Programming, numerics and optimization

Lecture B-4: Linear systems II

Łukasz Jankowski
ljank@ippt.pan.pl

Institute of Fundamental Technological Research
Room 4.32, Phone +22.8261281 ext. 428

April 20, 2021[1]

[1] Current version is available at http://info.ippt.pan.pl/~ljank.

Outline

Outline

Least-squares problem

Least-squares problems originate from estimation of model parameter in problems of fitting noisy data.

Consider the following model

$$y(t) = f(t; \mathbf{x}),$$

where $\mathbf{x} \in \mathbb{R}^n$ is a vector of unknown model parameters, to be determined from a series of (noisy) measurements (y_i, t_i), $i = 1, 2, \ldots, m$. To reduce the noise influence, $m > n$.

The related least-squares problem is in general a nonlinear optimization problem:

$$\text{find } \mathbf{x}, \text{ which minimizes } \sum_{i=1}^{m} (y_i - f(t_i; \mathbf{x}))^2.$$

Linear least-squares problem

If the dependence on parameters $\mathbf{x}$ is linear,

$$y(t) = \sum_{j=1}^{n} x_j f_j(t),$$

a linear least-squares problem is obtained,

$$\text{find } \mathbf{x}, \text{ which minimizes } \sum_{i=1}^{m} \left[y_i - \sum_{j=1}^{n} x_j f_j(t_i) \right]^2,$$

which is equivalent to the following convex optimization problem:

$$\text{find } \mathbf{x}, \text{ which minimizes } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2,$$

where $\mathbf{x} = [x_1, \ldots, x_n]^\mathsf{T}$, $\mathbf{y} = [y_1, \ldots, y_m]^\mathsf{T}$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$ is an $m \times n$ matrix composed of $f_j(t_i)$.

Linear least-squares problem

Linear least-squares problem

$$\text{find } \mathbf{x}, \text{ which minimizes } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2,$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$.

The related linear equation $\mathbf{y} = \mathbf{A}\mathbf{x}$ can be in general inconsistent ($m > n$) and have no exact solutions. However, the least-squares solution $\mathbf{x}_{\text{LS}}$ is the best solution in the sense of the *minimum residual norm*:

- $\mathbf{A}\mathbf{x}_{\text{LS}}$ is the closest point to $\mathbf{y}$ out of all points from range $\mathbf{A}$.
- $\mathbf{A}\mathbf{x}_{\text{LS}}$ is the orthogonal projection of $\mathbf{y}$ onto range $\mathbf{A}$.

The least-squares solution is unique, if rank $\mathbf{A} = n$ (full column rank), that is if $\mathbf{A}$ corresponds to an injection.

Normal equations

Normal equations

The vector $\mathbf{x}$ minimizes $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2$ if and only if it satisfies the *normal equations*,

$$\mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{x} = \mathbf{A}^\mathsf{T}\mathbf{y},$$

or equivalently $\mathbf{A}^\mathsf{T}(\mathbf{y} - \mathbf{A}\mathbf{x}) = \mathbf{0}$.

- If $m >> n$, the matrix $\mathbf{A}^\mathsf{T}\mathbf{A} \in \mathbb{R}^{n\times n}$ is much smaller than the initial matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$.
- In the general case, $\mathbf{A}^\mathsf{T}\mathbf{A}$ is (square) symmetric positive semidefinite.
- If $\mathbf{A}$ is full column rank, then $\mathbf{A}^\mathsf{T}\mathbf{A}$ is full-rank positive definite and normal equations have a unique solution.
- A disadvantage of the normal equations is squaring of the condition number (if $\mathbf{A}$ is ill-conditioned, then conditioning $\mathbf{A}^\mathsf{T}\mathbf{A}$ is even worse).

Gauss-Markov theorem

The Gauss-Markov theorem

- recasts the problem of estimating a vector of parameters in a (noisy) linear model using the statistical terminology, and
- states that the least-squares solution is the optimum estimator of the parameter vector in the statistical sense, that it is
  1. unbiased (mean estimation error is zero) and
  2. of minimum variance.

Gauss-Markov theorem

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be a full column rank matrix of real numbers, $\mathbf{y} \in \mathbb{R}^m$ be a vector of (noisy) measurements (or observations) and $\mathbf{x} \in \mathbb{R}^n$ be an unknown parameter vector. Assume the linear model

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon}, \qquad (\star)$$

where $\boldsymbol{\epsilon}$ is a vector of uncorrelated random measurement errors with zero mean and the same variance,

$$\mathrm{E}[\boldsymbol{\epsilon}] = \mathbf{0}, \qquad \mathrm{V}[\boldsymbol{\epsilon}] = \sigma^2 \mathbf{I}.$$

The best unbiased estimator $\hat{\mathbf{x}}$ of $\mathbf{x}$ is the solution to the least-squares problem associated with $(\star)$:

$$\hat{\mathbf{x}} = (\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\mathbf{y}, \qquad \mathrm{V}[\hat{\mathbf{x}}] = \sigma^2(\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}.$$

Singular value decomposition (SVD)

Every rectangular matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ can be expressed as

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{T}},$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices, and $\mathbf{\Sigma}$ is an $m \times n$ diagonal matrix with rank $\mathbf{A}$ nonnegative diagonal elements σ_i (called singular values) ordered in a nonincreasing way.

Given matrix $\mathbf{A}$, its rank can be marked explicitly in the SVD:

$$\mathbf{A} = [\mathbf{U}_1 \mathbf{U}_2] \begin{bmatrix} \mathbf{\Sigma}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^{\mathsf{T}} \\ \mathbf{V}_2^{\mathsf{T}} \end{bmatrix} = \mathbf{U}_1 \mathbf{\Sigma}_0 \mathbf{V}_1^{\mathsf{T}},$$

where the number of columns of $\mathbf{U}_1$, $\mathbf{V}_1$ and $\mathbf{\Sigma}_0$ is rank $\mathbf{A}$. The reduced decomposition $\mathbf{A} = \mathbf{U}_1 \mathbf{\Sigma}_0 \mathbf{V}_1^{\mathsf{T}}$ is called the thin SVD of $\mathbf{A}$.

SVD-based least-squares solution

Let $\mathbf{A} = \mathbf{U}_1 \mathbf{\Sigma}_0 \mathbf{V}_1^{\mathsf{T}}$ be the thin SVD of $\mathbf{A} \in \mathbb{R}^{m \times n}$. The least-squares problem

$$\text{find } \mathbf{x}, \text{ which minimizes } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2,$$

has always a unique minimum norm solution[a] given by

$$\mathbf{x}_{\text{svd}} = \mathbf{V}_1 \mathbf{\Sigma}_0^{-1} \mathbf{U}_1^{\mathsf{T}} \mathbf{y}.$$

[a] That is, if solution to the least-squares problem is nonunique ($\operatorname{rank} \mathbf{A} < n$), the solution with the minimum norm is chosen.

The matrix

$$\mathbf{A}^{+} = \mathbf{V}_1 \mathbf{\Sigma}_0^{-1} \mathbf{U}_1^{\mathsf{T}}$$

is called the pseudo-inverse or the Moore-Penrose inverse of $\mathbf{A}$.

SVD-based least-squares solution

- The SVD expresses the matrix $\mathbf{A}$ as a sum of rank-one matrices:

$$\mathbf{A} = \mathbf{U}_1 \mathbf{\Sigma}_0 \mathbf{V}_1^{\mathsf{T}} = \sum_{i=1}^{\operatorname{rank} \mathbf{A}} \sigma_i \, \mathbf{u}_i \mathbf{v}_i^{\mathsf{T}},$$

  where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the ith columns of $\mathbf{U}$ and $\mathbf{V}$, respectively.

- The minimum norm minimizer of $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2$ can be thus expressed as

$$\mathbf{x}_{\text{svd}} = \mathbf{V}_1 \mathbf{\Sigma}_0^{-1} \mathbf{U}_1^{\mathsf{T}} \mathbf{y} = \left[\sum_{i=1}^{\operatorname{rank} \mathbf{A}} \frac{\mathbf{v}_i \mathbf{u}_i^{\mathsf{T}}}{\sigma_i} \right] \mathbf{y}.$$

SVD-based approach vs. normal equations

In comparison to the normal equations

+ In the SVD-based approach, the initial matrix $\mathbf{A}$ is used directly. Therefore, the condition number is not squared.
– If $m >> n$, the matrix $\mathbf{A}$ is much larger than $\mathbf{A}^{\mathsf{T}}\mathbf{A}$ and the numerical cost of computing the SVD can be prohibitive.

Iterative methods

The linear least-squares problem, expressed as

$$\text{find } \mathbf{x}, \text{ which minimizes } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2,$$

is in fact the problem of unconstrained minimization of a convex quadratic objective function

$$\phi(\mathbf{x}) = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 = \mathbf{x}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{A}\mathbf{x} - 2\,\mathbf{y}^{\mathsf{T}}\mathbf{A}\mathbf{x} + \mathbf{y}^{\mathsf{T}}\mathbf{y},$$

where $\mathbf{A}^{\mathsf{T}}\mathbf{A}$ is positive semidefinite.

Therefore, any iterative optimization scheme can be used to find an approximation to the exact solution, in particular the conjugate gradient least-squares (CGLS) method described in Lecture B-3.

Outline

Norm on a linear space

A norm $\|\cdot\|$ on a linear space $\mathcal{X}$

Let $\mathcal{X}$ be a real linear space. A function $\|\cdot\| : \mathcal{X} \to \mathbb{R}$ with the properties

$$\begin{aligned} \|\mathbf{x}\| &\geq 0 && \text{(positivity)} \\ \|\mathbf{x}\| &= 0 \text{ iff } \mathbf{x} = \mathbf{0} && \text{(definitness)} \\ \|k\mathbf{x}\| &= |k|\|\mathbf{x}\| && \text{(homogeneity)} \\ \|\mathbf{x}+\mathbf{y}\| &\leq \|\mathbf{x}\| + \|\mathbf{y}\| && \text{(triangle inequality)} \end{aligned}$$

for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ and all $k \in \mathbb{R}$ is called a norm on $\mathcal{X}$.

Vector norms

Vector p-norms

The most often used vector norms are the so-called p-norms $\|\cdot\|_p$:

$$\|\mathbf{x}\|_p = \left[\sum_i |x_i^p|\right]^{\frac{1}{p}}.$$

In particular,

$$\|\mathbf{x}\|_1 = \sum_i |x_i|,$$

$$\|\mathbf{x}\| = \|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2},$$

$$\|\mathbf{x}\|_\infty = \max_i |x_i|.$$

Matrix norms

Operator norms

A vector norm $\|\cdot\|$ induces the corresponding matrix operator norm

$$\|\mathbf{A}\| = \max_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}.$$

In particular, the vector p-norms induce the matrix norms:

$$\begin{aligned}
\|\mathbf{A}\|_1 &= \max_j \sum_i |a_{ij}|, \\
\|\mathbf{A}\|_2 &= \sqrt{\max_i \lambda_i(\mathbf{A}^\mathsf{T}\mathbf{A})} = \max_i \sigma_i(\mathbf{A}), \\
\|\mathbf{A}\|_\infty &= \max_i \sum_j |a_{ij}|.
\end{aligned}$$

The Frobenius matrix norm is defined by

$$\|\mathbf{A}\|_\mathsf{F} = \sqrt{\sum_{i,j} a_{ij}^2} = \sqrt{\sum_i \lambda_i^2}.$$

Consistent matrix and vector norms

It is important to use the matrix norms that are *consistent* with the used vector norms, that is

$$\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|\|\mathbf{x}\|.$$

The following norms are consistent:

$$\|\mathbf{x}\|_1 \ldots \|\mathbf{A}\|_1$$
$$\|\mathbf{x}\|_2 \ldots \|\mathbf{A}\|_2, \|\mathbf{A}\|_F$$
$$\|\mathbf{x}\|_\infty \ldots \|\mathbf{A}\|_\infty$$

Condition number

The notion of the condition number of a problem has been introduced in Lecture B-1:

- The condition number is the amplification factor of the relative error between input and output data. It measures how much the errors in the input data can affect the errors in the output data.
- Conditioning is a property of the problem, which (independent of any algorithm used to solve it) can be well-conditioned or ill-conditioned.
- If a problem is ill-conditioned, an algorithm can give better results only by chance. The errors in input data will inevitably propagate to the output data.

Condition number — diagonal matrices

Let $\mathbf{D} = \operatorname{diag}(d_1, d_2, \ldots, d_n)$ be a nonsingular diagonal matrix. The linear system $\mathbf{Dx} = \mathbf{y}$ is a system of simple decoupled equations:

$$d_i x_i = y_i, \qquad i = 1, 2, \ldots, n,$$

which has the exact solution

$$x_i = \frac{y_i}{d_i}, \qquad i = 1, 2, \ldots, n.$$

In practice, the right-hand side is known only approximately (due to measurement errors, floating-point representation errors, etc.), so that the system equation takes the form

$$d_i(x_i + \Delta x_i) = y_i + \Delta y_i, \qquad i = 1, 2, \ldots, n,$$

where $\frac{|\Delta y_i|}{\|\mathbf{y}\|} \approx \epsilon > 0$. Notice that $d_i \Delta x_i = \Delta y_i$.

Condition number — diagonal matrices

Since $d_i \Delta x_i = \Delta y_i$, the absolute error of the solution is

$$\Delta x_i = \frac{\Delta y_i}{d_i}, \qquad i = 1, 2, \ldots, n.$$

Hence, the relative error level of the solution is bounded as follows

$$\frac{|\Delta x_i|}{\|\mathbf{x}\|} = \frac{1}{d_i} \frac{|\Delta y_i|}{\|\mathbf{x}\|} = \frac{1}{d_i} \frac{\|\mathbf{y}\|}{\|\mathbf{x}\|} \frac{|\Delta y_i|}{\|\mathbf{y}\|} \leq \frac{\max_j |d_j|}{d_i} \frac{|\Delta y_i|}{\|\mathbf{y}\|} \leq \frac{\max_j |d_j|}{\min_j |d_j|} \frac{|\Delta y_i|}{\|\mathbf{y}\|}.$$

The condition number of the linear system $\mathbf{D}\mathbf{x} = \mathbf{y}$ is thus

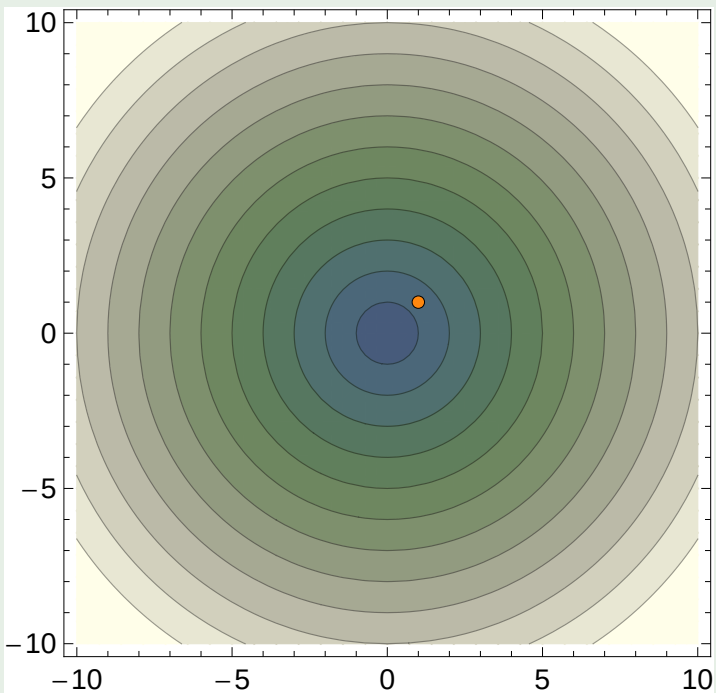
$$\frac{|\Delta x_i|}{\|\mathbf{x}\|} \Big/ \frac{|\Delta y_i|}{\|\mathbf{y}\|} \leq \frac{\max_j |d_j|}{\min_j |d_j|} = \kappa(\mathbf{D})$$

and can be very large, if d_i are of very different magnitudes.

Condition number — diagonal matrices — 2D example

$$\mathbf{D} = \begin{bmatrix} 1 & 0 \\ 0 & 0.2 \end{bmatrix}, \quad \kappa(\mathbf{D}) = 5, \quad \mathbf{y} = \begin{bmatrix} 1 \\ 0.2 \end{bmatrix}, \quad \|\Delta \mathbf{y}\| = 0$$

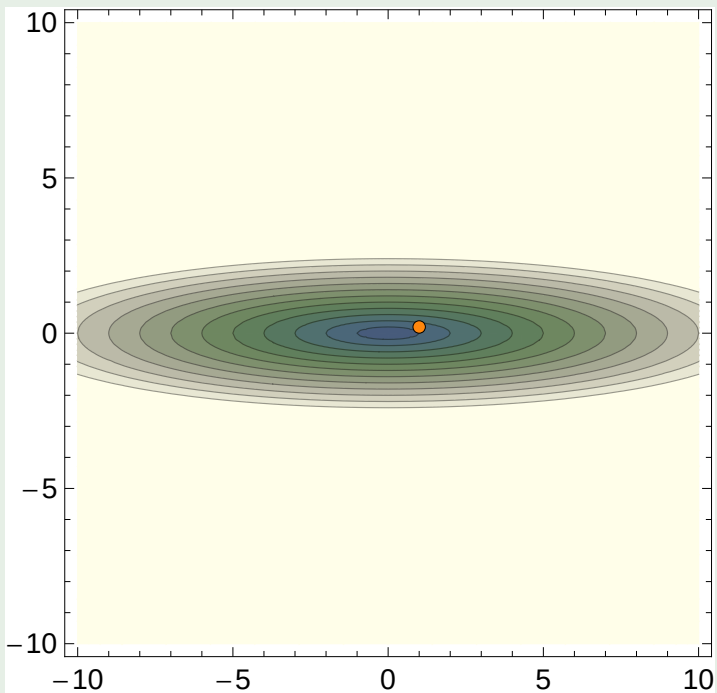
space of $\mathbf{x}$



$\mathbf{D}$ $\longrightarrow$

$\longleftarrow$ $\mathbf{D}^{-1}$

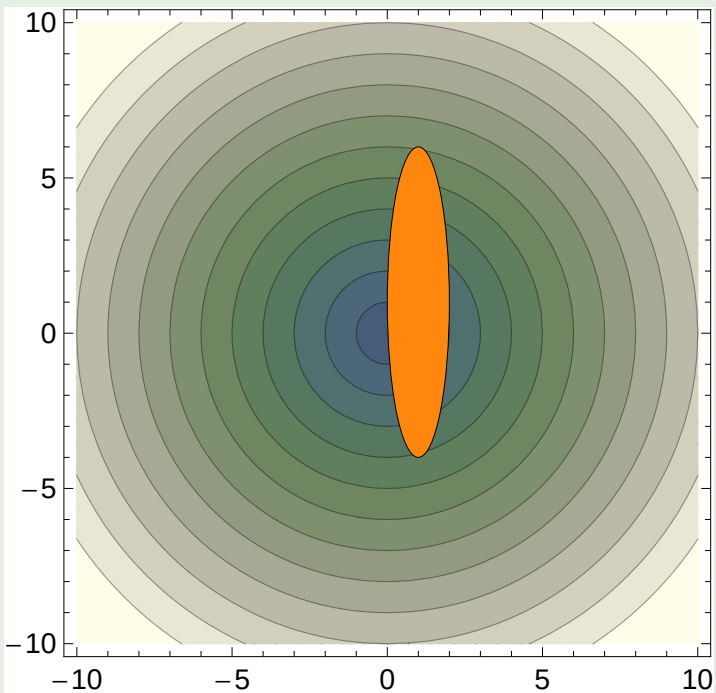
space of $\mathbf{y}$



Condition number — diagonal matrices — 2D example

$$\mathbf{D} = \begin{bmatrix} 1 & 0 \\ 0 & 0.2 \end{bmatrix}, \quad \kappa(\mathbf{D}) = 5, \quad \mathbf{y} = \begin{bmatrix} 1 \\ 0.2 \end{bmatrix}, \quad \|\Delta \mathbf{y}\| \leq 1$$

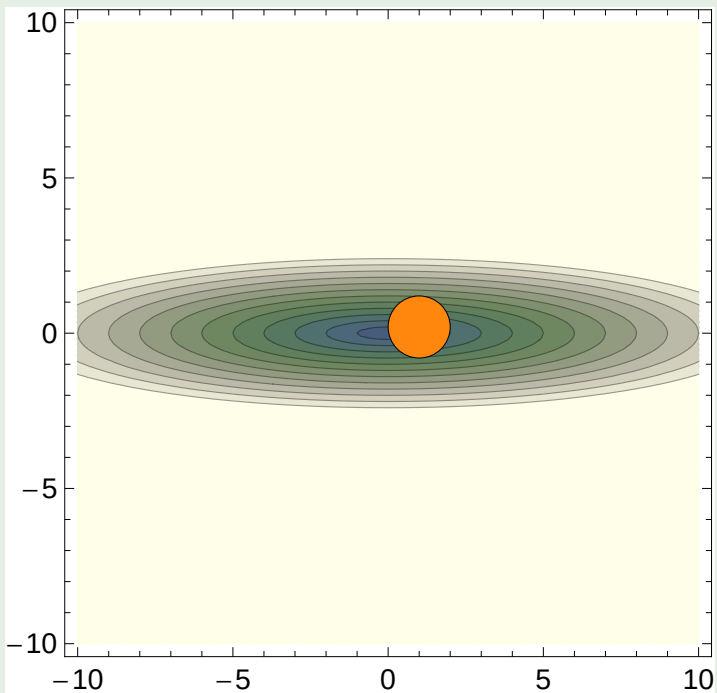
space of $\mathbf{x}$



$\mathbf{D}$ $\longrightarrow$

$\longleftarrow$ $\mathbf{D}^{-1}$

space of $\mathbf{y}$



Condition number

The diagonal matrix is an exemplary case, since all matrices can be transformed to diagonal matrices using the SVD. The linear system

$$\mathbf{A}\mathbf{x} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{T}}\mathbf{x} = \mathbf{y}$$

by the following transformation of variables[2]

$$\hat{\mathbf{x}} = \mathbf{V}^{\mathsf{T}}\mathbf{x}, \qquad \hat{\mathbf{y}} = \mathbf{U}^{\mathsf{T}}\mathbf{y}$$

is reduced to the diagonal system

$$\mathbf{\Sigma}\hat{\mathbf{x}} = \hat{\mathbf{y}}.$$

The condition number of both systems is thus

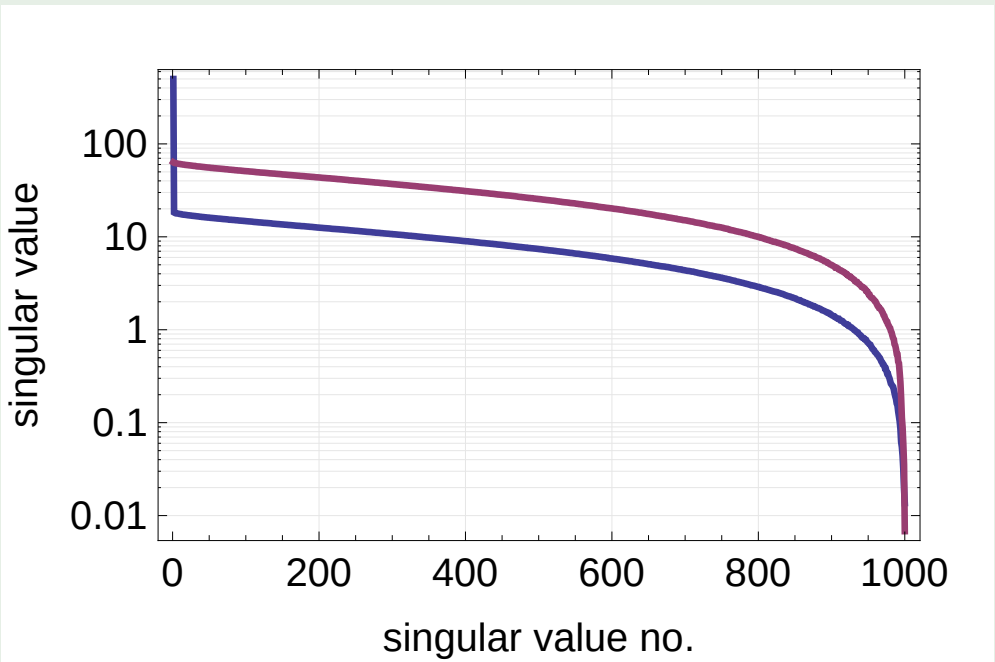
$$\kappa(\mathbf{A}) = \kappa(\mathbf{\Sigma}) = \frac{\sigma_{\max}}{\sigma_{\min}}.$$

[2] Transformations by unitary matrices have always the best-possible condition number of 1.

Condition number

Although it is easy to construct a counterexample, highly ill-conditioned matrices of small dimensions are rather rare in practice. However, ill-conditioning of medium and large matrices is much more common.

Logarithmic plot of the singular values of two 1000×1000 matrices of random numbers N(0,1) and U(0,1):



Deconvolution as a common source of ill-conditioning

In proper infinite-dimensional function spaces, the convolution with a continuous function $h(t)$,

$$y(t) = \int_0^T h(t-s)x(s)\,\mathrm{d}s, \qquad t \in [0, T],$$

can be stated in the form of a simple linear operator equation $y = \mathcal{H}x$. The operator $\mathcal{H}$ is compact[3], so its inverse $\mathcal{H}^{-1}$ is unbounded and noncontinuous. Thus, $\kappa(\mathcal{H}) = \infty$ and the problem of finding $x(t)$, given $y(t)$ and $h(t)$, is extremely ill-conditioned (ill-posed).

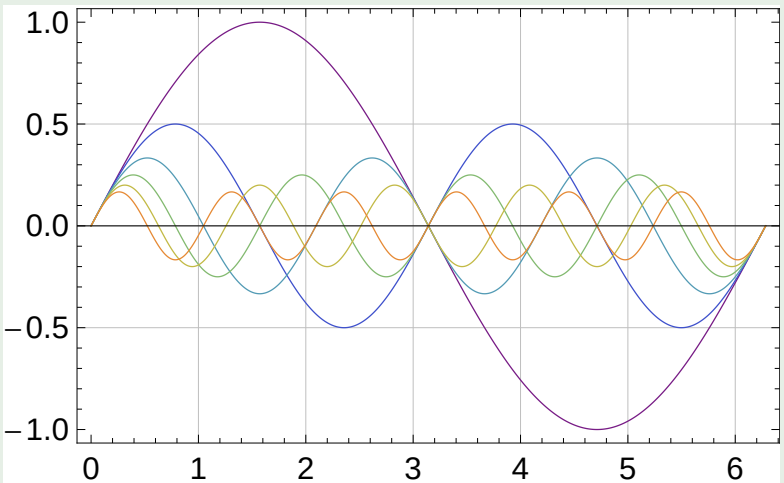
[3] Intuitively, compact $\approx$ smoothing.

Deconvolution as a common source of ill-conditioning

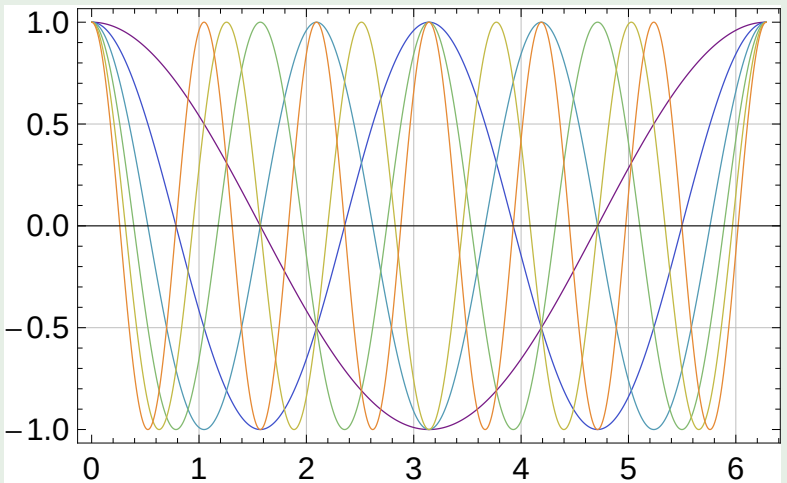
$$(\mathcal{H}x)(t) = \int_0^t x(t)\,dt \qquad (\mathcal{H}^{-1}y)(t) = y'(t)$$

Both $\mathcal{H}$ and its inverse $\mathcal{H}^{-1}$ are linear. However, the inverse is everywhere non-continuous.

$$y_n(t) = \frac{\sin\, nt}{n} \qquad (\mathcal{H}^{-1}y_n)(t) = \cos\, nt$$



uniform convergence to 0



non-convergent

Deconvolution as a common source of ill-conditioning

In practice, all computations are performed using finite-dimensional data, so that discretized h and y are used to compute the discretized x. If the data originate from measurements or simulations, the considered time interval $[0, T]$ is usually sampled into N equal time steps $t_i = i\Delta t$, so that

$$y_i = \sum_{j=0}^{N-1} h_{i-j}\, x_j, \qquad i = 0, \ldots, N-1,$$

where $y_i = y(t_i)$, $x_j = x(t_j)$ and $h_k = \Delta t h(t_k)$.

Deconvolution as a common source of ill-conditioning

The discretized equations can be stated in the matrix form as a single large finite-dimensional linear system

$$\mathbf{y} = \mathbf{H}\mathbf{x},$$

where

$$\mathbf{y} = [y_1, \ldots, y_N]^{\mathsf{T}}$$
$$\mathbf{x} = [x_1, \ldots, x_N]^{\mathsf{T}}$$

and $\mathbf{H} = [h_{ij}]$ is a diagonal-constant matrix, that is

$$h_{ij} = h_{i-j}.$$

Teoplitz matrix

A diagonal-constant matrix is a matrix $\mathbf{H} = [h_{ij}]$, which satisfies $h_{ij} = h_{i-j}$. Such matrices are also called Toeplitz matrices and are examples of structured matrices.

Deconvolution as a common source of ill-conditioning

Discretized version of the deconvolution problem:

$$\mathbf{y} = \mathbf{H}\mathbf{x},$$

where $\mathbf{H} = [h_{ij}]$ is a Toeplitz matrix, that is $h_{ij} = h_{i-j}$.

1. Since the original continuous system was extremely ill-conditioned (unbounded inverse, hence $\kappa(\mathcal{H}) = \infty$ and the inverse problem is even ill-posed), the condition number $\kappa(\mathbf{H})$ grows to ∞ as the discretization time step Δt tends to zero.
2. An important conclusion: finer mesh (discretization) need not lead to a better accuracy.
3. The singular values of a Toeplitz matrix that "arises from the discretization of first-kind Fredholm integral equations [...] decay gradually, until they level off at a plateau approximately at the machine precision times $\sigma_{\max}$ (in infinite precision they would decay to zero)" (P. Ch. Hansen).

Deconvolution — example

Consider the following deconvolution problem:

$$y(t) = \int_0^t \cos(t-s)x(s)ds,$$

where $y(t)$ can be interpreted as the velocity response of an undamped single degree of freedom system to a force excitation described by $x(t)$.

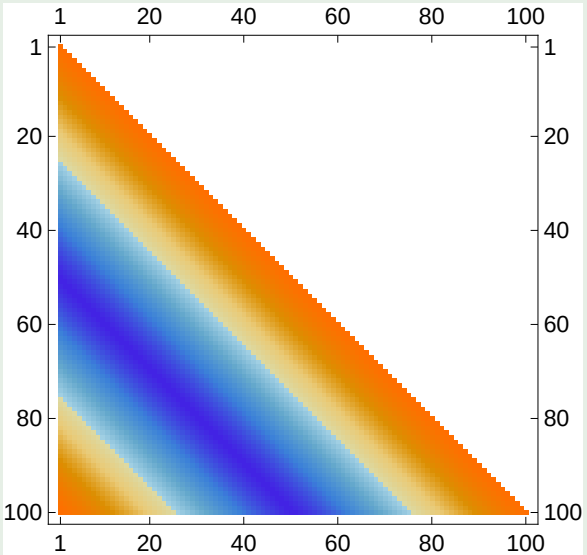
Assume the observed response

$$y(t) = \frac{1}{2}t\sin t, \qquad t \in [0, 2\pi].$$

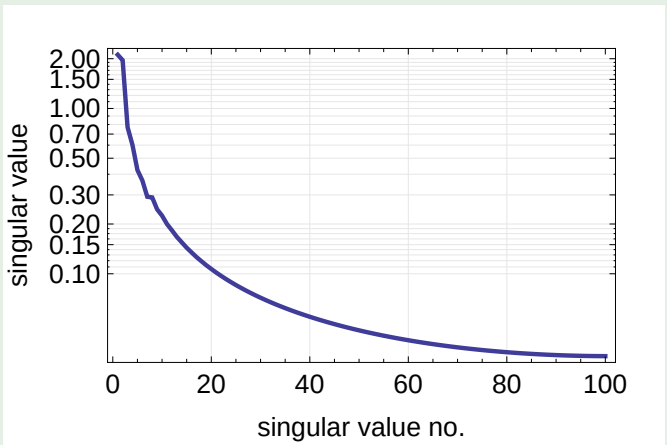
The unique exact solution is $x(t) = \sin t$. To find it numerically, the time interval $[0, 2\pi]$ has been discretized into 100 time steps.

Deconvolution — example

The Toeplitz structure of the discretized 100×100 matrix $\mathbf{H}$.

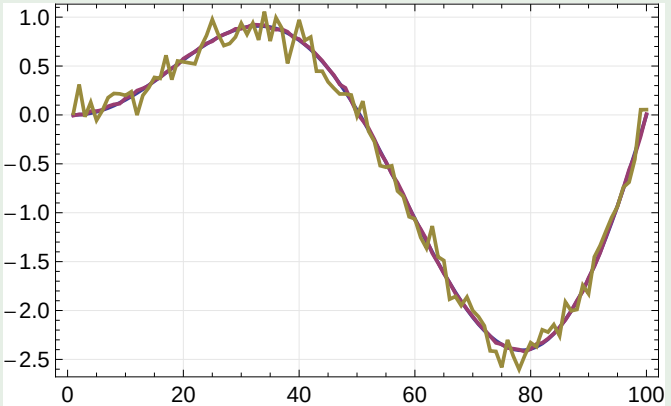


The singular values of $\mathbf{H}$ plotted in the logarithmic scale.
The singular values decay quickly and span across 1.8 orders of magnitude, so that $\kappa(\mathbf{H}) \approx 67$.

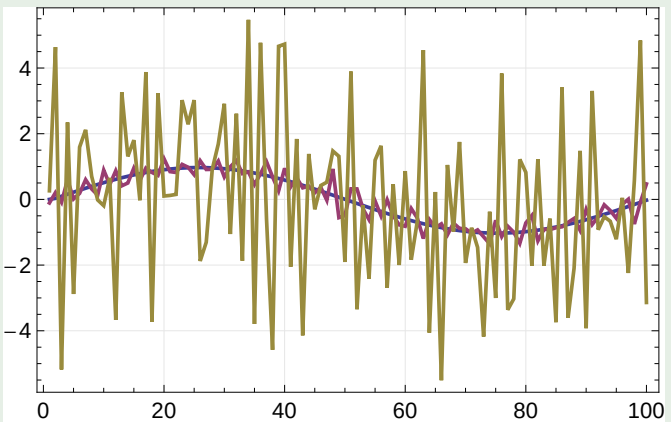


Deconvolution — example

Three discretized right-hand sides (measurements $y(t)$): exact and contaminated with uncorrelated Gaussian measurement errors at 1% and 10% rms levels.



Corresponding computed solutions (the exact solution is $x(t) = \sin t$). Note the increasing noise.



Outline

3. Numerical regularization
   - Regularization for direct methods
   - Regularization for iterative methods
   - Regularization parameter

Numerical regularization

Consider a linear system $\mathbf{Ax} = \mathbf{y}$ with a severely ill-conditioned $\mathbf{A}$.

- Such a system cannot be solved accurately based on $\mathbf{y}$ only, no matter what algorithm is used, because even tiny inaccuracies of $\mathbf{y}$ are highly amplified and dominate the computed $\mathbf{x}$.
- However, the accuracy can be improved if *additional information about some expected or typical characteristics of* $\mathbf{x}$ *is exploited*. This is called numerical regularization of the solution.
- The amount of regularization is controlled by the so-called regulariztion parameter α:
  - When α is too large, the regularization information tends to dominate and distort the computed regularized solution $\mathbf{x}_\alpha$.
  - When α is too small, the measurement noise (amplified via ill-conditioning) tends to dominate and distort the computed regularized solution $\mathbf{x}_\alpha$.

Numerical regularization

The additional information is usually related to smoothness of $\mathbf{x}$ and takes the form of the requirement of a limited magnitude of $\|\mathbf{D}^p\mathbf{x}\|$, where $\mathbf{D}$ is the matrix of the pth order differences,

$$\mathbf{D}^0 = \mathbf{I},$$

$$\mathbf{D}^1 = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 \end{bmatrix},$$

$$\mathbf{D}^2 = \begin{bmatrix} 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

Regularization for direct methods

Consider $\mathbf{Ax} = \mathbf{y}$ and the formula for its SVD-based solution,

$$\mathbf{x} = \mathbf{V}_1 \mathbf{\Sigma}_0^{-1} \mathbf{U}_1^{\mathsf{T}} \mathbf{y} = \left[\sum_{i=1}^{\operatorname{rank} \mathbf{A}} \frac{\mathbf{v}_i \mathbf{u}_i^{\mathsf{T}}}{\sigma_i} \right] \mathbf{y}.$$

- The formula clearly explains the reason of error amplification: divisions by the increasingly tinier singular values.
- In many practical problems, the components of $\mathbf{x}$ and $\mathbf{y}$, which correspond to tiny singular values, are of high frequency[4]. As a result, these components are often disproportionally affected by the noise.

[4] *The singular vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ [of a matrix that arises from the discretization of first-kind Fredholm integral equations] have an increasing number of sign changes in their elements as i increases, i.e., as the corresponding singular values σ_i decrease. Often, the number of sign changes is precisely $i - 1$.* (P. Ch. Hansen, 1995)

TSVD and standard Tikhonov regularization

However, the noisy components can be filtered out from the solution by explicit damping of the excessively amplified components,

$$\tilde{\mathbf{x}} = \left[\sum_{i=1}^{\operatorname{rank} \mathbf{A}} \phi(\sigma_i) \mathbf{v}_i \mathbf{u}_i^{\mathsf{T}} \right] \mathbf{y},$$

where $\phi(\sigma)$ is a filtering function. The most popular choices are

- Truncated Singular Value Decomposition (TSVD)

$$\phi(\sigma) = \begin{cases} \sigma^{-1} & \text{if } \sigma \geq \alpha \\ 0 & \text{otherwise} \end{cases}$$

- Standard Tikhonov regularization with the parameter α

$$\phi(\sigma) = \frac{\sigma}{\sigma^2 + \alpha^2}.$$

In fact, these methods regularize the solution by limiting $\|\mathbf{x}\|$.

Tikhonov regularization

The Tikhonov regularized solution of $\mathbf{Ax} = \mathbf{y}$ is defined as

$$\text{find } \mathbf{x}, \text{ which minimizes } \|\mathbf{y} - \mathbf{Ax}\|^2 + \alpha^2 \|\mathbf{Dx}\|^2,$$

where $\alpha \geq 0$ is the regularization parameter. The solution is regularized with respect to the norm $\|\mathbf{Dx}\|$, where the matrix $\mathbf{D}$ represents the additionally available information. The solution satisfies the regularized counterpart of the normal equation,

$$\left(\mathbf{A}^\mathsf{T}\mathbf{A} + \alpha^2 \mathbf{D}^\mathsf{T}\mathbf{D}\right) \mathbf{x} = \mathbf{A}^\mathsf{T}\mathbf{y},$$

which corresponds to the following least-squares problem:

$$\text{minimize } \left\| \begin{bmatrix} \mathbf{A} \\ \alpha\mathbf{D} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} \right\|^2.$$

Given the SVD of the matrix $\mathbf{A}$, the standard case of $\mathbf{D} = \mathbf{I}$ can be computed directly (see the previous slide).

Regularization for iterative methods

Iterative methods are often used for large-scale problems (direct methods are then impractical).

- The solution is approximated iteratively by computing a sequence of approximated solutions $\mathbf{x}_k$.
- In regularizing iterative methods, the components corresponding to large singular values (low frequency) are retrieved before the components corresponding to small singular values (high frequency). Therefore, the number of iterations plays the role of the regularization parameter: the more iterations, the less regularized the solution.

Regularization for iterative methods

The conjugate gradient least squares method (CGLS) (Lectures B-3 and C-4) is probably the most popular regularizing iterative method.

- In the standard formulation, the CGLS method iteratively minimizes

$$\phi(\mathbf{x}) = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2$$

  and the regularization is with respect to $\|\mathbf{x}\|$.
- A regularization with respect to $\|\mathbf{D}\mathbf{x}\|$ can be performed via a change of variables,

$$\mathbf{w} = \mathbf{D}\mathbf{x}, \qquad \mathbf{x} = \mathbf{D}^{-1}\mathbf{w},$$

  so that the method minimizes iteratively

$$\phi(\mathbf{w}) = \|\mathbf{y} - \mathbf{A}\mathbf{D}^{-1}\mathbf{w}\|^2$$

  with respect to $\|\mathbf{w}\| = \|\mathbf{D}\mathbf{x}\|$.

Regularization parameter

The "amount of regularization" is controlled by the regularization parameter α:

TSVD	number of truncated singular values
Tikhonov	weighting parameter α
iterative	number of iterations

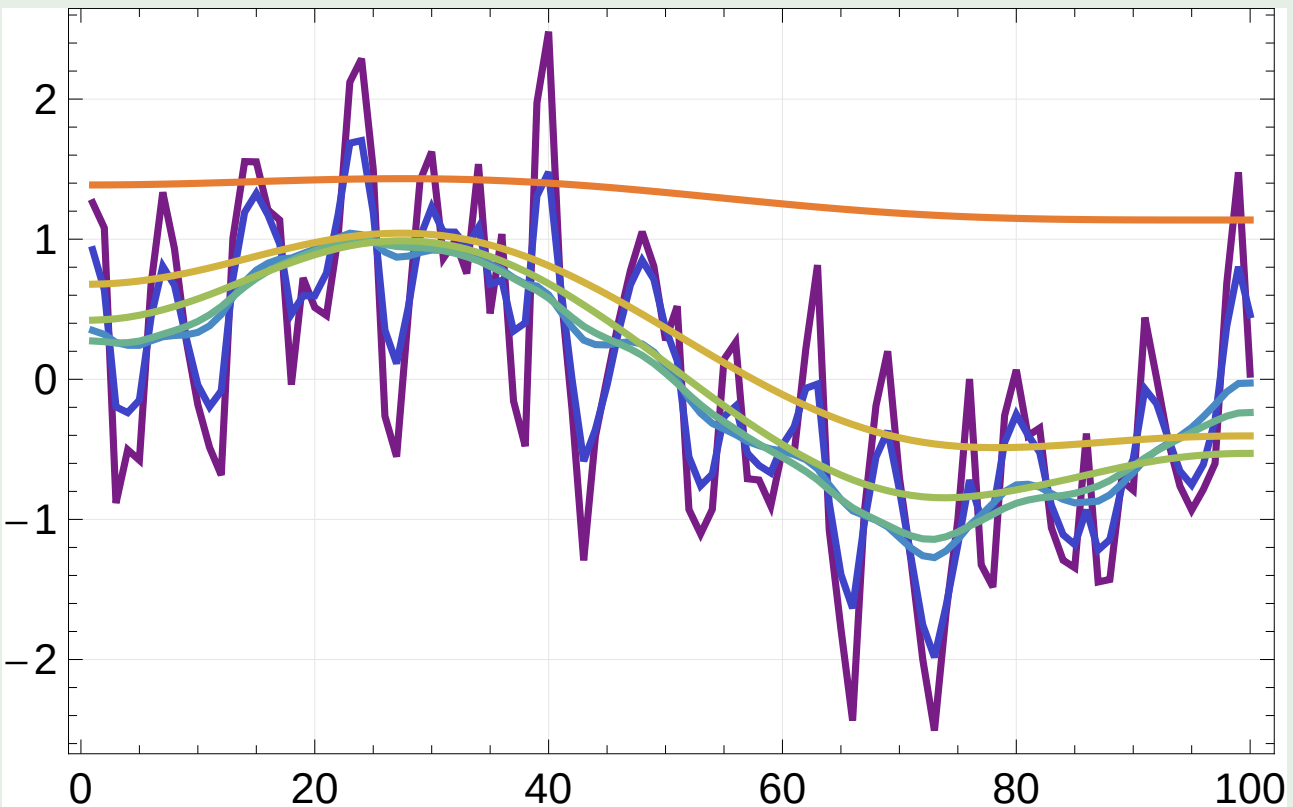
In all regularization methods, direct or iterative, the proper choice of the regularization parameter α is crucial:

- Too small α results in a computed solution being too noisy.
- If α is too large, the computed solution is overly distorted by the regularizing condition (overregularized).

Regularization parameter — example

Tikhonov-regularized solution to the deconvolution example at 10% measurement error (exact solution $x(t) = \sin t$)

$$\mathbf{D} = \mathbf{D}^1, \qquad \alpha = 0.05,\ 0.1,\ 0.5,\ 1,\ 5,\ 10,\ 50.$$



Regularization parameter — the L-curve method

In the absence of information about the noise level in $\mathbf{y}$, all methods of choice of the regularization parameter are more or less heuristic. The most popular method is the L-curve method[5].

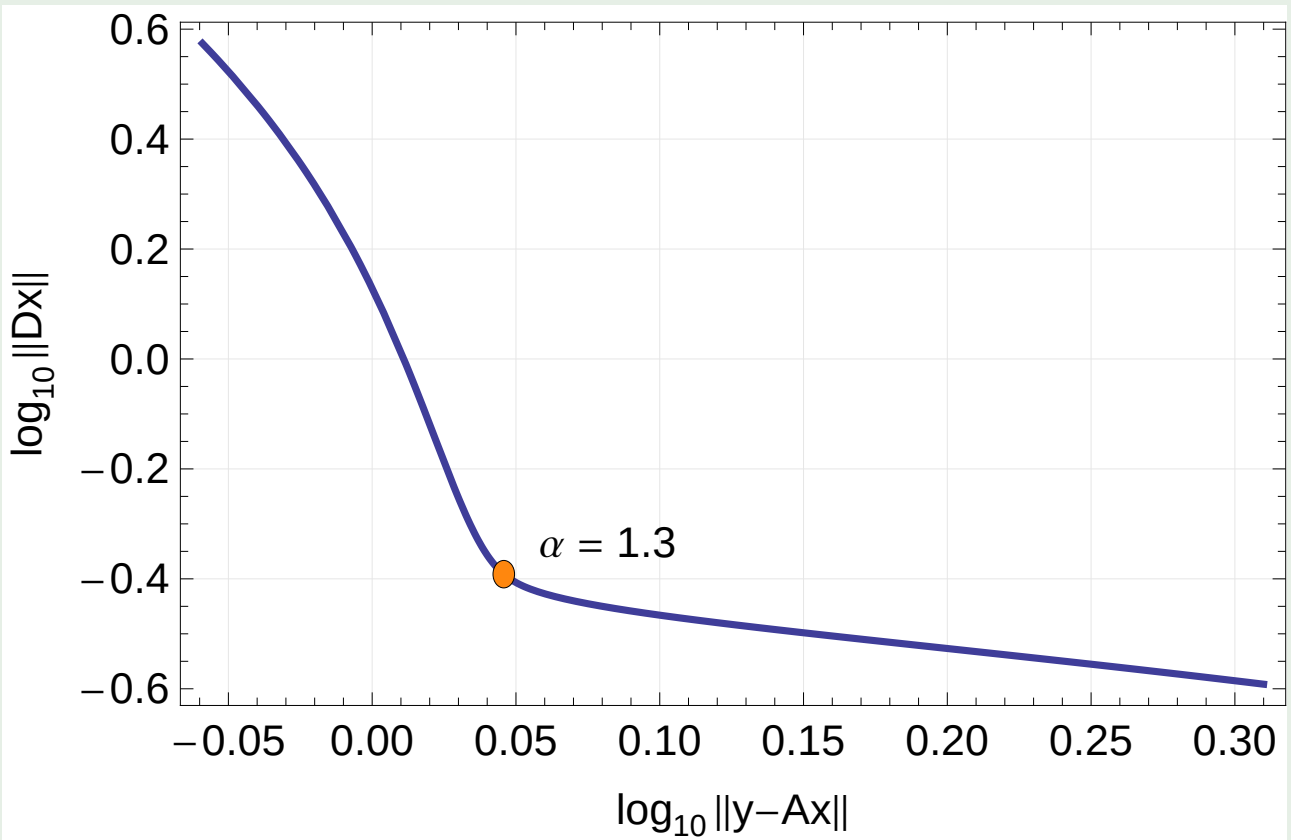
- The L-curve is a log-log plot of the norm of the regularization condition $\|\mathbf{Dx}\|$ versus the norm of the residuum $\|\mathbf{y} - \mathbf{Ax}\|$ in dependence on the regularization parameter α.
- The L-curve usually consists of two branches corresponding to
  - excessive (horizontal, lower) and
  - insufficient (vertical, upper)

  regularization. The corner is assumed to correspond to the proper value of α; it is usually defined as the point of maximum curvature.

[5] For an overview and comparison of other methods, see F. Bauer, M.A. Lukas (2011) Comparing parameter choice methods for regularization of ill-posed problems. *Mathematics and Computers in Simulation*, 81(9):1795–1841. doi:10.1016/j.matcom.2011.01.016

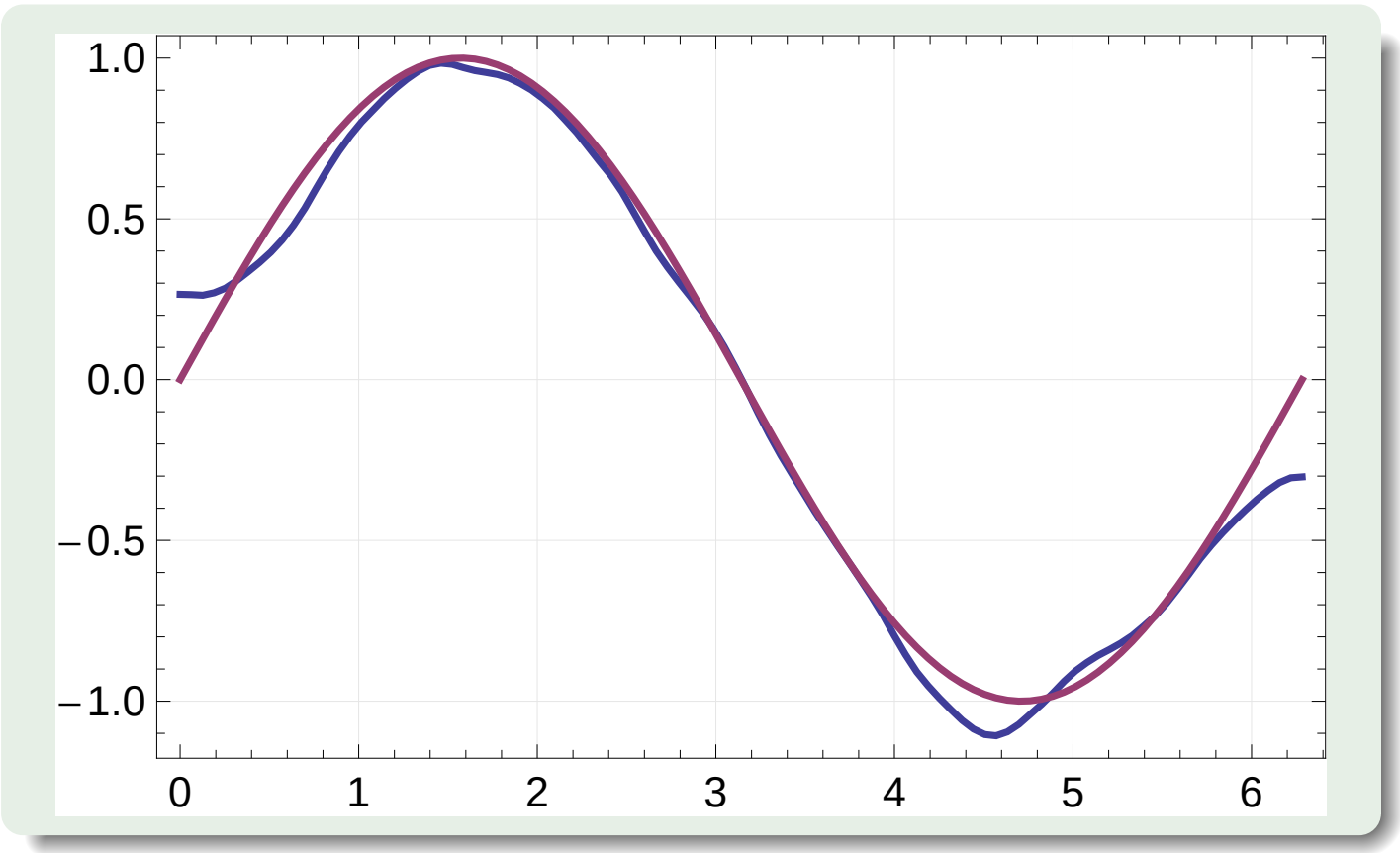
Regularization parameter — the L-curve — example

The deconvolution example at 10% measurement error ($\mathbf{D} = \mathbf{D}^1$)



Regularization parameter — the L-curve — example

The deconvolution example at 10% measurement error, the regularized solution compared to the exact solution ($\mathbf{D} = \mathbf{D}^1$, $\alpha = 1.3$ found by the L-curve method)



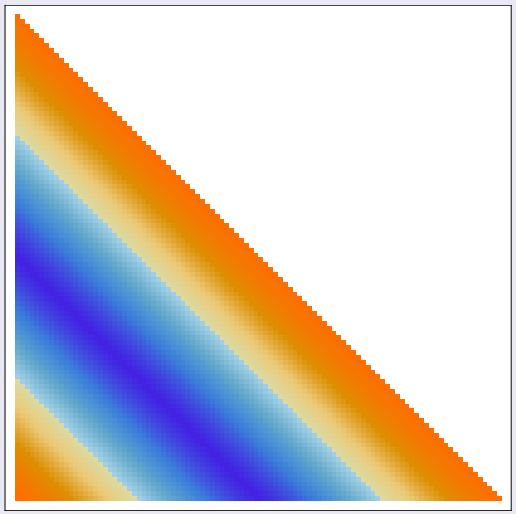
Outline

Toeplitz matrices

Toeplitz matrices

A matrix $\mathbf{H} = [h_{ij}]$ is called a Toeplitz matrix, if it satisfies $h_{ij} = h_{i-j}$,

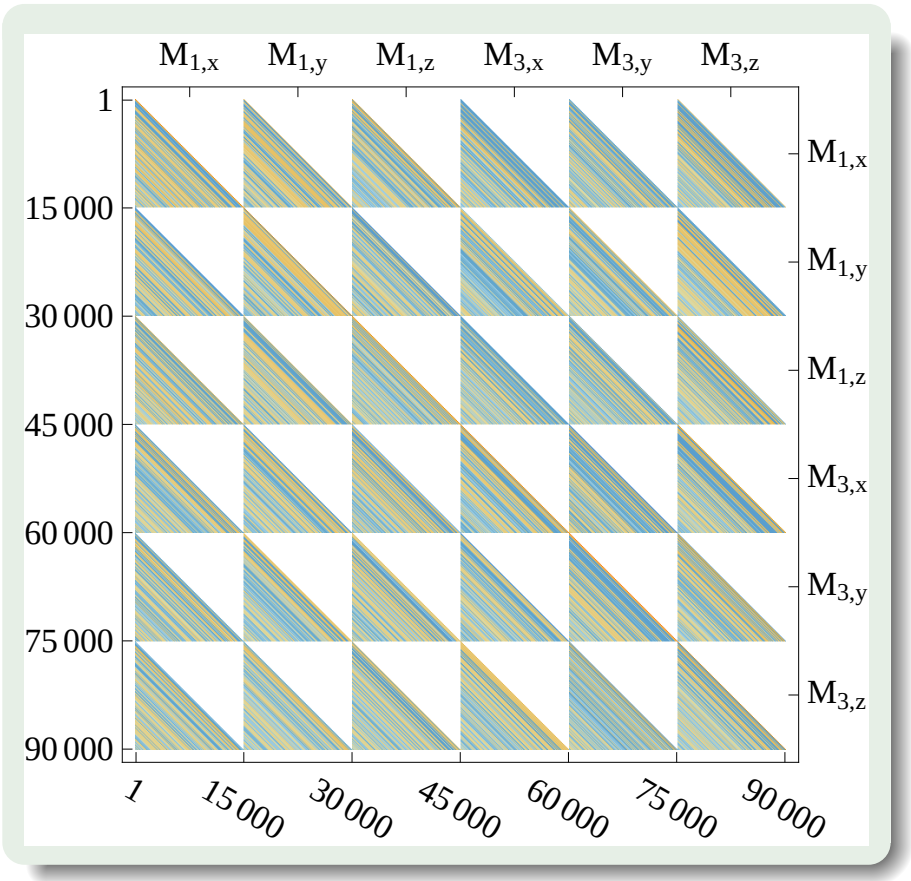
$$\mathbf{H} = \begin{bmatrix} h_0 & h_{-1} & h_{-2} & \cdots & h_{-n+1} \\ h_1 & h_0 & h_{-1} & \cdots & h_{-n+2} \\ h_2 & h_1 & h_0 & \cdots & h_{-n+3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{n-1} & h_{n-2} & h_{n-3} & \cdots & h_0 \end{bmatrix}$$



Systems with Toeplitz matrices often arise from the disretization of linear integral equations in time domain, e.g. in inverse problems of input identification.

Block matrices with Toeplitz blocks

Discretization of matrix linear integral equations, e.g. in problems of input identification in MIMO[6] systems, may yield linear equations with a block matrix with Toeplitz blocks.



[6] Multiple inputs, multiple outputs.

Toeplitz systems — common problems

Common problems:

Memory If many time steps are considered, such a matrix and the space required for its storage can be huge.

Time Most of direct operations on such large matrices can be too time-consuming.

Accuracy Large Toeplitz matrices are usually extremely ill-conditioned.

Toeplitz matrices correspond to discrete convolutions: use frequency domain, if possible! But in transient analysis, other problems can then appear:

- spectral leakage
- windowing functions
- regularization

Memory

$$\mathbf{H} = \begin{bmatrix} h_0 & h_{-1} & h_{-2} & \cdots & h_{-n+1} \\ h_1 & h_0 & h_{-1} & \cdots & h_{-n+2} \\ h_2 & h_1 & h_0 & \cdots & h_{-n+3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{n-1} & h_{n-2} & h_{n-3} & \cdots & h_0 \end{bmatrix}$$

- A general $n \times n$ matrix requires n^2 storage for all elements h_{ij}.
- An $n \times n$ Toeplitz matrix is defined by $2n - 1$ elements: $h_{-n+1}, \ldots, h_{n-1}$. Such a matrix can be thus stored in a reduced form. For large matrices, the difference is tremendous: e.g. 7.5 GB vs. 176 kB for a 90000×90000 **double** matrix.
- This is typical for structured systems (and for sparse systems).

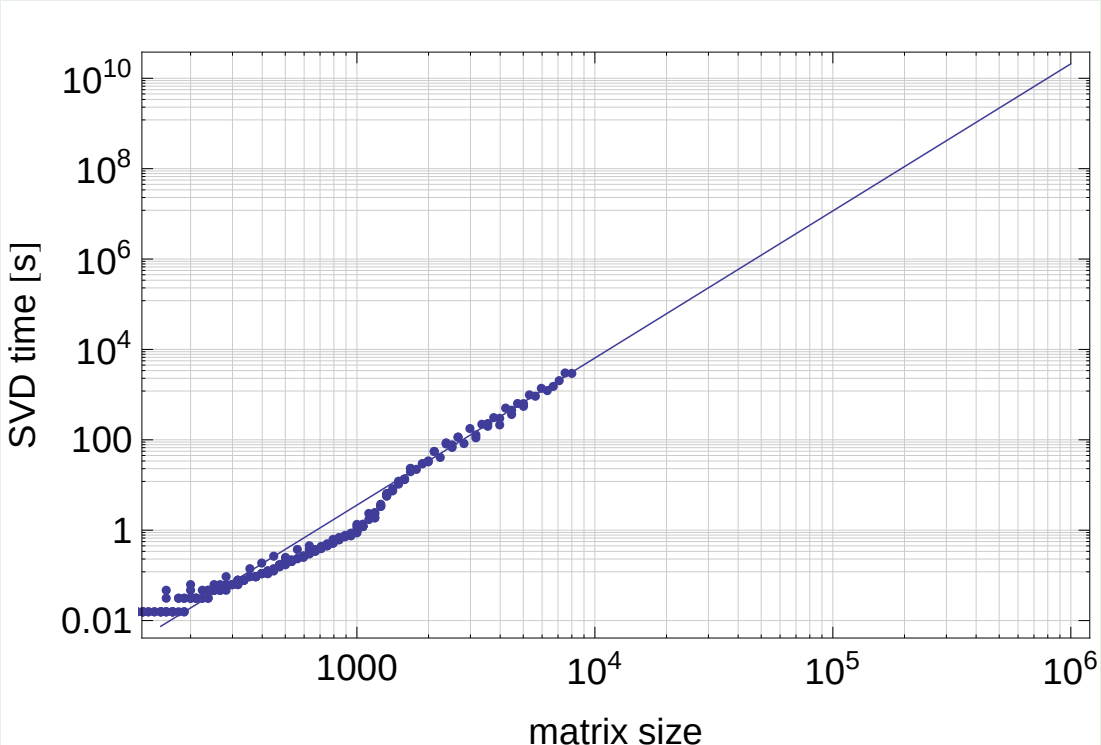
In case of discretization of time-convolutions, lower-triangular Toeplitz systems may arise, which are defined by n elements only.

Time

Iterative methods

Most of direct operations on large Toeplitz matrices, including all decompositions and factorizations, would be too time-consuming. Iterative methods will yield a reasonably accurate solution in a much shorter time.

SVD time (PC, Wolfram Mathematica 8)



Time

Quick matrix-vector multiplication

Most of iterative methods (like CGLS) make extensive use of matrix-vector multiplications. Standard multiplications requires $O(n^2)$ operations. However, Toeplitz matrices are discrete versions of convolution operators.

- In continuous time, the convolution can be performed in frequency domain by simple multiplication of spectra:

$$\mathcal{F}(f \star g) = (\mathcal{F}f)(\mathcal{F}g),$$

so that

$$(f \star g)(t) = \mathcal{F}^{-1}\left[(\mathcal{F}f)(\mathcal{F}g)\right](t). \qquad (\star)$$

- In discrete time, the fast Fourier transform (FFT) can be used, which requires only $O(n \log n)$ time. However, the discrete counterpart of $(\star)$ is valid only for *circulant matrices*, which form a subset of Toeplitz matrices.

Time

Circulant matrices

Teoplitz matrix

A matrix $\mathbf{H} = [h_{ij}]$, which satisfies $h_{ij} = h_{i-j}$, is called a Toeplitz matrix.

Circulant matrix

A Toeplitz matrix $\mathbf{H} = [h_{ij}]$, where $h_{ij} = h_{i-j}$, is called a circulant matrix, if $h_{-k} = h_{n-k}$.

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 3 & 0 & 1 & 2 \\ 2 & 3 & 0 & 1 \\ 1 & 2 & 3 & 0 \end{bmatrix}$$

Time

Quick matrix-vector multiplication (circulant matrices)

An $n \times n$ circulant matrix $\hat{\mathbf{H}}$ can be multiplied by any vector $\hat{\mathbf{x}}$ in $O(n \log n)$ time instead of $O(n^2)$ time via the FFT[7]:

$$\hat{\mathbf{H}}\hat{\mathbf{x}} = \mathrm{FFT}^{-1}\left[\mathrm{FFT}(\hat{\mathbf{h}})\,\mathrm{FFT}(\hat{\mathbf{x}})\right],$$

where $\hat{\mathbf{h}}$ is the first column of $\hat{\mathbf{H}}$.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

Toeplitz matrices that arise from the discretization of time convolutions are usually lower-triangular and so they are not circulant.

[7] Depending on the exact definition of the FFT, sometimes the right-hand side has to be multiplied by $\sqrt{n}$.

Time

Quick matrix-vector multiplication (non-circulant Toeplitz matrices)

However, non-circulant Toeplitz matrices can be augmented to become circulant:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \\ 4 & 3 & 2 & 1 \end{bmatrix} \longrightarrow \mathbf{\hat{H}} = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 4 & 3 & 2 \\ 2 & 1 & 0 & 0 & 0 & 0 & 4 & 3 \\ 3 & 2 & 1 & 0 & 0 & 0 & 0 & 4 \\ 4 & 3 & 2 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 4 & 3 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 4 & 3 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 & 3 & 2 & 1 \end{array}\right]$$

Time

Quick matrix-vector multiplication (non-circulant Toeplitz matrices)

A product of a non-circulant Toeplitz matrix with a vector, $\mathbf{Hx}$, can be computed via the augmented version of the matrix by dropping the trailing zeros from the product $\hat{\mathbf{H}}\hat{\mathbf{x}}$, where $\hat{\mathbf{x}} = \begin{bmatrix}\mathbf{x}^T \mathbf{0}^T\end{bmatrix}^T$, that is, where $\hat{\mathbf{x}}$ is the original vector $\mathbf{x}$ padded with zeros.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 2 & 1 & 1 \\ 4 & 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} \sim \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 4 & 3 & 2 \\ 2 & 1 & 0 & 0 & 0 & 0 & 4 & 3 \\ 3 & 2 & 1 & 0 & 0 & 0 & 0 & 4 \\ 4 & 3 & 2 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 4 & 3 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 4 & 3 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 & 3 & 2 & 1 \end{array}\right] \left[\begin{array}{c} w \\ x \\ y \\ z \\ \hline 0 \\ 0 \\ 0 \\ 0 \end{array}\right]$$

Accuracy

Large Toeplitz systems are usually extremely ill-conditioned

- use a regularizing iterative method, like CGLS, which will retrieve well-conditioned components before the ill-conditioned components
- The number of iterations will play the role of the regularization parameter

  the more iterations, the less regularized the solution

A not-so-large example (10 000 × 10 000)

Consider the deconvolution problem:

$$y(t) = \int_0^t \cos(t-s)x(s)ds,$$

where $y(t)$ is the velocity response of an undamped single DOF system to a force excitation $x(t)$.

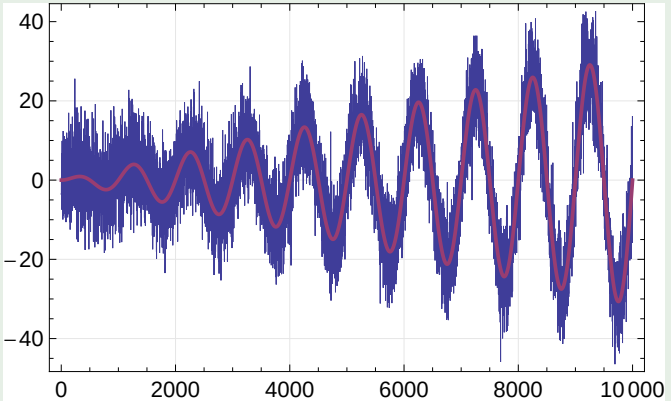
Assume the observed response

$$y(t) = \frac{1}{2}t\sin t, \qquad t \in [0, 10 \times 2\pi].$$

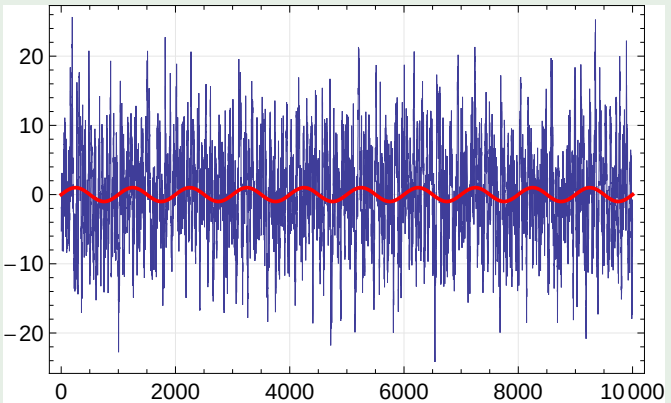
The unique exact solution is $x(t) = \sin t$. To find it numerically, the time interval $[0, 10 \times 2\pi]$ has been discretized into 10 000 time steps.

A not-so-large example (10 000 × 10 000)

Two discretized right-hand sides (measurements $y(t)$): exact and contaminated with uncorrelated Gaussian measurement errors at 50% rms level.



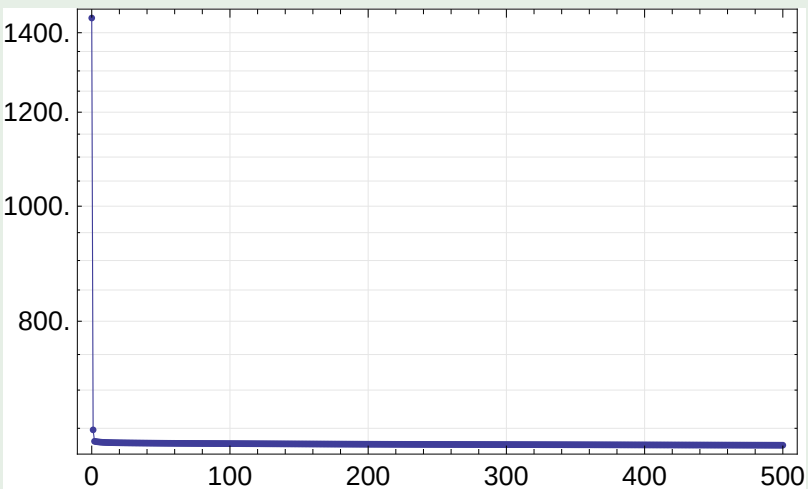
The exact solution is $x(t) = \sin t$. The solution computed in the noisy case (without regularization, here the 10 000 iteration) would be useless.

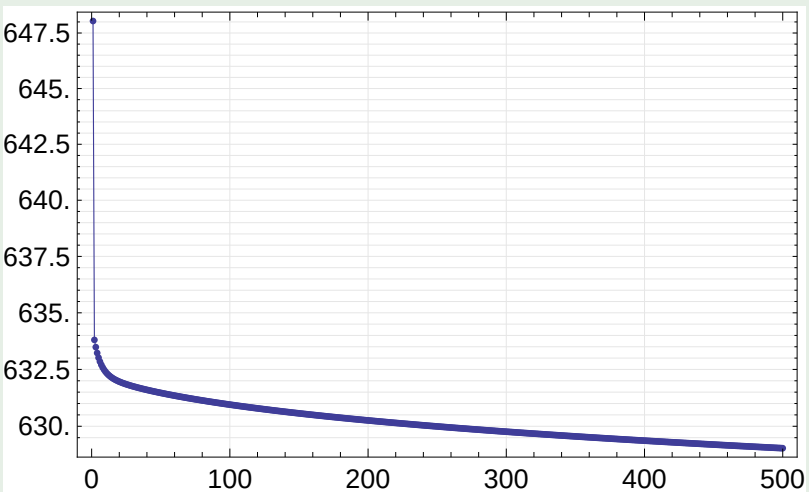


A not-so-large example ($10\,000 \times 10\,000$)

Residuum norms of the iterates

A common stop condition for CGLS is based on the norm of the residuum $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|$.



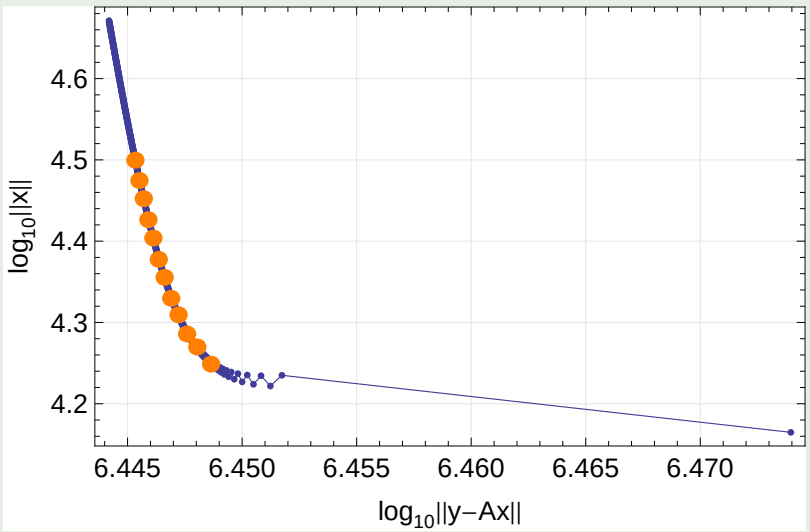


A not-so-large example (10 000 × 10 000)

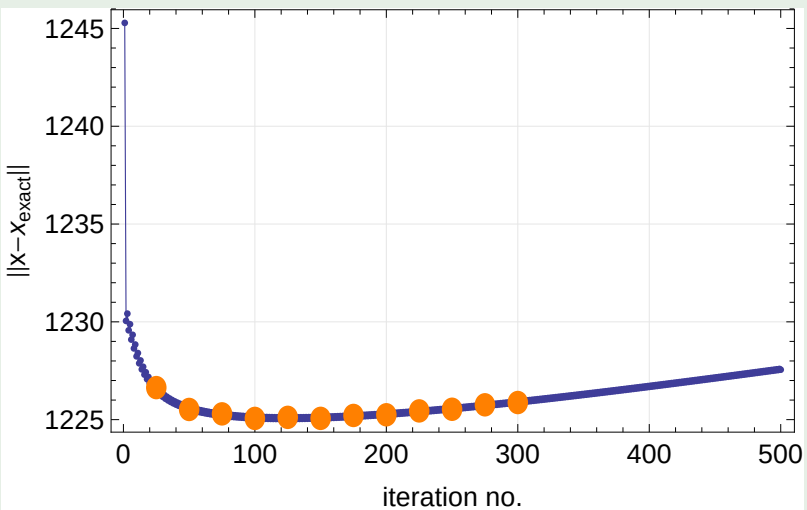
L-curve and the accuracy

The stop condition can be also
based on the L-curve (as the actual accuracy is in practice unknown).

L-curve



actual accuracy

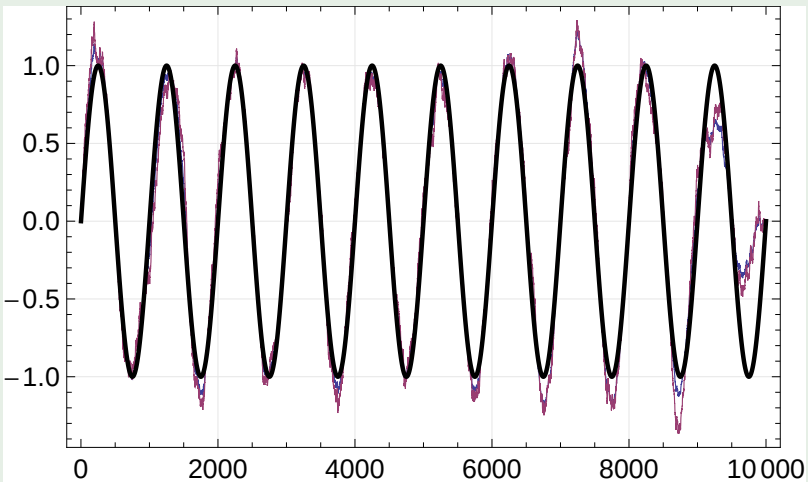


Orange dots mark the iterates no. $25, 50, 75, \ldots, 300$.

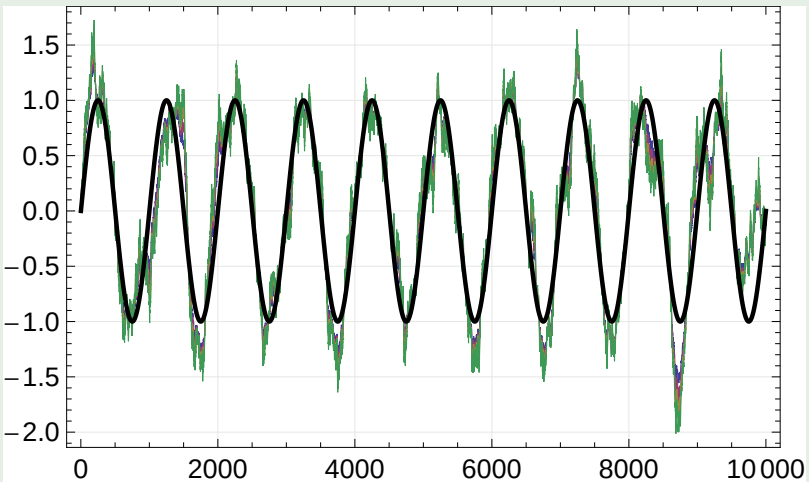
A not-so-large example ($10\,000 \times 10\,000$)

CGLS iterates

iterates $25, 50$



iterates $75, 100, 125, 150$



Outline

Further reading

- Linear least-squares problems
  - G. Dahlquist, Å. Björck, Linear Least Squares Problems, [in:] *Numerical Methods in Scientific Computing*, vol. 2.
- Conditioning, regularization
  - P.Ch. Hansen, Discrete inverse problems: Insight and Algorithms, SIAM 2010.
  - P.Ch. Hansen, Deconvolution and regularization with Toeplitz matrices, Numerical Algorithms 29:323–378, 2002.
  - F. Bauer, M.A. Lukas, Comparing parameter choice methods for regularization of ill-posed problems. Mathematics and Computers in Simulation 81(9):1795–1841, 2011.
- Large Toeplitz systems:
  - I. Gohberg, V. Olshevsky, Fast algorithms with preprocessing for matrix-vector multiplication problems, J. Complexity, 10(4):411–427, 1994.
  - I. Gohberg, V. Olshevsky, Complexity of multiplication with vectors for structured matrices, Linear Algebra Appl., 202:163–192, 1994.

Outline

Homework 8 (25 points)

Regularization and iterative linear solvers

Available soon at `http://info.ippt.pan.pl/~ljank`.

E-mail the answer and the source code to `ljank@ippt.pan.pl`.