Lectures on Linear Inverse Problems

Per Christian Hansen

Technical University of Denmark

1. Introduction to ill-posed problems.
2. More insight into their behavior and treatment.
3. Discrete ill-posed problems.
4. Regularization methods for discrete ill-posed problems.
5. Parameter-choice methods.
6. Iterative regularization methods.
7. Large-scale problems.

Contents of This Lecture

The three IPs:

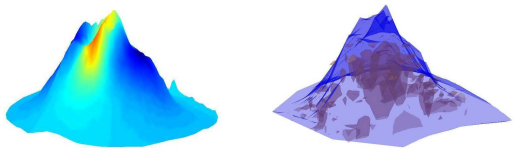
1. **I**nverse **P**roblems.
   (a) Motivation.
   (b) Characterization.
2. **I**ll-Conditioned **P**roblems.
   (a) A small example.
   (b) Stabilization.
3. **I**ll-Posed **P**roblems.
   (a) Definition and properties.
   (b) Examples.

What to do with these IPs?

Motivation: Why Inverse Problems?

A large-scale example, coming from a collaboration with the University of Naples.

From measurements of the magnetic field above Vesuvius, determine the activity inside the volcano.

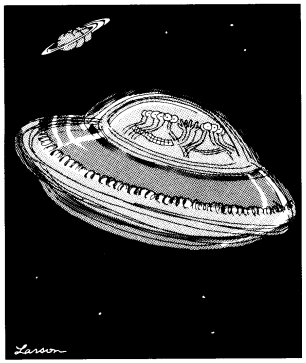


Measurements on the surface

Reconstruction inside the volcano

Another Example: the Hubble Space Telescope

For several years, the HST produced blurred images.



Another photograph from the Hubble telescope

Inverse Problems

. . . typically arise when one wants to compute information about some "interior" properties using "exterior" measurements.

$$\boxed{\int_\Omega \text{input} \times \text{system}\; d\Omega = \text{output}}$$

Image restoration

$$\text{scenery} \rightarrow \text{lens} \rightarrow \text{image}$$

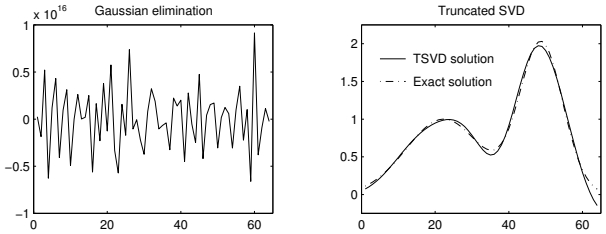
Tomography

$$\text{X-ray source} \rightarrow \text{object} \rightarrow \text{damping}$$

Seismology

$$\text{seismic wave} \rightarrow \text{layers} \rightarrow \text{reflections}$$

Computational Issues



- Standard numerical methods produce useless results.
- Specialized methods can produce "reasonable" results.

The Mechanisms of Ill-Conditioned Problems

Consider a linear system with coefficient matrix and right-hand side

$$A = \begin{pmatrix} 0.16 & 0.10 \\ 0.17 & 0.11 \\ 2.02 & 1.29 \end{pmatrix}, \qquad b = \begin{pmatrix} 0.27 \\ 0.25 \\ 3.33 \end{pmatrix} = A \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0.01 \\ -0.03 \\ 0.02 \end{pmatrix}.$$

There is no vector x such that $A\,x = b$.

The least squares solution, which solves the problem

$$\min_x \|A\,x - b\|_2,$$

is given by

$$x_{\text{LSQ}} = \begin{pmatrix} 7.01 \\ -8.40 \end{pmatrix} \qquad \Rightarrow \qquad \|A\,x_{\text{LSQ}} - b\|_2 = 0.022\ .$$

Far from exact solution $(\,1\,,\,1\,)^T$ yet the residual is small.

Other Solutions with Small Residual

Two other "solutions" with a small residual are

$$x_{\text{B}}^{(1)} = \begin{pmatrix} 1.65 \\ 0 \end{pmatrix} \qquad \Rightarrow \qquad \|A\,x_{\text{B}}^{(1)} - b\|_2 = 0.031$$

$$x_{\text{B}}^{(2)} = \begin{pmatrix} 0 \\ 2.58 \end{pmatrix} \qquad \Rightarrow \qquad \|A\,x_{\text{B}}^{(1)} - b\|_2 = 0.036\ .$$

All the "solutions" x_{LSQ}, $x_{\text{B}}^{(1)}$ and $x_{\text{B}}^{(2)}$ have small residuals, yet they are far from the exact solution!

- The matrix A is ill conditioned.
- Small perturbations of the data (here: b) can lead to large perturbations of the solution.
- A small residual does not imply a good solution.

(All this is well known stuff from matrix computations.)

Stabilization!

It turns out that we can modify the problem such that the solution is more stable, i.e., less sensitive to perturbations.

Example: enforce an upper bound on the solution norm $\|x\|_2$:

$$\min_x \|A\,x - b\|_2 \qquad \text{subject to} \qquad \|x\|_2 \le \alpha \ .$$

The solution x_α depends in a nonlinear way on α:

$$x_{0.1} = \begin{pmatrix} 0.08 \\ 0.05 \end{pmatrix} \ , \qquad x_1 = \begin{pmatrix} 0.84 \\ 0.54 \end{pmatrix}$$

$$x_{1.385} = \begin{pmatrix} 1.17 \\ 0.74 \end{pmatrix} \ , \qquad x_{10} = \begin{pmatrix} 6.51 \\ -7.60 \end{pmatrix} \ .$$

By supplying the correct additional information we can compute a good approximate solution.

Inverse Problems → Ill-Conditioned Problems

Whenever we solve an inverse problem on a computer, we face difficulties because the computational problems are ill conditioned.

The purpose of my lectures are:

1. To explain why ill-conditioned computations always arise when solving inverse problems.
2. To explain the fundamental "mechanisms" underlying the ill conditioning.
3. To explain how we can modify the problem in order to stabilize the solution.
4. To show how this can be done efficiently on a computer.

Regularization methods is at the heart of all this.

Inverse Problems are Ill-Posed Problems

Hadamard's definition of a *well-posed problem* (early 20th century):

1. The problem must have a solution,
2. the solution must be unique, and
3. it must depend continuously on data and parameters.

If the problem violates any of these requirements, it is *ill posed*.

Condition 2 can be "fixed" by additional requirements to the solution, e.g., that of minimum norm.

Condition 3 is harder to "fix" because it implies that

- arbitrarily small perturbations of data and parameters can produce arbitrarily large perturbations of the solution.

Fredholm Integral Equations of the First Kind

Our generic inverse problem:

$$\int_0^1 K(s,t)\,f(t)\,dt = g(s), \qquad 0 \le s \le 1 \ .$$

Here, the *kernel* $K(s,t)$ and the *right-hand side* $g(s)$ are known functions, while $f(t)$ is the unknown function.

In multiple dimensions, this equation takes the form

$$\int_{\Omega_{\mathbf{t}}} K(\mathbf{s},\mathbf{t})\,f(\mathbf{t})\,d\mathbf{t} = g(\mathbf{s}), \qquad \mathbf{s} \in \Omega_{\mathbf{s}} \ .$$

An important special case: *deconvolution*

$$\int_0^1 h(s-t)\,f(t)\,dt = g(s), \qquad 0 \le s \le 1$$

(and similarly in more dimensions).

The Riemann-Lebesgue Lemma

Consider the function

$$f(t) = \sin(2\pi p\, t)\ , \qquad p = 1, 2, \ldots$$

then for $p \to \infty$ and "arbitrary" K we have

$$g(s) = \int_0^1 K(s,t)\, f(t)\, dt \to 0\ .$$

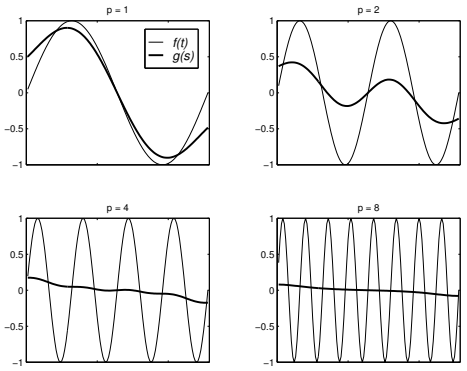
Smoothing: high frequencies are damped in the mapping $f \mapsto g$.

Hence, the mapping from g to f must amplify the high frequencies.

Therefore we can expect difficulties when trying to reconstruct f from noisy data g.

Illustration of the Riemann-Lebesgue Lemma

Gravity problem with $f(t) = \sin(2\pi p\, t)$, $p = 1$, 2, 4, and 8.



A Problem with no Solution

Ursell (1974) presented the following innocently-looking problem:

$$\int_0^1 \frac{1}{s+t+1}\, f(t)\, dt = 1, \qquad 0 \le s \le 1.$$

This problem *has no square integrable solution!*

Investigation of the Ursell Problem

The kernel has a set of orthonormal eigenfunctions ϕ_i such that

$$\int_0^1 \frac{1}{s+t+1}\, \phi_i(t)\, dt = \lambda_i\, \phi_i(s), \qquad i = 1, 2, \ldots$$

Expand the right-hand side $g(s) = 1$ in terms of the eigenfunctions:

$$g_k(s) = \sum_{i=1}^{k} (\phi_i, g)\, \phi_i(s); \qquad \|g - g_k\|_2 \to 0 \text{ for } k \to \infty.$$

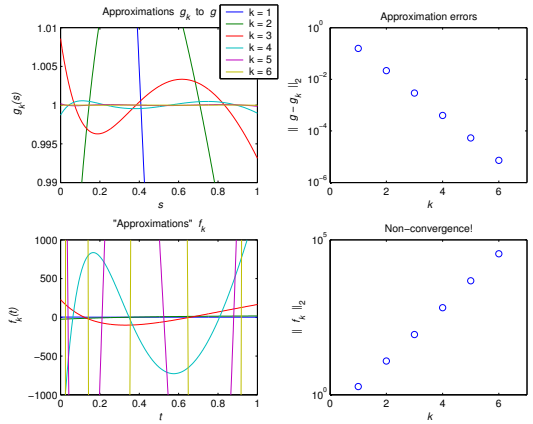
Now consider the expansion

$$f_k(t) = \sum_{i=1}^{k} \frac{(\phi_i, g)}{\lambda_i}\, \phi_i(t).$$

Each f_k is obviously a solution to $\int_0^1 \frac{f(t)}{s+t+1}\, dt = g_k(s)$; but

$$\|f_k\|_2 \to \infty \text{ for } k \to \infty.$$

Ursell Problem – Numerical Results



Why do We Care?

Why bother about these (strange) issues?

- Ill-posed problems model a variety of real applications:
  - Medical imaging (brain scanning, etc.)
  - Geophysical prospecting (search for oil, land-mines, etc.)
  - Image deblurring (astronomy, CSI[a], etc.)
  - Deconvolution of instrument's response.
- We can only hope to compute useful solutions to these problems if we fully understand their *inherent* difficulties ...
- and how these difficulties carry over to the discretized problems involved in a computer solution,
- and how to deal with them in a satisfactory way.

[a] Crime Scene Investigation.

Some Important Questions

- How to discretize the inverse problem; here, the integral equation?
- Why is the matrix in the discretized problem always so ill conditioned?
- Why can we still compute an approximate solution?
- How can we compute it stably and efficiently?
- Is additional information available?
- How can we incorporate it in the solution scheme?
- How should we implement the numerical scheme?
- How do we solve large-scale problems?

Contents of The Second Lecture

1. Model problems
   (a) Deconvolution
   (b) Gravity surveying
2. The singular value expansion (SVE)
   (a) Formulation
   (b) The smoothing effect
   (c) The discrete Picard condition
3. Discretization
   (a) Quadrature methods
   (b) Galerkin methods

Model Problem: Deconvolution

Continuous form of (de)convolution:

$$\int_0^1 h(s-t)\, f(t)\, dt = g(s)\ , \qquad 0 \le s \le 1\ .$$

Discrete periodic signals of length N:

$$\begin{aligned} \mathrm{DFT}(g) &= \mathrm{DFT}(f) \odot \mathrm{DFT}(h) \\ f &= \mathrm{IDFT}\left(\mathrm{DFT}(g) \oslash \mathrm{DFT}(h)\right) \end{aligned}$$

where

$$\odot \text{ and } \oslash = \text{elementwise multiplication/division}$$

and the discrete Fourier transform $\mathrm{DFT}(f)$ is defined by

$$[\mathrm{DFT}(f)]_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j\, e^{-i\,2\pi jk/N}, \qquad k = 0, 1, \ldots, N-1\ .$$

Example from Signal Processing

Noisy discrete signal $\widetilde{g} = g + e$, where e is white noise:

$$\mathrm{DFT}(\widetilde{g}) = \mathrm{DFT}(g) + w,$$

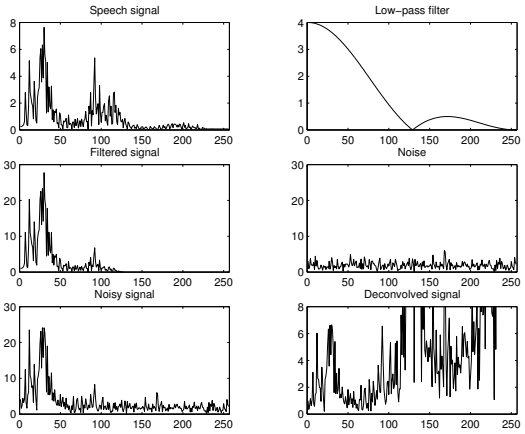
where all elements in $w = \mathrm{DFT}(e)$ have the same probability.

The "naive" expression for the solution $\widetilde{f}$ becomes

$$\begin{aligned} \mathrm{DFT}(\widetilde{f}) &= \mathrm{DFT}(\widetilde{g}) \oslash \mathrm{DFT}(h) \\ &= \mathrm{DFT}(g) \oslash \mathrm{DFT}(h) + w \oslash \mathrm{DFT}(h) \\ &= \mathrm{DFT}(f) + w \oslash \mathrm{DFT}(h)\ . \end{aligned}$$

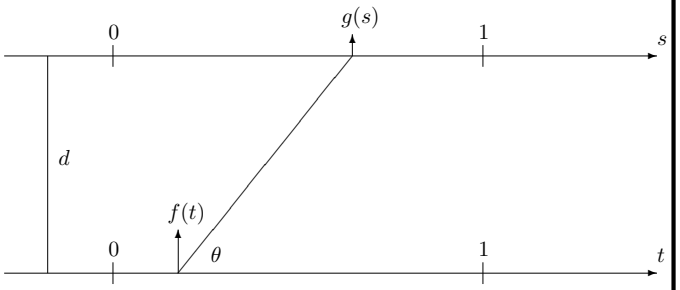
The last term represent high-frequent noise!

Power Spectra



Model Problem: Gravity Surveying

- Unknown mass density distribution $f(t)$ at depth d below surface, from 0 to 1 on t axis.
- Measurements of vertical component of gravitational field $g(s)$ at surface, from 0 to $b1$ on the s axis.



Setting Up the Integral Equation

The value of $g(s)$ due to the part dt on the t axis

$$dg = \frac{\sin\theta}{r^2}\, f(t)\, dt \ ,$$

where $r = \sqrt{d^2 + (s-t)^2}$. Using that $\sin\theta = d/r$, we get

$$\frac{\sin\theta}{r^2}\, f(t)\, dt = \frac{d}{(d^2 + (s-t)^2)^{3/2}}\, f(t)\, dt \ .$$

The total value of $g(s)$ for $a \le s \le b$ is therefore

$$g(s) = \int_0^1 \frac{d}{(d^2 + (s-t)^2)^{3/2}}\, f(t)\, dt \ .$$

This is the forward problem.

Our Integral Equation

Fredholm integral equation of the first kind:

$$\int_0^1 \frac{d}{(d^2 + (s-t)^2)^{3/2}}\, f(t)\, dt = g(s) \ , \qquad a \le s \le b \ .$$

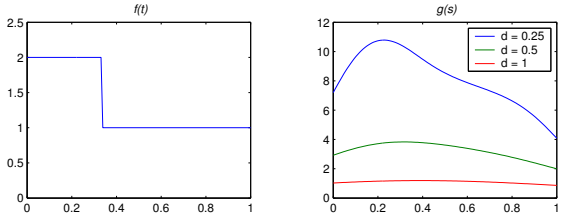
The kernel K, which represents the model, is

$$K(s,t) = h(s-t) = \frac{d}{(d^2 + (s-t)^2)^{3/2}} \ ,$$

and the right-hand side g is what we are able to measure.

From K and g we want to compute f, i.e., an inverse problem.

Numerical Examples



Observations:

- The signal/"data" $g(s)$ is a smoothed version of the source $f(t)$.
- The deeper the source, the weaker the signal.
- The discontinuity in $f(t)$ is not visible in $g(s)$.

The Singular Value Expansion (SVE)

For any square integrable kernel K holds

$$K(s,t) = \sum_{i=1}^{\infty} \mu_i \, u_i(s) \, v_i(t)$$

The "fundamental relation"

$$\int_0^1 K(s,t) \, v_i(t) \, dt = \mu_i \, u_i(s) \; , \qquad i = 1, 2, \ldots$$

and the expression for the solution

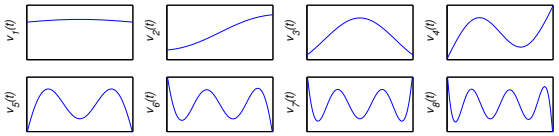
$$f(t) = \sum_{i=1}^{\infty} \frac{(u_i, g)}{\mu_i} \, v_i(t) \; .$$

The Smoothing Effect

The "smoother" the kernel K, the faster the μ_i decay to zero:

- If the derivatives of order $0, \ldots, q$ exist and are continuous, then μ_i is approximately $\mathcal{O}(i^{-q-1/2})$.

The smaller the μ_i, the more oscillations (or zero-crossings) in the singular functions u_i and v_i.



Since $v_i(t) \to \mu_i \, u_i(s)$, higher frequencies are damped more than lower frequencies (smoothing) in the forward problem.

The Picard Condition

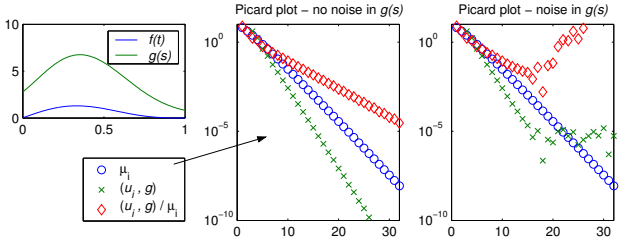
In order that there exists a square integrable solution f to the integral equation, the right-hand side g must satisfy

$$\sum_{i=1}^{\infty} \left(\frac{(u_i, g)}{\mu_i} \right)^2 < \infty \; .$$

Equivalent condition: $g \in \text{range}(K)$.

Main difficulty: a noisy g does not satisfy the PC!

Illustration of the Picard Condition



The violation of the Picard condition is the simple explanation of the instability of linear inverse problems in the form of first-kind Fredholm integral equations.

SVE analysis + Picard plot → insight → remedy → algorithms.

Example of SVE (Degenerate)

We can occasionally calculate the SVE analytically. Example

$$\int_{-1}^{1} (s + 2t)\, f(t)\, dt = g(s), \qquad -1 \le s \le 1.$$

For this kernel $K(s,t) = s + 2r$ we have

$$\mu_1 = \mu_2 = 2/\sqrt{3}, \quad \mu_3 = \mu_4 = \ldots = 0.$$
$$u_1(s) = 1/\sqrt{2}, \quad u_2(s) = \sqrt{3/2}\, s$$
$$v_1(t) = \sqrt{3/2}\, t, \quad v_2(t) = 1/\sqrt{2}.$$

A solution exists only if

$$g \in \text{range}(K) = \text{span}\{u_1, u_2\},$$

i.e., if g is of the form

$$g(s) = c_1 + c_2\, s.$$

Another Example of SVE

Consider the equation $\int_0^1 K(s,t)\, f(t)\, dt = g(s)$, $0 \le s \le 1$ with

$$K(s,t) = \begin{cases} s(t-1), & s < t \\ t(s-1), & s \ge t. \end{cases}$$

An alternative expression for the kernel:

$$K(s,t) = -\frac{2}{\pi^2} \sum_{i=1}^{\infty} \frac{\sin(i\,\pi\, s)\, \sin(i\,\pi\, t)}{i^2}\ .$$

From this we get, for $i = 1, 2, \ldots$

$$\mu_i = \frac{1}{(i\pi)^2}, \quad u_i(s) = \sqrt{2}\sin(i\pi s), \quad v_i(t) = -\sqrt{2}\sin(i\pi t).$$

Analytic SVEs are Rare

A few cases where analytic SVEs are available, e.g., the Radon transform.

But in most applications we must use numerical methods for analysis and solution of the integral equation.

The rest of these lectures are devoted to numerical methods!

Our analysis has given us an understanding to the difficulties we are facing – and they will manifest themselves again in any numerical approach we're using to solve the integral equation.

Discretization Methods

Must replace the problem of computing the unknown function f with a discrete problem that we can solve on a computer.

Linear integral equation ⇒ system of linear algebraic equations.

Quadrature Methods.

Compute approximations $\tilde{f}_j = \tilde{f}(t_j)$ to the solution f at the abscissas $t_1, t_2, \ldots, t_n$.

Expansions Methods.

Compute an approximation of the form

$$\tilde{f}(t) = \sum_{j=1}^{n} \alpha_j\, \phi_j(t),$$

where $\phi_1(t), \ldots, \phi_n(t)$ are expansion/basis functions.

Quadrature Discretization

Recall the quadrature rule

$$\int_0^1 \varphi(t)\,dt = \sum_{j=1}^{n} w_j\,\varphi(t_j) + R_n\ ,$$

where R_n is the quadrature error, and

$$w_j = \text{weights}\ , \quad t_j = \text{abscissas}\ , \qquad j = 1, \ldots, n\ .$$

Now apply this rule *formally* to the integral,

$$\Psi(s) = \int_0^1 K(s,t)\,f(t)\,dt = \sum_{j=1}^{n} w_j\,K(s,t_j)\,f(t_j) + R_n(s)\ .$$

Quadrature Discretization + Collocation

Now enforce the collocation requirement that Ψ equals the right-hand side g at n selected points:

$$\Psi(s_i) = g(s_i)\ , \qquad i = 1, \ldots, n\ ,$$

where $g(s_i)$ are sampled/measured values of the function g.

Must neglect the error term $R_n(s)$, and thus replace $f(t_j)$ by $\tilde{f}_j$:

$$\sum_{j=1}^{n} w_j\,K(s_i,t_j)\,\tilde{f}_j = g(s_i), \quad i = 1, \ldots, n\ .$$

Could use $m > n$ collocation points $\rightarrow$ overdetermined system.

The Discrete Problem in Matrix Form

Write out the last equation to obtain

$$\begin{pmatrix} w_1K(s_1,t_1) & w_2K(s_1,t_2) & \cdots & w_nK(s_1,t_n) \\ w_1K(s_2,t_1) & w_2K(s_2,t_2) & \cdots & w_nK(s_2,t_n) \\ \vdots & \vdots & & \vdots \\ w_1K(s_n,t_1) & w_2K(s_n,t_2) & \cdots & w_nK(s_n,t_n) \end{pmatrix} \begin{pmatrix} \tilde{f}_1 \\ \tilde{f}_2 \\ \vdots \\ \tilde{f}_n \end{pmatrix} = \begin{pmatrix} g(s_1) \\ g(s_2) \\ \vdots \\ g(s_n) \end{pmatrix}$$

or simply

$$\boxed{A\,x = b}$$

where A is $n \times n$ with

$$\left.\begin{array}{l} a_{ij} = w_j\,K(s_i,t_j) \\ x_j = \tilde{f}(t_j) \\ b_i = g(s_i) \end{array}\right\} \qquad i,j = 1, \ldots, n\ .$$

Discretization: the Galerkin Method

Select two sets of functions ϕ_i and ψ_j, and write

$$f(t) = \tilde{f}(t) + R_f(t), \qquad \tilde{f}(t) \in \text{span}\{\phi_1, \ldots, \phi_n\}$$

$$g(s) = \tilde{g}(s) + R_g(s), \qquad \tilde{g}(s) \in \text{span}\{\psi_1, \ldots, \psi_n\}\ .$$

Write $\tilde{f}$ as the expansion

$$\tilde{f}(t) = \sum_{j=1}^{n} \alpha_j\,\phi_j(t)\ .$$

and determine the coefficients α_j such that

$$\int_0^1 K(s,t)\,\tilde{f}(t)\,dt = \tilde{g}(s)\ .$$

This ensures that the residual $R_g(s) = g(s) - \int_0^1 K(s,t)\tilde{f}(t)dt$ is orthogonal to $\text{span}\{\psi_1, \ldots, \psi_n\}$.

Computation of the Galerkin Solution

The orthogonality is enforced by requiring that, for $i = 1, \ldots, n$,

$$(\psi_i \, , \, R_g) = 0 \qquad \Leftrightarrow$$

$$(\psi_i \, , \, g) = \left(\psi_i \, , \, \int_0^1 K(s,t) \, \tilde{f}(t) \, dt \right) .$$

Inserting the expansion for $\tilde{f}$, we obtain the $n \times n$ system

$$\boxed{A \, x = b}$$

with $x_i = \alpha_i$ and

$$\begin{aligned} a_{ij} &= \int_0^1 \int_0^1 \psi_i(s) \, K(s,t) \, \phi_j(t) \, ds \, dt \\ b_i &= \int_0^1 \psi_i(s) \, g(s) \, ds \; . \end{aligned}$$

The integrals are often evaluated numerically.

Contents of The Fifth Lecture

Summary of Tikhonov Regularization

Focus on Tikhonov regularization; the ideas carry over to many other methods.

Recall that the Tikhonov solution x_λ solves the problem

$$\min \left\{ \|A \, x - b\|_2^2 + \lambda^2 \|L \, x \, \|_2^2 \right\} ,$$

and that it is formally given by

$$x_\lambda = (A^T A + \lambda^2 L^T L)^{-1} A^T b = A_\lambda^\# b,$$

where $A_\lambda^\# = (A^T A + \lambda^2 L^T L)^{-1} A^T$ is a "regularized inverse."

Perspectives on Regularization

Problem formulation: balance fit (residual) and size of solution.

$$x_\lambda = \arg\min \left\{ \|A\,x - b\|_2^2 + \lambda^2 \|L\,x\|_2^2 \right\}$$

Cannot be used for choosing λ.

Backward error: balance regularization and perturbation errors.

$$\begin{aligned} x^{\text{exact}} - x_\lambda &= x^{\text{exact}} - A_\lambda^{\#}(b^{\text{exact}} + e) \\ &= \left(I - A_\lambda^{\#} A\right) x^{\text{exact}} - A_\lambda^{\#} e \ . \end{aligned}$$

Forward/prediction error: balance residual and perturbation.

$$\begin{aligned} b^{\text{exact}} - A\,x_\lambda &= A\,(x^{\text{exact}} - x_\lambda) \\ &= \left(I - A\,A_\lambda^{\#}\right) b^{\text{exact}} - A\,A_\lambda^{\#} e \ . \end{aligned}$$

Classical and Pragmatic Parameter-Choice

Assume we are given the problem $A\,x = b$ with $b = b^{\text{exact}} + e$, and that we have a strategy for choosing the regularization parameter λ as a function of the "noise level" $\|e\|_2$.

Then *classical* parameter-choice analysis is concerned with the convergence rates of

$$x_\lambda \to x^{\text{exact}} \quad \text{as} \quad \|e\|_2 \to 0 \quad \text{and} \quad \lambda \to 0 \ .$$

The typical situation in practice:

- The norm $\|e\|_2$ is not known, and
- the errors are fixed (not practical to repeat the measurements).

The *pragmatic* approach to choosing the regularization parameter is based on the forward/prediction error, or the backward error.

The Discrepancy Principle

For a continuous parameter λ: solve

$$\|A\,x_\lambda - b\|_2 = \delta_e \ , \quad \text{where} \qquad \|e\|_2 \le \delta_e \ .$$

If the noise is white then $\mathcal{E}(\|e\|_2^2) = m\,\sigma_0^2$.

For a discrete parameter k: choose the smallest k such that

$$\|A\,x_k - b\|_2 \le \delta_e \ .$$

Can show that the convergence rate for the discrepancy principle is

$$\|x_\lambda - x^{\text{exact}}\| = \mathcal{O}\left(\|e\|^{1/2}\right) \qquad \text{(not optimal)}.$$

Optimal convergence is achieved by the rule

$$2\lambda^6\, b^T (A\,A^T + \lambda^2 I_m)^{-3} b = \gamma\,\delta_e^2, \qquad \gamma \ge 2.$$

The Compensated Discrepancy Principle

An estimate of $\|e\|_2^2$ may be a valid estimate for the residual norm $\|A\,x_\lambda - b\|_2$.

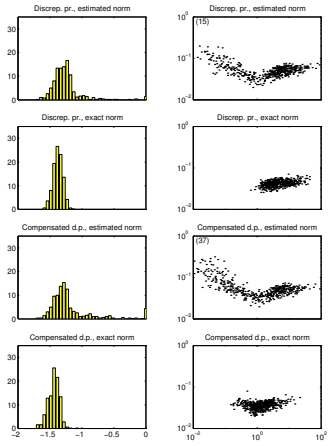
Write $x_\lambda = A_\lambda^{\#} b$ and assume $\text{Cov}(b) = \eta^2 I$; choose the λ that solves

$$\|A\,x_\lambda - b\|_2 = \left(\|e\|_2^2 - \eta^2\,\text{trace}(A\,A^{\#})\right)^{1/2} \ .$$

Note that the right-hand side now depends on λ.

Both the classical and the compensated discrepancy principle are very sensitive to the estimate used for $\|e\|_2$.

Ordinary/Compensated Discrepancy Principle



(Ordinary) Cross-Validation

Leave-one-out approach:
skip ith element b_i and predict this element.

$$\begin{aligned} A^{(i)} &= A([1\!:\!i-1, i+1\!:\!m], :) \\ b^{(i)} &= b([1\!:\!i-1, i+1\!:\!m]) \\ x_\lambda^{(i)} &= \left(A^{(i)}\right)_\lambda^{\#} b^{(i)} \\ b_i^{\text{predict}} &= A(i, :)\, x_\lambda^{(i)}. \end{aligned}$$

The optimal λ minimizes the quantity

$$V_o = \sum_{i=1}^{m} \left(b_i - b_i^{\text{predict}}\right)^2 \ .$$

But λ depends on the ordering of the data.

Generalized Cross-Validation

Want a scheme for which λ is independent of any orthogonal transformation of b (incl. a permutation of the elements).

Results in a method that seeks to minimize the prediction error:

$$\|A\, x_\lambda - b^{\text{exact}}\|_2 \ .$$

Grace Wahba has shown that if the noise is white then

$$\lambda_{\text{GCV}} = \lambda_{\text{opt}}(1 + o(1)),$$

where λ_{opt} mimimizes the predictive mean-square error and $o(1) \to 0$ as $m \to \infty$.

Practical GCV

Minimize the GCV function

$$\mathcal{G}(\lambda) = \frac{\|A\, x_\lambda - b\|_2^2}{\text{trace}(I_m - A\, A_\lambda^{\#})^2} = \frac{\mathcal{V}(\lambda)}{\mathcal{T}(\lambda)}$$

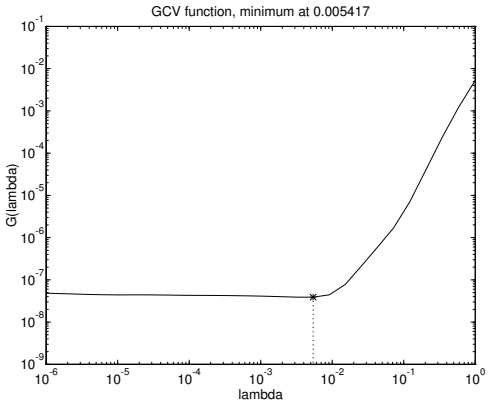
where

$$\begin{aligned} \mathcal{V}(\lambda) &= \frac{\|A\, x_\lambda - b\|_2^2}{\mathcal{T}(\lambda)} \\ \mathcal{T}(\lambda) &= \text{trace}(I_m - A\, A_\lambda^{\#}) = m - \sum_{i=1}^{n} f_i \ . \end{aligned}$$

If $\text{Cov}(b) = \eta^2 I$ then $\mathcal{V}(\lambda)$ levels off at an estimate of η^2.

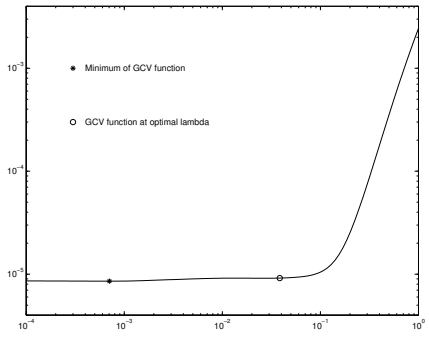
The function $\mathcal{T}(\lambda)$ is a slowly increasing function of λ.

The GCV Function



Occasional Failure

Occasional failure leading to a too small λ; more pronounced for correlated noise.



Parameter-Choice and the L-Curve

Recall that the L-curve basically consists of two parts.

- A "flat" part where the regularization errors dominates.
- A "steep" part where the perturbation error dominates.

The optimal regularization parameter (in the pragmatic sense) must lie somewhere near the L-curve's corner.

Assume for simplicity that $m = n$, and recall that

$$\begin{aligned} \|x_\lambda\|_2^2 &= \sum_{i=1}^{m} \left(\frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \, \frac{u_i^T b}{\sigma_i} \right)^2 \\ \|b - A\, x_\lambda\|_2^2 &= \sum_{i=1}^{n} \left(\frac{\lambda^2}{\sigma_i^2 + \lambda^2} \, u_i^T b \right)^2 . \end{aligned}$$

Also recall that $b = b^{\text{exact}} + e$.

The Flat and Steep Parts

The component b^{exact} dominates when λ is large:

$$\begin{aligned} \|x_\lambda\|_2 &\approx \|x^{\text{exact}}\|_2 \\ \|b - A\, x_\lambda\|_2^2 &\approx \lambda^4 \sum_{i=1}^{n} \left(\frac{u_i^T b}{\sigma_i^2} \right)^2 \end{aligned}$$

The error e dominates when λ is small ($u_i^T e \approx \pm\epsilon_0$):

$$\begin{aligned} \|A_\lambda^{\#} e\|_2^2 &\approx \lambda^{-4} \sum_{i=1}^{n} \left(\sigma_i \, u_i^T e \right)^2 \approx \lambda^{-4} \, \epsilon_0^2 \|A\|_{\text{F}}^2 \\ \|b - A\, A_\lambda^{\#} e\|_2^2 &\approx \epsilon_0^2 \sum_{i=1}^{n} \left(\frac{\lambda^2}{\sigma_i^2 + \lambda^2} \right)^2 \approx n \, \epsilon_0^2 \, . \end{aligned}$$

The Key Idea

The flat and the steep parts of the L-curve represent solutions that are dominated by regularization errors and perturbation errors.

- The balance between these two errors must occur near the L-curve's corner.
- The two parts – and the corner – are emphasized in log-log scale.
- Log-log scale is insensitive to scalings of A and b.

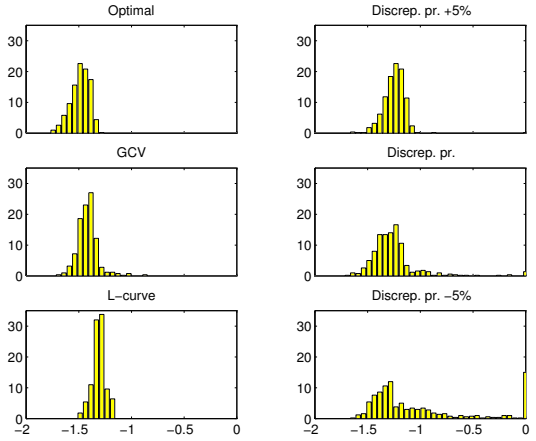
An *operational* definition of the corner is required.

Write the L-curve as

$$(\log \|A\,x_{\mathrm{reg}} - b\|_2\,,\ \log \|L\,x_{\mathrm{reg}}\|_2)$$

and seek the point with maximum curvature.

Comparison of Parameter-Choice Methods:



Conclusions About the Discrepancy Principle

- Using the estimate $\eta\sqrt{m}$ of $\|e\|_2$, both the ordinary and compensated discrepancy principles oversmooth as well as undersmooth. The large number of instances of undersmoothing leads to a large tail in the histogram.
- Using the exact value of $\|e\|_2$, the discrepancy principle consistently oversmooths while the compensated discrepancy principle yields relative errors comparable to GCV.
- Both discrepancy principles are less robust than GCV and the L-curve, except when a very good estimate of $\|e\|_2$ is available.

Conclusions About GCV and the L-Curve

- The GCV method, on the average, leads to a slight oversmoothing which accounts for the increased average error, compared to the optimal results. Occasionally GCV undersmooths, leading to larger errors that constitute the histogram's tail.
- The L-curve criterion consistently oversmooths—there is no $\lambda < \lambda_{\mathrm{opt}}$. Hence, the average error is greater than that for GCV, but the histogram has no tail.
- The L-curve criterion is more *robust* than GCV, in the sense that the L-curve criterion never leads to large errors while GCV occasionally does.

Contents of The Third Lecture

1. Discretized problems
2. Singular value decomposition (SVD)
   (a) Definition
   (b) Analysis
3. The discrete Picard condition
4. SVD $\sim$ SVE
5. Stabilization = reduction of noise

Discrete Problems

The rest of these lectures focus on discretized systems, either square

$$\boxed{x = b}$$

or overdetermined

$$\boxed{\min_x \|A\,x - b\|_2}$$

with a very ill-conditioned coefficient matrix A,

$$\text{cond}(A) \gg 1,$$

arising from the discretization of an ill-posed problem.

Noise model:

$$b = b^{\text{exact}} + e, \qquad \text{where} \qquad b^{\text{exact}} = A\,x^{\text{exact}}$$

and the vector e represents the noise in the data.

Statistical Issues

Let $\text{Cov}(b)$ be the covariance for the right-hand side.

Then the covariance matrix for the (least squares) solution is

$$\text{Cov}(x_{\text{LS}}) = (A^T A)^{-1} A^T\,\text{Cov}(b)\,A\,(A^T A)^{-1}.$$

Unless otherwise stated, we assume for simplicity that b^{exact} and e are uncorrelated, and that

$$\text{Cov}(b) = \text{Cov}(e) = \eta^2 I,$$

then

$$\text{Cov}(x_{\text{LS}}) = \eta^2 (A^T A)^{-1}.$$

$\text{cond}(A) \gg 1 \Rightarrow \text{Cov}(x_{\text{LS}})$ is likely to have very large elements.

The Singular Value Decomposition

Assume that A is $m \times n$ and, for simplicity, also that $m \geq n$:

$$A = U\,\Sigma\,V^T = \sum_{i=1}^{n} u_i\,\sigma_i\,v_i^T$$

where U and V consist of *singular vectors*

$$U = (u_1, \ldots, u_n)\,, \qquad V = (v_1, \ldots, v_n)$$

with $U^T U = V^T V = I_n$, and the *singular values* satisfy

$$\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)\,, \qquad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0\,.$$

Then $\|A\|_2 = \sigma_1$ and $\text{cond}(A) = \|A\|_2\,\|A^\dagger\|_2 = \sigma_1/\sigma_n$.

SVD Software for Dense Matrices

Software package	Subroutine
ACM TOMS	`HYBSVD`
EISPACK	`SVD`
IMSL	`LSVRR`
LAPACK	`_GESVD`
LINPACK	`_SVDC`
NAG	`F02WEF`
Numerical Recipes	`SVDCMP`
Matlab	`svd`

Complexity of SVD algorithms: $\mathcal{O}(m\,n^2)$.

Important SVD Relations

Relations similar to the SVE

$$\left.\begin{array}{rclcrcl} A\,v_i &=& \sigma_i\,u_i & & \|A\,v_i\|_2 &=& \sigma_i \\ A^T u_i &=& \sigma_i\,v_i & & \|A^T u_i\|_2 &=& \sigma_i \end{array}\right\} \quad i = 1, \ldots, n.$$

These equations are related to the (least squares) solution:

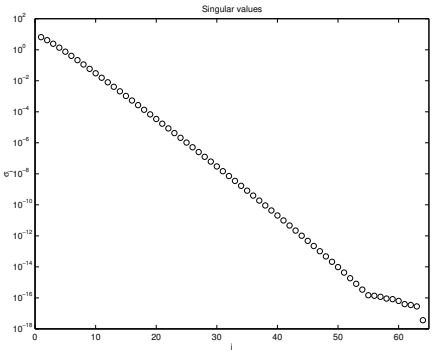
$$\begin{aligned} x &= \sum_{i=1}^{n} (v_i^T x)\,v_i \\ A\,x &= \sum_{i=1}^{n} \sigma_i\,(v_i^T x)\,u_i\ , \quad b = \sum_{i=1}^{n} (u_i^T b)\,u_i \\ A^{-1} b &= \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i}\,v_i\ . \end{aligned}$$

What the SVD Looks Like

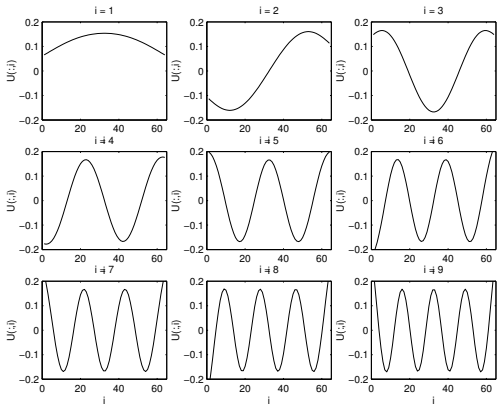
The following figures show the SVD of the 64×64 matrix A from the model problem, computed by means of `csvd`:

```
>> help svd
 SVD    Singular value decomposition.
    [U,S,V] = SVD(X) produces a diagonal matrix S, of the same
    dimension as X and with nonnegative diagonal elements in
    decreasing order, and unitary matrices U and V so that
    X = U*S*V'.

    S = SVD(X) returns a vector containing the singular values.

    [U,S,V] = SVD(X,0) produces the "economy size"
    decomposition. If X is m-by-n with m > n, then only the
    first n columns of U are computed and S is n-by-n.
```


The Singular Values



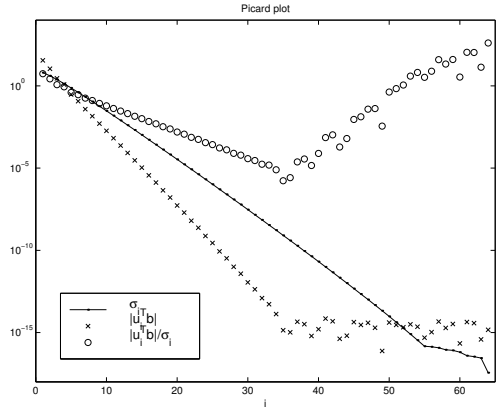
The Left and Right Singular Vectors



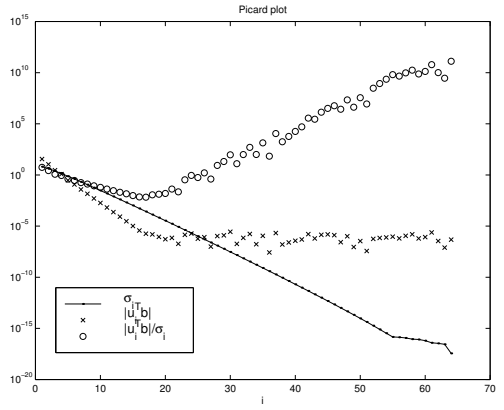
Some Observations

- The singular values decay gradually to zero.
- No gap in the singular value spectrum.
- Condition number cond(A) = "∞."
- Singular vectors have more oscillations as i increases.
- In this problem, # sign changes $= i - 1$.

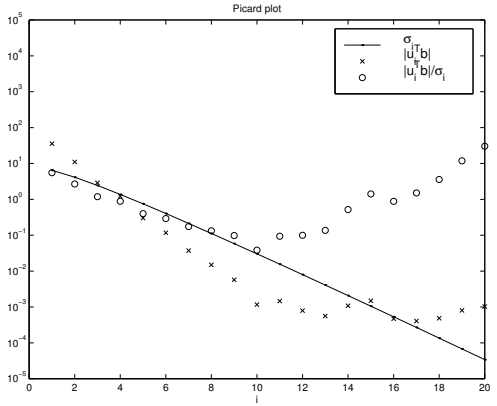
The Discrete Picard Plot



Discrete Picard Plot with Noise



Discrete Picard Plot – More Noise



The Discrete Picard Condition

The relative decay of the singular values σ_i and the right-hand side's SVD coefficients $u_i^T b$ plays a major role!

The Discrete Picard Condition. Let τ_A denote the level at which the singular values of A level off. Then the discrete Picard condition is satisfied if, for all singular values $\sigma_i > \tau_A$, the corresponding coefficients $|u_i^T b^{\text{exact}}|$, on the average, decay to zero faster than the σ_i.

Can base the analysis on the moving geometric mean

$$\rho_i = \sigma_i^{-1} \left(\prod_{j=i-q}^{i+q} |u_i^T b| \right)^{1/(2q+1)} , \quad i = 1+q, \ldots, n-q \; .$$

Computation of the SVE

Based on the Galerkin method with orthonormal ϕ_i and ψ_j.

1. Discretize K to obtain $n \times n$ matrix A, and compute its SVD.
2. Then $\sigma_j^{(n)} \to \mu_j$ as $n \to \infty$.
3. Define the functions

$$\tilde{u}_j(s) = \sum_{i=1}^{n} u_{ij} \, \psi_i(s) \; , \qquad j = 1, \ldots, n$$

$$\tilde{v}_j(t) = \sum_{i=1}^{n} v_{ij} \, \phi_i(t) \; , \qquad j = 1, \ldots, n \; .$$

Then $\tilde{u}_j(s) \to u_j(s)$ and $\tilde{v}_j(t) \to v_j(t)$ as $n \to \infty$.

More Precise Results

Let

$$\|K\|_2^2 \equiv \int_0^1 \int_0^1 |K(s,t)|^2 \, ds \, dt \; , \qquad \delta_n^2 \equiv \|K\|_2^2 - \|A\|_{\text{F}}^2 \; .$$

Then for $i = 1, \ldots, n$

$$0 \le \mu_i - \sigma_i^{(n)} \le \delta_n$$

$$\sigma_i^{(n)} \le \sigma_i^{(n+1)} \le \mu_i$$

$$\max \{ \|u_i - \tilde{u}_i\|_2 \, , \, \|v_i - \tilde{v}_i\|_2 \} \le \left(\frac{2 \, \delta_n}{\mu_i - \mu_{i+1}} \right)^{1/2} \; .$$

Stabilization = Reduction of Noise

Recall that both the (least squares) solution is given by

$$x = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i}\, v_i.$$

Must get rid of the "noisy" SVD components. Note that

$$u_i^T b = u_i^T b^{\text{exact}} + u_i^T e \approx \begin{cases} u_i^T b^{\text{exact}}, & |u_i^T b^{\text{exact}}| > |u_i^T e| \\ u_i^T e, & |u_i^T b^{\text{exact}}| < |u_i^T e|. \end{cases}$$

Hence, due to the DPC:

- "noisy" SVD components are those for which $|u_i^T b^{\text{exact}}|$ is small,
- and therefore they correspond to the smaller singular values σ_i.

Truncated SVD

A simple way to reduce the influence of the noise is to discard the SVD coefficients corresponding to the smallest singular values.

Define truncated SVD (TSVD) solution

$$x_k = \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i}\, v_i, \qquad k < n.$$

Can show that if $\text{Cov}(b) = \eta^2 I$ then

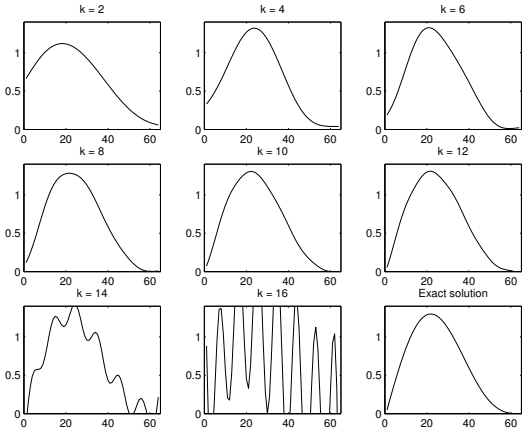
$$\text{Cov}(x_k) = \eta^2 \sum_{i=1}^{k} v_i\, v_i^T$$

and thus we can expect that both

$$\|x_k\|_2 \ll \|x\|_2 \qquad \text{and} \qquad \|\text{Cov}(x_k)\|_2 \ll \|\text{Cov}(x)\|_2.$$

The prize we pay for smaller covariance is *bias:* $\mathcal{E}(x_k) \neq \mathcal{E}(x_{\text{LS}})$.

Truncated SVD Solutions



The Truncation Parameter

Note: the truncation parameter k in

$$x_k = \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i}\, v_i$$

is dictated by the coefficients $u_i^T b$, not the singular values!

Basically we should choose k as the index i where $|u_i^T b|$ start to "level off" due to the noise.

Contents of The Fourth Lecture

1. Tikhonov regularization
   (a) Definition
   (b) The smoothing norm
   (c) Implementation
2. Perturbation results
3. L-curve analysis

Regularization

Regularization = stabilization: how to deal with solution components corresponding to the small singular values.

Most approaches involve the residual norm

$$\rho(f) = \left\| \int_0^1 K(s,t)\, f(t)\, dt - g(s) \right\| ,$$

and a *smoothing norm* $\omega(f)$ that measure the "size" of the solution f. Example of a common choice:

$$\omega(f)^2 = \int_0^1 |f^{(p)}(t)|^2\, dt$$

1. Minimize $\rho(f)$ s.t. $\omega(f) \leq \delta$.
2. Minimize $\omega(f)$ s.t. $\rho(f) \leq \alpha$.
3. Tikhonov: $\min \left\{ \rho(f)^2 + \lambda^2\, \omega(f)^2 \right\}$.

Discrete Tikhonov Regularization

Replace the continuous problem with a linear algebra problem.

Minimization of the residual ρ is replaced by

$$\min \|A\, x - b\|_2 , \qquad A \in R^{m \times n} ,$$

where A and b are obtained by discretization of the integral equation.

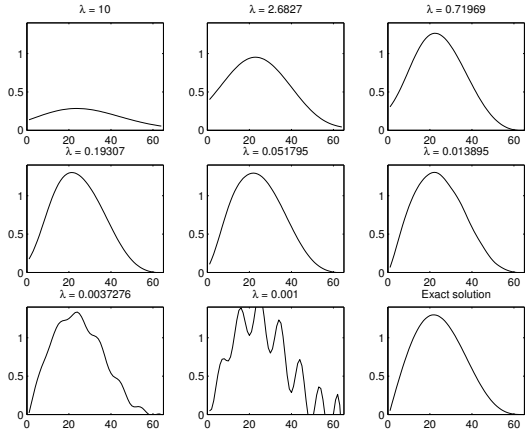
Must also discretize the smoothing norm

$$\Omega(x) \approx \omega(f).$$

The resulting discrete Tikhonov problem is

$$\min \left\{ \|A\, x - b\|_2^2 + \lambda^2\, \Omega(x)^2 \right\} .$$

Tikhonov Solutions



The Matrix L

Often $\Omega(x)$ can be written as

$$\Omega(x) = \|L\,x\|_2 \qquad \text{or} \qquad \Omega(x) = \|L\,(x - x^*)\|_2 \ ,$$

where L approximates a derivative operator.

Examples of the 1. and 2. derivative operator on a regular mesh

$$L_1 \;=\; \begin{pmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{pmatrix} \in R^{(n-1)\times n}$$

$$L_2 \;=\; \begin{pmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{pmatrix} \in R^{(n-2)\times n}$$

(we have omitted factors n^{-1} and n^{-2} that are "absorbed" in λ).

Sobolev Norms

Combinations of several derivative operators

$$\Omega(x)^2 = \sum_{i=0}^{q} \alpha_i^2 \, \|L_i\,(x - x^*)\|_2^2 \ .$$

Numerical treatment: compute the QR factorization of the "stacked" L_i matrices

$$\begin{pmatrix} \alpha_q\,L_q \\ \vdots \\ \alpha_0\,L_0 \end{pmatrix} = Q\,R$$

and use $\|R\,x\|_2$ instead of $\lambda^2\,\|L\,x\|_2$.

Inequality Constraints

Three important constraints to the solution: nonnegativity, monotonicity, convexity.

All three can be put in the general form $G\,x \geq 0$:

$$\begin{aligned} x &\geq 0 \qquad \text{(nonnegativity)} \\ L_1\,x &\geq 0 \qquad \text{(monotonicity)} \\ L_2\,x &\geq 0 \qquad \text{(convexity)} \end{aligned}$$

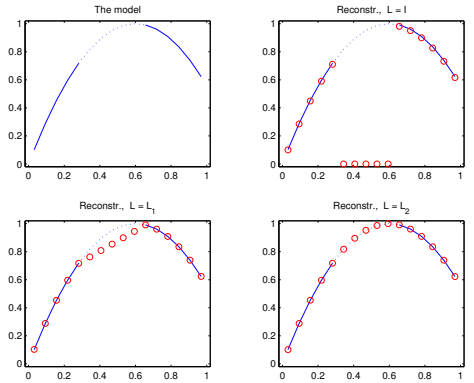
where L_1 and L_2 approximate the first and second derivative operators, respectively.

The resulting least squares problem is

$$\min \left\| \begin{pmatrix} A \\ \lambda L \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2 \quad \text{subject to} \quad G\,x \geq 0 \ .$$

The Need for Higher-Order Smoothing

A simple example with missing data.



SVD and Discrete Tikhonov Regularization

Assume – for simplicity – that we use Tikhonov regularization in standard form $\Rightarrow L = I_n$.

Then we can write the discrete Tikhonov solution x_λ in terms of the SVD of A

$$x_\lambda = \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \frac{u_i^T b}{\sigma_i} v_i.$$

The *filter factors* are given by

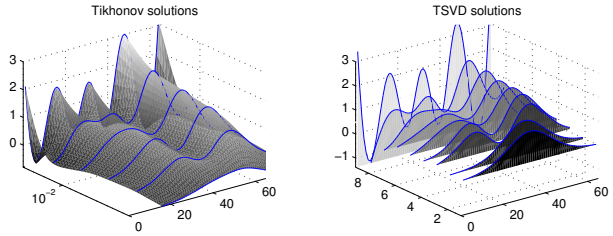
$$f_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \ ,$$

and their purpose is to dampen the components in the solution corresponding to small σ_i.

TSVD and Tikhonov Regularization

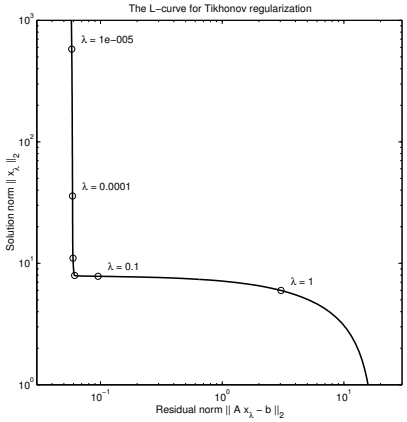
TSVD and Tikhonov solutions are both filtered SVD expansions.

The regularization parameter is either k or λ.



For each k, there exists a λ such that $x_\lambda \approx x_k$.

Solution and Residual Norms



Monotonic Behavior of the Norms

The TSVD solution and residual norms vary monotonically with k

$$\|x_k\|_2^2 = \sum_{i=1}^{k} (u_i^T b/\sigma_i)^2 \leq \|x_{k+1}\|_2^2$$

$$\|A\,x_k - b\|_2^2 = \sum_{i=k+1}^{m} (u_i^T b)^2 \geq \|A\,x_{k+1} - b\|_2^2$$

Similarly, the Tikhonov solution and residual norms cary monotonically with λ:

$$\|x_\lambda\|_2^2 = \sum_{i=1}^{n} \left(f_i \, \frac{u_i^T b}{\sigma_i} \right)^2$$

$$\|A\,x_\lambda - b\|_2^2 = \sum_{i=1}^{n} \left((1 - f_i)\, u_i^T b \right)^2 + \sum_{i=n+1}^{m} (u_i^T b)^2.$$

Efficient Implementation

The original formulation

$$\min \left\{ \|A\,x - b\|_2^2 + \lambda^2 \, \|L\,x\|_2^2 \right\}.$$

Two alternative formulations

$$(A^T A + \lambda^2 L^T L)\, x = A^T b$$

$$\min \left\| \begin{pmatrix} A \\ \lambda\, L \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2$$

The first shows that we have a linear problem. The second shows how to solve it stably:

- treat it as a least squares problem,
- utilize the sparsity.

TSVD Perturbation Bound

Theorem.

Let $b = b^{\text{exact}} + e$ and let x_k and x_k^{exact} denote the TSVD solutions. Then

$$\frac{\|x_k^{\text{exact}} - x_k\|_2}{\|x_k\|_2} \le \frac{\sigma_1}{\sigma_k} \, \frac{\|e\|_2}{\|A\,x_k\|_2}.$$

We see that the condition number for the TSVD solution is σ_1/σ_k.

Can be much smaller than $\text{cond}(A) = \sigma_1/\sigma_n$.

Tikhonov Perturbation Bound

Theorem.

Let $b = b^{\text{exact}} + e$ and let $x_{L,\lambda}^{\text{exact}}$ and $x_{L,\lambda}$ denote the solutions to

$$\min\{\|A\,x - b^{\text{exact}}\|_2^2 + \lambda^2 \|L\,x\|_2^2\} \quad \text{and} \quad \min\{\|A\,x - b\|_2^2 + \lambda^2 \|L\,x\|_2^2\}$$

computed with the *same* λ.

Assume that L^{-1} exists; then

$$\frac{\|x_{L,\lambda}^{\text{exact}} - x_{L,\lambda}\|_2}{\|x_{L,\lambda}\|_2} \le \frac{\kappa_\lambda}{1 - \epsilon\,\kappa_\lambda} \left(\frac{\|e\|_2}{\|A\,x_\lambda\|_2} + \epsilon\,\kappa_\lambda \, \frac{\|b - b_\lambda\|_2}{\|A\,x_\lambda\|_2} \right)$$

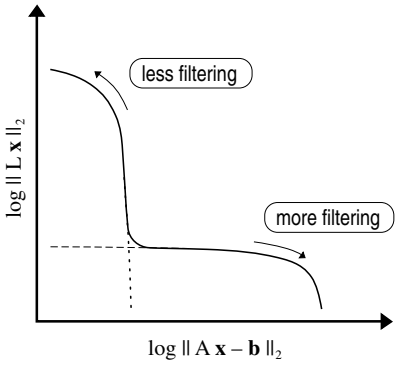
where

$$\kappa_\lambda = \frac{\|A\|_2 \, \|L^{-1}\|_2}{\lambda} \qquad \left(= \frac{\sigma_1}{\lambda} \quad \text{for} \quad L = I \right)$$

is the condition number for the Tikhonov solution.

The L-Curve for Tikhonov Regularization

Plot of $\|L\,x_{L,\lambda}\|_2$ versus $\|A\,x_{L,\lambda} - b\|_2$ in *log-log scale.*



Properties of the L-Curve

The semi-norm $\|L\,x_{L,\lambda}\|_2$ is a monotonically decreasing convex function of the norm $\|A\,x_{L,\lambda} - b\|_2$.

Define $x_{\rm LS}$ = least squares solution and

$$\delta_0^2 = \sum_{i=n+1}^{m} (u_i^T b)^2 \qquad \text{(inconsistency measure.)}$$

Then

$$\delta_0 \le \|A\,x_{L,\lambda} - b\|_2 \le \|b\|_2$$
$$0 \le \|L\,x_{L,\lambda}\|_2 \le \|L\,x_{\rm LS}\|_2 \ .$$

Any point (δ, η) on the L-curve is a solution to the following two inequality-constrained least squares problems:

$$\delta = \min \|A\,x - b\|_2 \qquad \text{subject to} \qquad \|L\,x\|_2 \le \eta$$
$$\eta = \min \|L\,x\|_2 \qquad \text{subject to} \qquad \|A\,x - b\|_2 \le \delta \ .$$

The L-Shaped Appearance of the L-curve

The L-curve has two distinctly different parts.

- The horizontal part where the regularization errors dominate.
- The vertical part where the perturbation errors dominate.

The log-log scale emphasizes the two different parts.

The "corner" is located approximately at

$$\left(\|A\,x_{L,\lambda} - b\|_2\,,\ \|L\,x_{L,\lambda}\|_2\right) \approx \left(\sqrt{\sigma_0^2(m-n+p)}\,,\ \|L\,x^{\rm exact}\|_2\right)$$

The Curvature of the L-Curve

Want to derive an analytical expression for the L-curve's curvature κ in log-log scale.

Consider $L = I$ and define

$$\eta = \|x_\lambda\|_2^2 \ , \qquad \rho = \|A\,x_\lambda - b\|_2^2$$

and

$$\hat\eta = \log\eta \ , \qquad \hat\rho = \log\rho \ .$$

Then the curvature is given by

$$\kappa = 2\,\frac{\hat\rho'\hat\eta'' - \hat\rho''\hat\eta'}{((\hat\rho')^2 + (\hat\eta')^2)^{3/2}} \ .$$

This can be used to define the "corner" of the L-curve as the point with maximum curvature.

A More Practical Formula

The first derivatives of $\hat\eta$ and $\hat\rho$ are *logarithmic derivatives*,

$$\hat\eta' = \eta'/\eta \ , \qquad \hat\rho' = \rho'/\rho$$

and it can be shown that

$$\rho' = -\lambda^2\eta' \ .$$

The second derivatives satisfy the relation

$$\rho'' = \frac{d}{d\lambda}\left(-\lambda^2\eta'\right) = -2\,\lambda\,\eta' - \lambda^2\eta'' \ .$$

When all this is inserted into the equation for κ, we get

$$\kappa = 2\,\frac{\eta\,\rho}{\eta'}\,\frac{\lambda^2\eta'\rho + 2\,\lambda\,\eta\,\rho + \lambda^4\eta\,\eta'}{(\lambda^2\eta^2 + \rho^2)^{3/2}} \ .$$

Efficient Computation of the Curvature

The quantities η and ρ readily available.

Straightforward to show that

$$\eta' = \frac{4}{\lambda}\, x_\lambda^T z_\lambda$$

where z_λ is given by

$$z_\lambda = \left(A^T A + \lambda^2 I\right)^{-1} A^T (A\, x_\lambda - b) \ ,$$

i.e., z_λ is the solution to the problem

$$\min \left\| \begin{pmatrix} A \\ \lambda\, I \end{pmatrix} z - \begin{pmatrix} A\, x_\lambda - b \\ 0 \end{pmatrix} \right\|_2 \ .$$

This can be used to compute z_λ efficiently.

Contents of The Sixth Lecture

- Why iterative methods?
- Landweber iteration
- ART
- Regularizing CGLS iterations
- Toeplitz structure
- Fast Toeplitz matrix multiplication

Advantages of Iterative Methods

- The matrix A is never altered, only "touched" via matrix-vector multiplications $A\, x$ and $A^T y$.
- The matrix A is not explicitly required – we only need a "black box" that computes the action of A or the underlying operator.
- May produce a natural sequence of regularized solutions; stop when the solution is "satisfactory" (parameter choice).
- Atomic operations are easy to parallelize.

Iterative Methods

1. Iterative solution of a regularized problem, such as Tikhonov

$$\left(A^T A + \lambda^2 L^T L\right) x = A^T b \ .$$

   Challenge: to construct a good preconditioner!

2. Iterate on the unregularized system, e.g., on

$$A^T A\, x = A^T b$$

   and use the iteration number as the regularization parameter.

The latter approach relies on *semi-convergence:*

- initial convergence towards x^{exact},
- followed by (slow) convergence to $x_{\text{LS}} = A^\dagger b$.

Must stop at the end of the first stage!

Working With Seminorms

No problem to work with seminorm $\|L\,x\|_2$. Assume that

$$\bar{x}^{(k)} = \mathcal{P}_k \left(\bar{A}^T \bar{A}\right) \bar{A}^T \bar{b}$$

where $\bar{A}$ and $\bar{b}$ are the standard-form quantities:

$$\bar{A} = A\, L_A^\dagger \qquad \bar{b} = b - A\, x_0 \ .$$

Inserting this, we obtain

$$\bar{x}^{(k)} = \mathcal{P}_k \left((L_A^\dagger)^T A^T A\, (L_A^\dagger)\right) (L_A^\dagger)^T A^T (b - A\, x_0)$$

and, using $x^{(k)} = L_A^\dagger \bar{x}^{(k)}$,

$$x^{(k)} = \mathcal{P}_k \left(L_A^\dagger (L_A^\dagger)^T A^T A\right) L_A^\dagger (L_A^\dagger)^T A^T b + x_0$$

Hence, the matrix $L_A^\dagger (L_A^\dagger)^T$ works as a "right preconditioner."

Landweber Iteration

Richardson, Landweber, Fridman, Picard, Cimino, ...

$$x^{(k)} = x^{(k-1)} + \omega\, A^T r^{(k-1)} \ , \qquad k = 1, 2, \ldots$$

where $r^{(k)} = b - A\, x^{(k)}$ and $0 < \omega < 2\, \|A^T A\|_2^{-1} = 2\, \sigma_1^{-2}$.

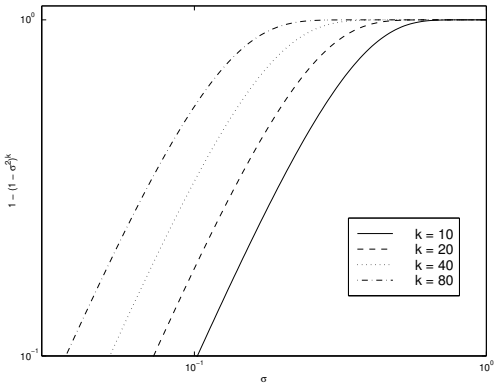
Generalization by Strand

$$x^{(k)} = x^{(k-1)} + \mathcal{F}(A^T A)\, A^T r^{(k-1)} \ , \qquad k = 1, 2, \ldots$$

where $\mathcal{F}$ is a rational function of $A^T A$. SVD analysis:

$$f_i^{(k)} = 1 - (1 - \omega\, \sigma_i^2)^k \qquad \text{ordinary method}$$
$$f_i^{(k)} = 1 - \left(1 - \sigma_i^2\, \mathcal{F}(\sigma_i^2)\right)^k \qquad \text{generalized method.}$$

Landweber Filter Factors



ART

... and the title to the best acronym goes to ...

Kaczmarz's method = algebraic reconstruction technique (ART):

$$x \leftarrow x + \frac{b_i - a_i^T x}{\|a_i\|_2^2}\, a_i \ , \qquad i = 1, \ldots, m \ ,$$

where b_i is the ith component b.

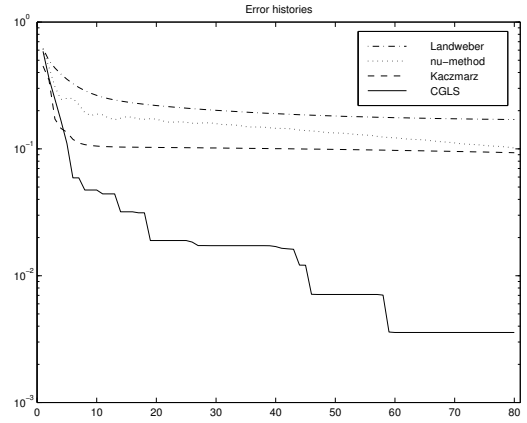
Mathematically equivalent to Gauss-Seidel's method for the problem

$$x = A^T y \ , \qquad A\, A^T y = b \ .$$

Used successfully in computerized tomography.

In general: fast initial convergence, then slow.

Slow Convergence



Regularizing CGLS Iterations

CGLS: CG applied to the normal equations $A^T A\, x = A^T b$:

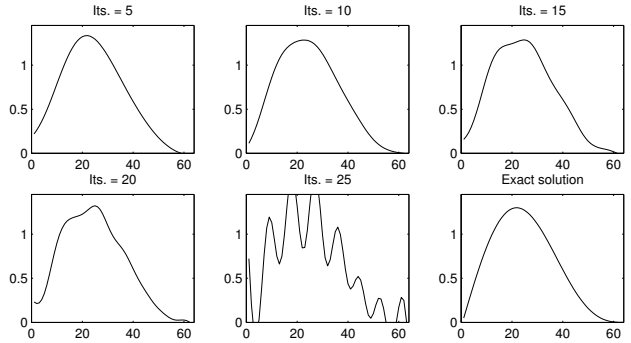
$$\begin{aligned}
\alpha_k &= \|A^T r^{(k-1)}\|_2^2 \,/\, \|A\, d^{(k-1)}\|_2^2 \\
x^{(k)} &= x^{(k-1)} + \alpha_k\, d^{(k-1)} \\
r^{(k)} &= r^{(k-1)} - \alpha_k\, A\, d^{(k-1)} \\
\beta_k &= \|A^T r^{(k)}\|_2^2 \,/\, \|A^T r^{(k-1)}\|_2^2 \\
d^{(k)} &= A^T r^{(k)} + \beta_k\, d^{(k-1)}
\end{aligned}$$

where $r^{(k)} = b - A\, x^{(k)}$ = residual vector, and $d^{(k)}$ search direction.

Initialization:

$$\begin{aligned}
x^{(0)} &= \text{starting vector; often zero)} \\
r^{(0)} &= b - A\, x^{(0)} \\
d^{(0)} &= A^T r^{(0)} .
\end{aligned}$$

CGLS Solutions



Return to Deconvolution Problems

For deconvolution problems, the kernel satisfies

$$K(s,t) = h(s-t)$$

and the matrix elements in a quadrature discretization become

$$a_{ij} = w_j\, h(s_i - t_j), \qquad i,j = 1,\ldots,n.$$

Hence, $A = H\,W$ where $W = \mathrm{diag}(w_1,\ldots,w_n)$ and

$$h_{ij} = h(s_i - t_j)\ , \qquad i,j = 1,\ldots,n\ .$$

Simple sampling: $s_i = i/n$, $t_j = j/n$ and $W = n^{-1}I$; then

$$a_{ij} = n^{-1}\, h\big((i-j)/n\big), \qquad i,j = 1,\ldots,n.$$

Note that $a_{i+1,j+1} = a_{ij} \rightarrow$ structure!

Toeplitz Matrices

When the matrix elements only depend on $i-j$:

$$t_{ij} = t_{i+\ell,j+\ell} = t_{i-j} \qquad \text{for all relevant } i,\ j,\text{ and } \ell.$$

What the Toeplitz matrix looks like:

$$T = \begin{pmatrix} t_0 & t_{-1} & t_{-2} & \cdots & t_{1-n} \\ t_1 & t_0 & t_{-1} & \cdots & t_{2-n} \\ t_2 & t_1 & t_0 & \cdots & t_{3-n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{n-1} & t_{n-2} & t_{n-3} & \cdots & t_0 \end{pmatrix}$$

Can utilize this structure in various ways ...

Toeplitz Matrices and Persymmetry

Persymmetry – symmetry across the anti-diagonal:

$$t_{ij} = t_{n-j+1,n-i+1} \qquad \text{for all relevant } i \text{ and } j.$$

Matrix notation: define the exchange matrix

$$J = \begin{pmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{pmatrix},$$

then persymmetry is expressed as $T\,J = (T\,J)^T = J\,T^T$.

Now use $J^2 = I \Leftrightarrow J^{-1} = J$ to show that $T = J\,T^T J$, which implies that the inverse is also persymmetric:

$$T^{-1} = (J\,T^T J)^{-1} = J^{-1}(T^{-1})^T J^{-1} = J\,(T^{-1})^T J\ .$$

SVD of a Toeplitz Matrix

Thus the SVD of T is given by

$$T = \sum_{ji=1}^{n} u_i\,\gamma_i\lambda_i\,(\gamma_i J i_j)^T$$

where (λ_i, u_i) = eigenpairs of $T\,J$ (symmetric), and $\gamma_i = \pm 1$ is chosen to make $\gamma_i\lambda_i$ positive.

Except perhaps for a sign change, the vector v_j is identical to u_j with its elements in reverse order.

If T is *symmetric* then $u_j = \widehat{\gamma}_j v_j$, with $\widehat{\gamma} = \pm 1$, and:

- the left and right singular vectors are identical, except perhaps for a sign change;
- the sequence of elements in each vector is symmetric around the middle, except perhaps for a sign change.

Numerical Example

$$T = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

$$U = \begin{pmatrix} .628 & .707 & -.325 \\ .460 & 0 & .888 \\ .628 & -.707 & -.325 \end{pmatrix}, \qquad V = \begin{pmatrix} .628 & -.707 & .325 \\ .460 & 0 & -.888 \\ .628 & .707 & .325 \end{pmatrix}$$

$$u_1 = J\,u_1 = v_1 = J\,v_1$$

$$u_2 = -J\,u_2 = -v_2 = J\,v_2$$

$$u_3 = J\,u_3 = -v_3 = -J\,v_3$$

Convolution

Define the circulant matrix

$$C_h = \begin{pmatrix} h_0 & h_{n-1} & h_{n-2} & \cdots & h_1 \\ h_1 & h_0 & h_{n-1} & \cdots & h_2 \\ h_2 & h_1 & h_0 & \cdots & h_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{n-1} & h_{n-2} & h_{n-3} & \cdots & h_0 \end{pmatrix},$$

Then it is easy to see that

$$g = h * f = \sum_{j=0}^{N-1} f_j\, h_{(i-j) \bmod N} = C_h\, f\ ,$$

where $*$ denotes cyclic convolution of the two vectors h and F.

We can compute $h * f$ in about $2.5\, n\, \log_2(n)$ flops via the FFT.

Convolution and DFT

Can write the DFT as (where $\hat{\imath}$ = imaginary unit):

$$\mathrm{DFT}(f) = F_n\, f, \qquad (F_n)_{ij} = (\exp(-2\pi\hat{\imath}/n))^{(i-1)(j-1)}\ .$$

Thus

$$\begin{aligned} g &= \mathrm{IDFT}\,(\mathrm{DFT}(h) \odot \mathrm{DFT}(f)) \\ &= F_n^{-1}\,((F_n\, h) \odot (F_n\, f)) \\ &= F_n^{-1}(\mathrm{diag}(F_n\, h)\, F_n\, f) \\ &= F_n^{-1}\mathrm{diag}(F_n\, h)\, F_n\, f. \end{aligned}$$

We conclude that

$$C_h = F_n^{-1}\mathrm{diag}(F_n\, h)\, F_n$$

is the *eigenvalue decomposition* of C_h.

Imbedding Toeplitz Matrix in Circulant Matrix

We can always embed a Toeplitz matrix in a circulant matrix:

$$C = \begin{pmatrix} T & \hat{T} \\ \hat{T} & T \end{pmatrix} = \begin{pmatrix} t_0 & t_{-1} & t_{-2} & t_{-3} & 0 & t_3 & t_2 & t_1 \\ t_1 & t_0 & t_{-1} & t_{-2} & t_{-3} & 0 & t_3 & t_2 \\ t_2 & t_1 & t_0 & t_{-1} & t_{-2} & t_{-3} & 0 & t_3 \\ t_3 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} & t_{-3} & 0 \\ 0 & t_3 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} & t_{-3} \\ t_{-3} & 0 & t_3 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} \\ t_{-2} & t_{-3} & 0 & t_3 & t_2 & t_1 & t_0 & t_{-1} \\ t_{-1} & t_{-2} & t_{-3} & 0 & t_3 & t_2 & t_1 & t_0 \end{pmatrix}$$

such that

$$C \begin{pmatrix} u \\ 0 \end{pmatrix} = \begin{pmatrix} T & \hat{T} \\ \hat{T} & T \end{pmatrix} \begin{pmatrix} u \\ 0 \end{pmatrix} = \begin{pmatrix} T\,u \\ z \end{pmatrix}.$$

FFT-Based Toeplitz Multiplication Algorithm

Assume that T is symmetric and $n \times n$ with n a power of 2.

Also define $t = (t_0, t_1, \ldots, t_{n-1})^T$ and $\hat{t} = (0, t_{n-1}, \ldots, t_1)^T$.

The algorithm for computing $v = T\,u$:

1. $h = \begin{pmatrix} t \\ \hat{t} \end{pmatrix}$
2. $f = \begin{pmatrix} u \\ 0 \end{pmatrix}$
3. $g = h \star f$ via FFT
4. $v = g(1\colon n)$

Flop count: $(2.5 + 5)\,(2n)\,\log_2(2n) \approx 15\,n\,\log_2(n)$ (instead of $2n^2$), once DFT(h) has been computed and stored – assuming $n = 2^p$.

Does It Pay Off?

The answer depends on the structure of the matrix.

If the Toeplitz matrix is full, the answer is "yes."

If the Toeplitz matrix is

- banded, i.e., has zero elements for $|i - j| > n_b$,
- effectively banded with very small elements outside a band

then it may be be efficient to use conventional banded matrix multiplication.

See the following figure.

Flop Counts Toeplitz Matrix-Vector Muliplication