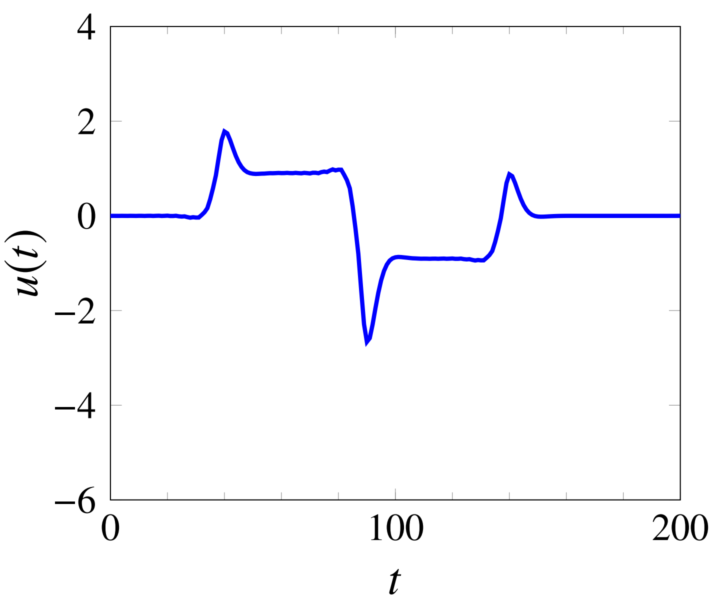
Example

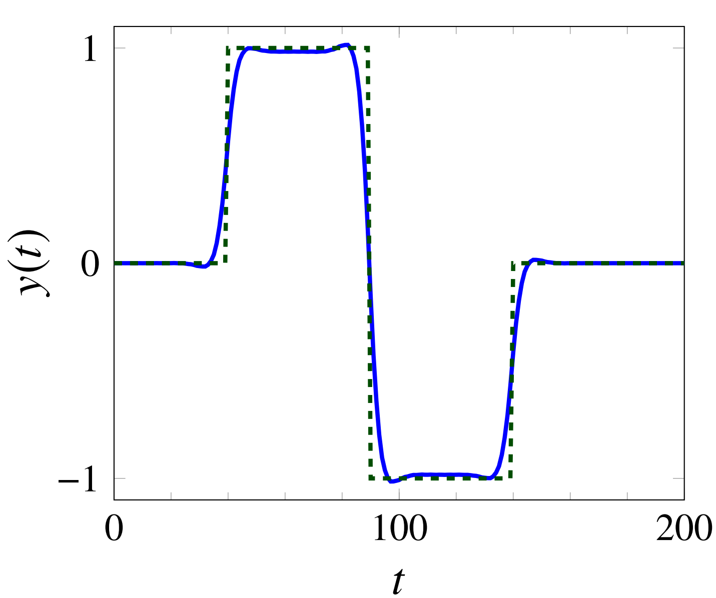
$\lambda_{\mathrm{v}} = 0$,
small λ_{m}





larger λ_{v}
larger λ_{m}





Outline

- multi-objective least squares
- regularized data fitting
- control
- **estimation and inversion**

Estimation

Linear measurement model

$$y = Ax_{\text{ex}} + v$$

- n-vector x_{ex} contains parameters that we want to estimate
- m-vector v is unknown measurement error or noise
- m-vector y contains measurements
- $m \times n$ matrix A relates measurements and parameters

Least squares estimate: use as estimate of x_{ex} the solution $\hat{x}$ of

$$\text{minimize} \quad \|Ax - y\|^2$$

Regularized estimation

add other terms to $\|Ax - y\|^2$ to include information about parameters

Example: Tikhonov regularization

$$\text{minimize} \quad \|Ax - y\|^2 + \lambda \|x\|^2$$

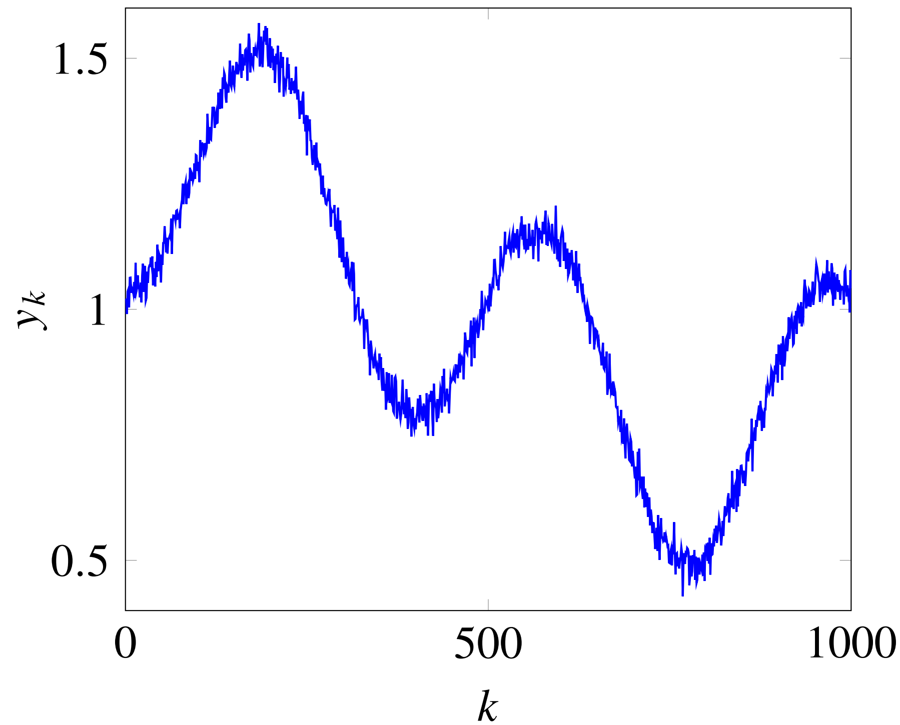
- goal is to make $\|Ax - y\|$ small with small x
- equivalent to solving
$$(A^T A + \lambda I)x = A^T y$$
- solution is unique (if $\lambda > 0$) even when A has linearly dependent columns

Signal denoising

- observed signal y is n-vector

$$y = x_{\text{ex}} + v$$

- x_{ex} is unknown signal
- v is noise



Least squares denoising: find estimate $\hat{x}$ by solving

$$\text{minimize} \quad \|x - y\|^2 + \lambda \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2$$

goal is to find slowly varying signal $\hat{x}$, close to observed signal y

Matrix formulation

$$\text{minimize} \quad \left\| \begin{bmatrix} I \\ \sqrt{\lambda} D \end{bmatrix} x - \begin{bmatrix} y \\ 0 \end{bmatrix} \right\|^2$$

- D is $(n-1) \times n$ finite difference matrix

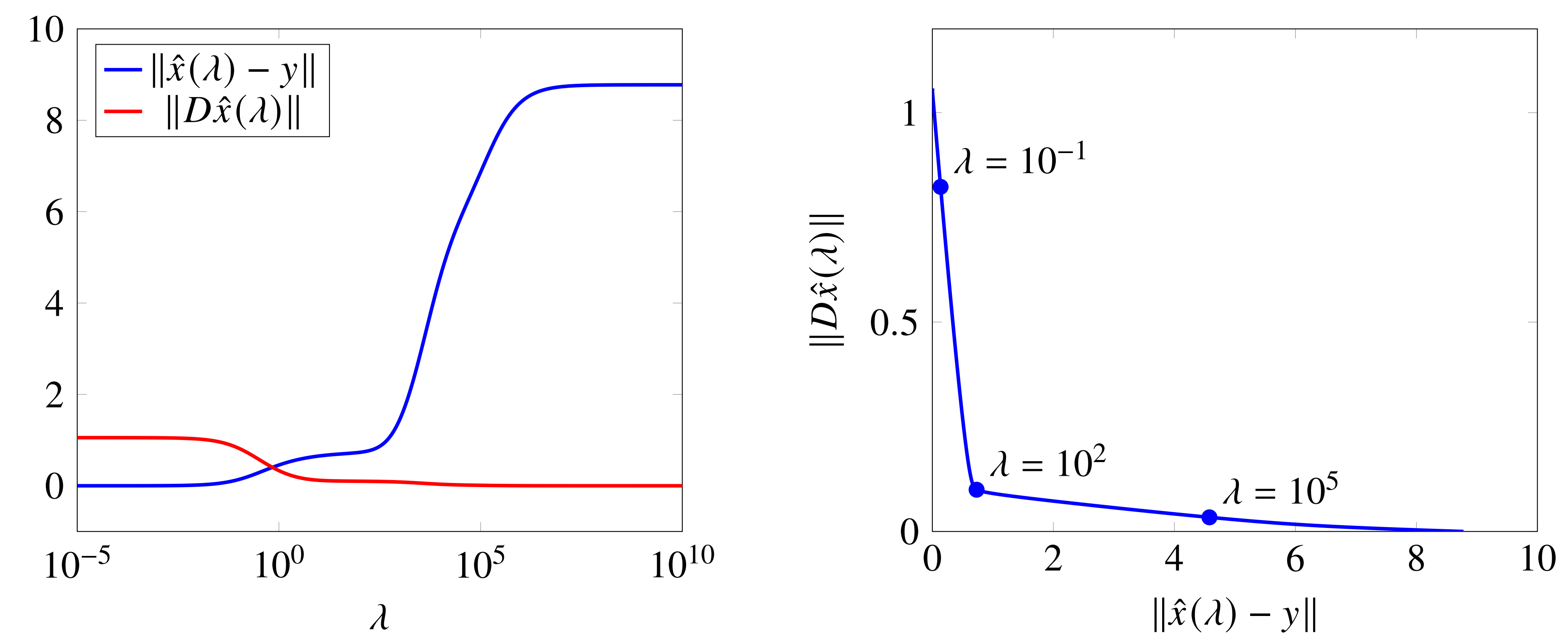
$$D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}$$

- equivalent to linear equation

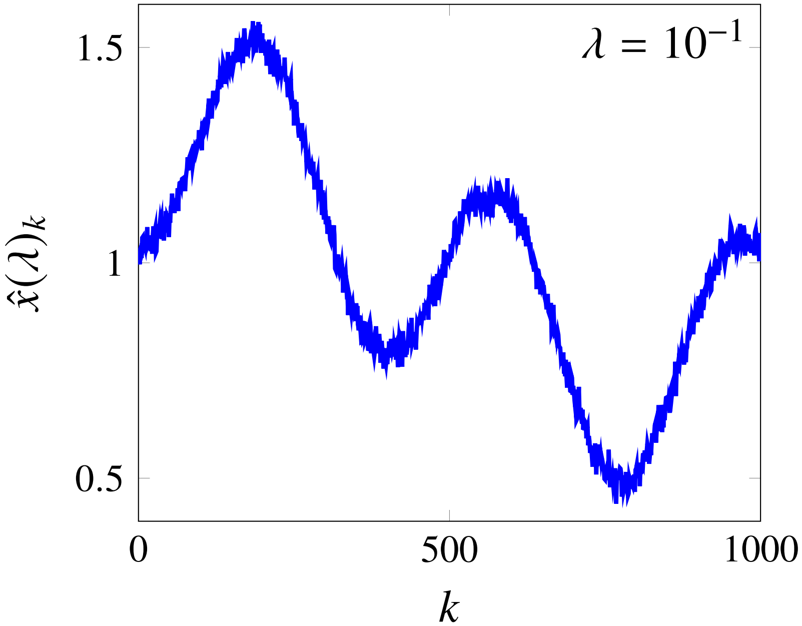
$$(I + \lambda D^T D)x = y$$

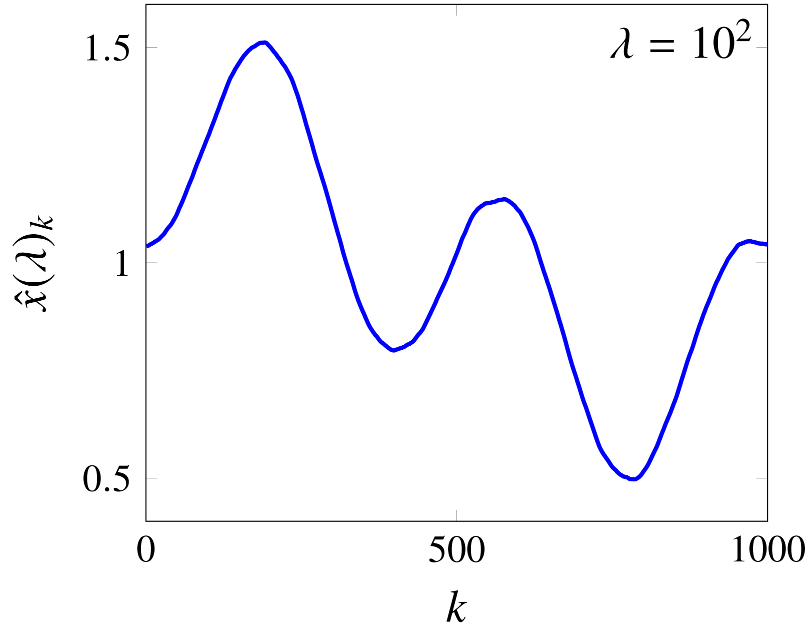
Trade-off

the two objectives $\|\hat{x}(\lambda) - y\|$ and $\|D\hat{x}(\lambda)\|$ for varying λ



Three solutions





- $\hat{x}(\lambda) \to y$ for $\lambda \to 0$
- $\hat{x}(\lambda) \to \mathbf{avg}(y)\mathbf{1}$ for $\lambda \to \infty$
- $\lambda \approx 10^2$ is good compromise

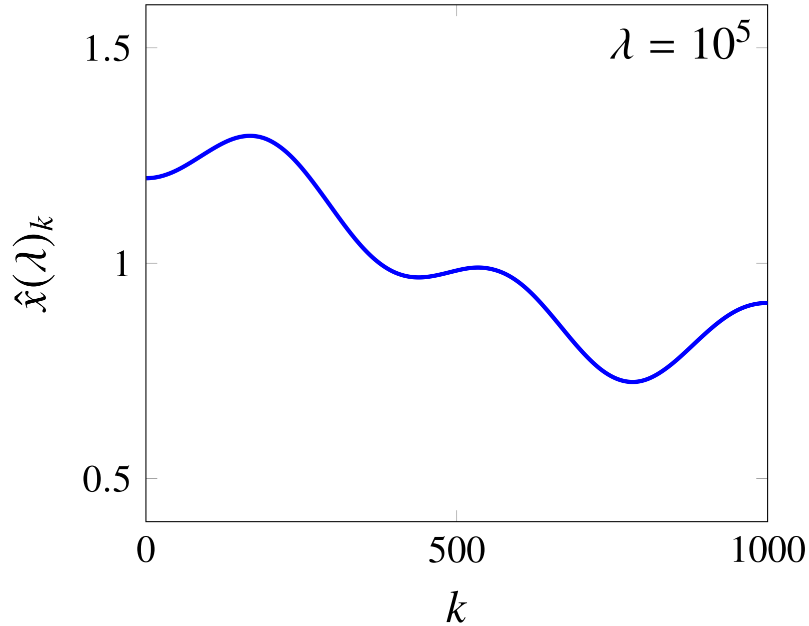


Image deblurring

$$y = Ax_{\text{ex}} + v$$

- x_{ex} is unknown image, y is observed image
- A is (known) blurring matrix, v is (unknown) noise
- images are $M \times N$, stored as MN-vectors

blurred, noisy image y

deblurred image $\hat{x}$

Least squares deblurring

$$\text{minimize} \quad \|Ax - y\|^2 + \lambda(\|D_\text{v}x\|^2 + \|D_\text{h}x\|^2)$$

- 1st term is *"data fidelity"* term: ensures $A\hat{x} \approx y$
- 2nd term penalizes differences between values at neighboring pixels

$$\|D_\text{h}x\|^2 + \|D_\text{v}x\|^2 = \sum_{i=1}^{M} \sum_{j=1}^{N-1} (X_{i,j+1} - X_{ij})^2 + \sum_{i=1}^{M-1} \sum_{j=1}^{N} (X_{i+1,j} - X_{ij})^2$$

if X is the $M \times N$ image stored in the MN-vector x

Differencing operations in matrix notation

suppose x is the $M \times N$ image X, stored column-wise as MN-vector

$$x = \left(X_{1:M,1},\ X_{1:M,2},\ \ldots,\ X_{1:M,N}\right)$$

- horizontal differencing: $(N-1) \times N$ block matrix with $M \times M$ blocks

$$D_{\mathrm{h}} = \begin{bmatrix} -I & I & 0 & \cdots & 0 & 0 & 0 \\ 0 & -I & I & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -I & I \end{bmatrix}$$

- vertical differencing: $N \times N$ block matrix with $(M-1) \times M$ blocks

$$D_{\mathrm{v}} = \begin{bmatrix} D & 0 & \cdots & 0 \\ 0 & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & D \end{bmatrix}, \qquad D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}$$

Deblurred images

$\lambda = 10^{-6}$

$\lambda = 10^{-4}$

$\lambda = 10^{-2}$

$\lambda = 1$

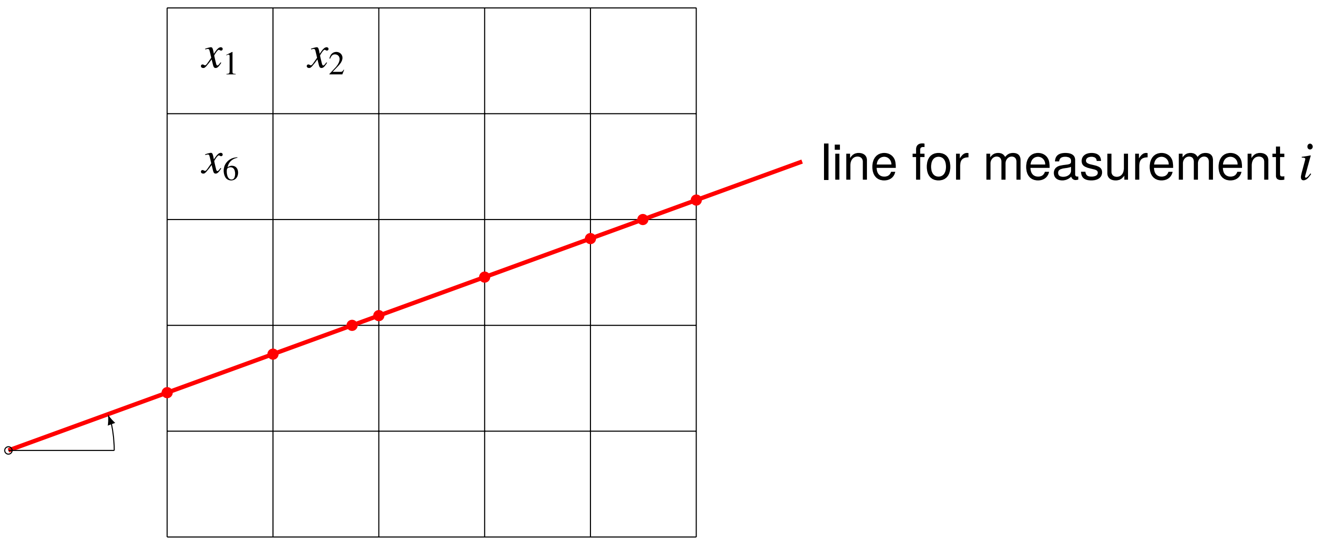
Tomography

$$y = Ax_{\mathrm{ex}} + v$$

- x_{ex} represents values of some quantity in a region of interest of n voxels (pixels)
- Ax represents measurements of the integral along lines through the region

$$(Ax)_i = \sum_{j=1}^{n} A_{ij} x_j$$

A_{ij} is the length of the intersection of the line in measurement i with voxel j

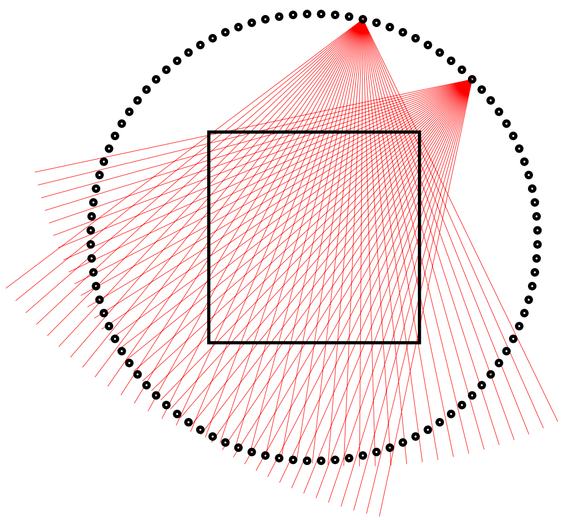


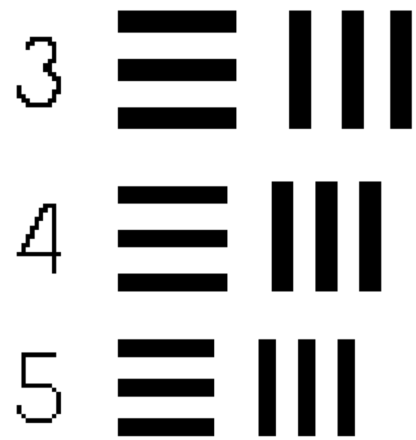
Tomographic reconstruction

$$\text{minimize} \quad \|Ax - y\|^2 + \lambda(\|D_{\mathrm{v}}x\|^2 + \|D_{\mathrm{h}}x\|^2)$$

D_{v} and D_{h} are defined as in image deblurring example on page 10.23

Example

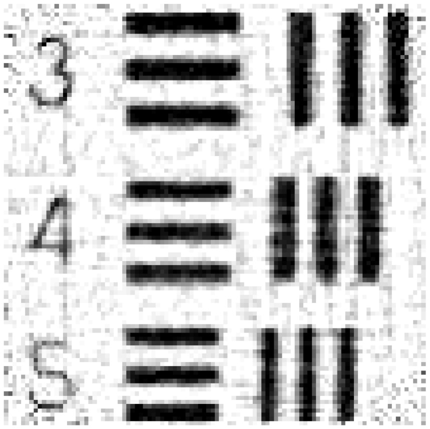




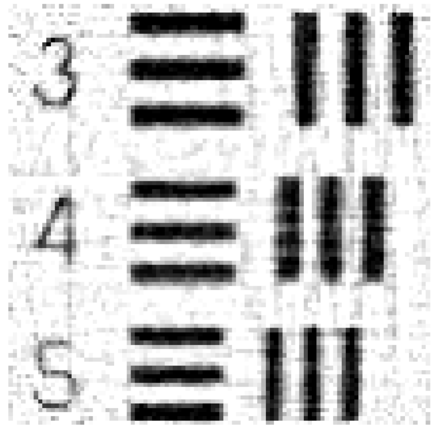
- left: 4000 lines (100 points, 40 lines per point)
- right: object placed in the square region at the center of the picture on the left
- region of interest is divided in 10000 pixels

Regularized least squares reconstruction

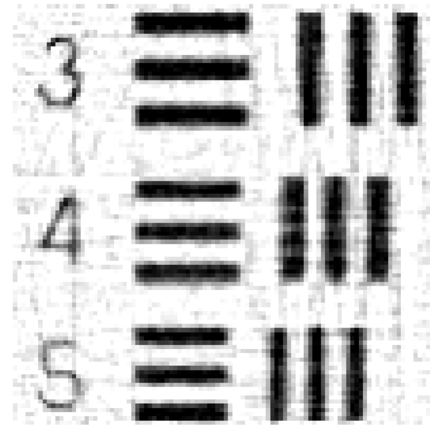
$\lambda = 10^{-2}$



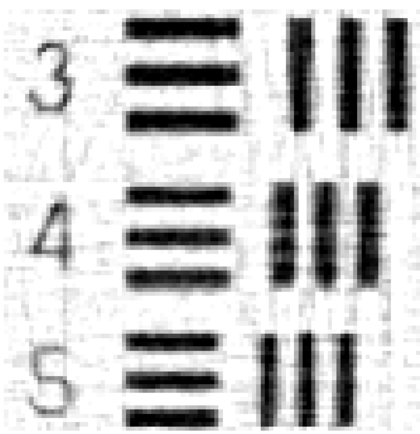
$\lambda = 10^{-1}$



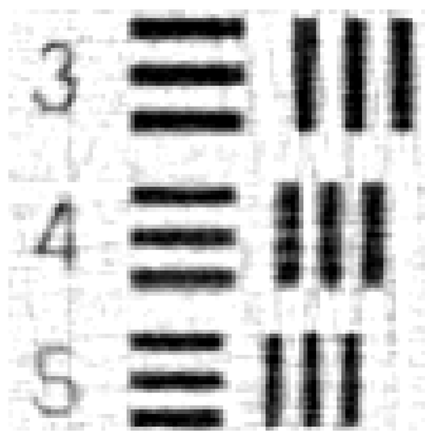
$\lambda = 1$



$\lambda = 5$



$\lambda = 10$



$\lambda = 100$