INVERSE PROBLEMS
Inverse Problems

Tomography
Overview

- Tomography
  - Absorption / emission
  - Fourier Slice Theorem and Filtered Back Projection
  - Algebraic Reconstruction
  - Applications
Outline

- Computed Tomography (CT)
  - Radon transform
  - Filtered Back-Projection
  - natural phenomena
  - glass objects
Computed Tomography (CT)
Some History

- **Radon transform** (1917)
  \[ \mathcal{R}\{f\}(\alpha, s) = \int_{c_{\alpha, s}} f \circ c_{\alpha, s}(t)dt \]

  - Radon: Inverse transform exists if all \((\alpha, s)\) are covered
  - First numerical application

  Viktor Ambartsumian (1936, astrophysics)
Some History

- **CT Scanning**

  Sketch of the invention

  - Godfrey Hounsfield (1919-2004)
  - Allan Cormack (1924-1998)

  - 1979 Nobel prize in Physiology or Medicine

  Prototype scanner

  Hounsfield's abdomen
The math

- X-rays are attenuated by body tissue and bones
  - Attenuation is spatially variant (attenuation coeff. $\sigma_a(x, y)$)

$$I(x) = I_0(x)e^{-\int_c \sigma_a(x, y)dy}$$

$$\Rightarrow \frac{I(x)}{I_0(x)} = e^{-\int_c \sigma_a(x, y)dy}$$

$$\Rightarrow \log \frac{I(x)}{I_0(x)} = -\int_c \sigma_a(x, y)dy$$

- $I(x), I_0(x)$ are known, determine $\sigma_a(x, y)$
- Ill-posed for only one direction $\alpha$
  - Need all
Well-Posed and Ill-Posed Problems

Definition [Hadamard1902]

- a problem is well-posed if
  1. a solution exists
  2. the solution is unique
  3. the solution continually depends on the data

- a problem is ill-posed if it is not well-posed
Inverse Problems

Tomography

-- Fourier-Based Techniques --
Computed Tomography

- Tomography is the problem of computing a function from its projections.
- A projection is a set of line integrals over function $m$ along some ray $c$.

\[ o = \int_{c} m(c(s)) \, ds \]

- Invert this equation (noise is present).

\[ o = \int_{c} m(c(s)) \, ds + n \]

- If infinitely many projections are available this is possible (Radon transform) \cite{Radon1917}.
Computed Tomography – Frequency Space Approach

- Fourier Slice Theorem
- The Fourier transform of an orthogonal projection is a slice of the Fourier transform of the function
Computed Tomography – FST

Fourier Slice Theorem

2D spatial domain

Integration

Slicing

1D spatial domain

1D frequency domain

\[ F \{ I(x') \} \]
Computed Tomography – Frequency Space Approach

- for recovery of the 2D function we need several slices
- slices are usually interpolated onto a rectangular grid
- inverse Fourier transform
- gaps for high frequency components
  → artifacts
Frequency Space Approach - Example

without noise!

original (Shepp-Logan head phantom)

reconstruction from 18 directions

reconstruction from 36 directions

reconstruction from 90 directions
Filtered Back-Projection

- Fourier transform is linear
  - \( \rightarrow \) we can sum the inverse transforms of the lines in frequency space instead of performing the inverse transform of the sum of the lines

backprojection:
Filtered Back-Projection

- Why filtering?
- Discrete nature of measurements gives unequal weights to samples
- Compensate

Would like to have wedge shape for one discrete measurement

Have a bar shape (discrete measurement)

Compensate to have equal volume under filter

Frequency domain

(a)

(b)

(c)

High pass filter
Filtered Back-Projection (FBP)

- high pass filter 1D projections in spatial domain
- back-project
- blurring is removed
  - FBP can be implemented on the GPU
  - projective texture mapping
Frequency Space based Methods

- **Advantages**
  - Fast processing
  - Incremental processing (FBP)

- **Disadvantages**
  - need orthogonal projections
  - sensitive to noise because of high pass filtering
  - Frequency-space artifacts, e.g. ringing
  - Equal angular view spacing (or adaptive filtering)
Inverse Problems

Tomography

-- Algebraic Techniques --
Algebraic Reconstruction Techniques (ART)

- object described by $\Phi$, a density field of e.g. emissive soot particles
- pixel intensities are line integrals along line of sight

$$I_p = \int_{C} \phi \, ds$$

- Task: Given intensities, compute $\Phi$
ART

- Algebraic Reconstruction Technique (ART)
- Discretize unknown $\Phi$ using a linear combination of basis functions $\Phi_i$

\[ I_p = \int c \left( \sum_i a_i \Phi_i \right) ds \]

\[ \rightarrow \text{linear system} \quad p = Sa \]

\[ I_p = \sum_i a_i \left( \int_{c_p} \phi_i \, ds \right) \]
ART

- Discretize unknown \( \Phi \) using a linear combination of basis functions \( \Phi_i \)

\[ I_p = \int_c \left( \sum_i a_i \phi_i \right) ds \]

\[ I_p = \sum_i a_i \left( \int_{c_p} \phi_i ds \right) \]

- Need several views
## ART – Matrix Structure

\[ I_p = \sum_i a_i \left( \int_{c_p} \phi_i \, ds \right) \]

\[ I = S a \]

### Basis functions

\[ i \]

\[ p \]

<table>
<thead>
<tr>
<th>( \int \phi_1 , ds )</th>
<th>( \int \phi_2 , ds )</th>
<th>( \int \phi_3 , ds )</th>
<th>( \int \phi_4 , ds )</th>
<th>( \int \phi_5 , ds )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \int_{c_1} \phi_1 , ds )</td>
<td>( \int_{c_1} \phi_2 , ds )</td>
<td>( \int_{c_1} \phi_3 , ds )</td>
<td>( \int_{c_1} \phi_4 , ds )</td>
<td>( \int_{c_1} \phi_5 , ds )</td>
</tr>
<tr>
<td>( \int_{c_2} \phi_1 , ds )</td>
<td>( \int_{c_2} \phi_2 , ds )</td>
<td>( \int_{c_2} \phi_3 , ds )</td>
<td>( \int_{c_2} \phi_4 , ds )</td>
<td>( \int_{c_2} \phi_5 , ds )</td>
</tr>
<tr>
<td>( \int_{c_3} \phi_1 , ds )</td>
<td>( \int_{c_3} \phi_2 , ds )</td>
<td>( \int_{c_3} \phi_3 , ds )</td>
<td>( \int_{c_3} \phi_4 , ds )</td>
<td>( \int_{c_3} \phi_5 , ds )</td>
</tr>
</tbody>
</table>

invert LS in a least squares sense:

\[ a = (S^T S)^{-1} S^T I \]
Frequency Space based Methods - Disadvantages

- **Advantages**
  - Accommodates flexible acquisition setups
  - Can be made robust to noise (next lecture)
  - Arbitrary or adaptive discretization
  - Can be implemented on GPU

- **Disadvantages**
  - May be slow
  - May be memory-consuming
Inverse Problems

Tomography

--- Applications ---
CT Applications in measurement and quality control

- Acquisition of difficult to scan objects
- Visualization of internal structures (e.g. cracks)
- No refraction
reconstruction of flames using a multi-camera setup
Flame tomography

- Calibrated, synchronized camera setup
  - 8 cameras, 320 x 240 @ 15 fps

8 input views in original camera orientation

Camera setup

[Ihrke’ 04]
Sparse View ART - Practice

- Large number of projections is needed
- In case of dynamic phenomena
  - → many cameras
    - expensive
    - inconvenient placement
- Straightforward application of ART with few cameras not satisfactory
Visual Hull Restricted Tomography

Zero coefficients

Object

C_1

C_2

C_3

[Ihrke’ 04]
Visual Hull Restricted Tomography

- Only a **small** number of voxels contribute
- Remove voxels that do not contribute from linear system
- Complexity of inversion is significantly reduced

[Ihrke’ 04]
Animated Flame Reconstruction

frame 86

frame 194

animated reconstructed flames

[Ihrke’ 04]
Smoke Reconstructions

[Ihrke’ 06]
3D Reconstruction of Planetary Nebulae

- only one view available
- exploit axial symmetry
- essentially a 2D problem

[Magnor04]
Schlieren Tomography

Input

Optical flow

Tomography

Output
Schlieren Tomography - Acquisition

16 camera array (consumer camcorders)
Synchronization & rolling shutter compensation
Schlieren Tomography - Acquisition

16 camera array (consumer camcorders)

Synchronization & rolling shutter compensation
Schlieren Tomography - Acquisition

16 camera array (consumer camcorders)

Synchronization & rolling shutter compensation
Schlieren Tomography - Acquisition

16 camera array (consumer camcorders)

Synchronization & rolling shutter compensation
Schlieren CT – Image Processing

Input

Optical flow

Mask
Schlieren CT – Background Pattern

High frequency detail everywhere

Decouple pattern resolution from sensor

Wavelet noise [Cook 05]
Schlieren CT - Image Formation

Image formation in continuously refracting media

Curved Rays

Described well by Ray Equation of Geometric Optics
Continuous ray tracing, e.g. [Stam 96, Ihrke 07]

Set of 1st order ODE’s:

\[
\frac{d}{ds} \left( n \frac{dx}{ds} \right) = \nabla n
\]

\[
n \frac{dx}{ds} = d
\]

\[
\frac{d}{ds} d = \nabla n
\]
Continuous ray tracing, e.g. [Stam 96, Ihrke 07]

Set of ODE’s:

$$\frac{ndx}{ds} = d$$

$$\frac{d\nabla n}{ds} = \nabla n$$
Schlieren CT - Ray equation

Integrating

\[ \frac{d\mathbf{d}}{ds} = \nabla n \]

yields

\[ \mathbf{d}^{out} = \mathbf{d}^{in} + \int_{c} \nabla n \, ds \]
Schlieren Tomography

Basic equation for Schlieren Tomography

\[ d^{out} - d^{in} = \int_{c} \nabla n \, ds \]
Schlieren Tomography

Based on measurements of line integrals from different orientations

\[ \mathbf{d}^{out} - \mathbf{d}^{in} = \int_{\mathcal{C}} \nabla n \, ds \]
Schlieren Tomography

Ray path must be known

BUT: unknown refractive index

In practice, ray bending negligible

[Venkatakrishnan’04]

\[ d^{out} - d^{in} = \int_{c} \nabla n \, ds \]
Schlieren Tomography

Ray path must be known

BUT: unknown refractive index

Affects integration path only, equation still holds approximately!

\[ \mathbf{d}_{\text{out}} - \mathbf{d}_{\text{in}} = \int_{c} \nabla n \, ds \]
Schlieren Tomography - Measurements

Measure difference vector

\[ \mathbf{d}_{\text{out}} - \mathbf{d}_{\text{in}} = \int_{c} \nabla n \, ds \]
Schlieren Tomography - Measurements

Measure difference vector

Component parallel to optical axis is lost

\[ \mathbf{d}_{out} - \mathbf{d}_{in} = \int_{c} \nabla n \, ds \]
Schlieren Tomography – Linear System

Vector-valued tomographic problem

Discretize gradient

Radially symmetric basis functions

\[ \nabla n = \sum_i n_i \phi_i \]

Linear system in

\[ d^{out} - d^{in} = \int \sum n_i \phi_i \, ds = \sum n_i \int \phi_i \, ds \]
Given \( \nabla n \) from tomography

Compute \( n \) from definition of Laplacian

\[
\nabla \cdot \nabla n = \Delta n
\]

Solve Poisson equation to get refractive index

- Inconsistent gradient field due to noise and other measurement error
- Anisotropic diffusion
Schlieren Tomography - Results
Schlieren Tomography - Results

Camera A  Camera B  Camera C
Schlieren Tomography - Results
3D Scanning of Glass Objects [Trifonov06]

- visible light tomography of glass objects
  - needs straight ray paths
- compensate for refraction
  - immerse glass object in water
  - add refractive index matching agent
  - “ray straightening”
- apply tomographic reconstruction
3D Scanning of Glass Objects [Trifonov06]

- Tomographic reconstruction results in volume densities
- use marching cubes to extract object surfaces
Layered 3D: Multi-Layer Displays

Layered 3D

[Wetzstein’11]
Tomographic Light Field Synthesis

[Wetzstein’11]
Tomographic Light Field Synthesis

[Wetzstein’11]
Tomographic Light Field Synthesis

virtual plane

attenuator

backlight

2D Light Field

[Wetzstein’11]

Ivo Ihrke - “Optimization Techniques in Computer Graphics” – Strasbour, 07/04/2014
Tomographic Light Field Synthesis

Image formation model:

\[ L(x, q) = I_0 e^{-m(r)dr} \]

\[ \bar{L}(x, q) = \ln \frac{L(x, q)}{I_0} = -m(r)dr \]

\[ \bar{I} = Pa \]

Tomographic synthesis:

\[ \arg \min_a \| \bar{I} + Pa \|^2, \text{ for } a \geq 0 \]

[Wetzstein'11]
Tomographic Light Field Synthesis

Image formation model:

\[ L(x, r) = I_0 e^{-\int_c (r)dr} \]

\[ \bar{L}(x, r) = \ln \frac{L(x, r)}{I_0} = \int_c (r)dr \]

\[ \bar{I} = Pa \]

Tomographic synthesis:

\[ \arg \min_a \| \bar{I} + Pa \|^2, \text{ for } a \geq 0 \]

[Wetzstein’11]
Multi-Layer Light Field Decomposition

Target 4D Light Field

Reconstructed Views

Multi-Layer Decomposition

[Wetzstein’11]

Prototype Layered 3D Display

Transparency stack with acrylic spacers

Prototype in front of LCD (backlight source)

[Wetzstein’11]
Deconvolution
Outline

 Deconvolution Theory
  – example 1D deconvolution
  – Fourier method
  – Algebraic method
    – discretization
    – matrix properties
    – regularization
    – solution methods

 Deconvolution Examples
Applications - Astronomy

- BEFORE

- AFTER

Images courtesy of Robert Vanderbei

Applications - Microscopy

Images courtesy Meyer Instruments

Inverse Problem - Definition

- forward problem
  - given a mathematical model $M$ and its parameters $m$, compute (predict) observations $o$

  \[ o = M(m) \]

- inverse problem
  - given observations $o$ and a mathematical model $M$, compute the model's parameters

  \[ m = M^{-1}(o) \]
Inverse Problems – Example
Deconvolution

- forward problem – convolution
  - example blur filter
  - given an image \( m \) and a filter kernel \( k \),
    compute the blurred image \( o \)

\[
o = m \otimes k
\]
Inverse Problems – Example

- inverse problem – deconvolution
  - example blur filter
  - given a blurred image $o$ and a filter kernel $k$, compute the sharp image
  - need to invert

\[ o = m \otimes k + n \]

- $n$ is noise
Deconvolution

--Fourier Solution--
Deconvolution - Theory

- deconvolution in Fourier space
- convolution theorem (F is the Fourier transform):
  \[ o = m \otimes k, \quad \Rightarrow \mathcal{F}\{o\} = \mathcal{F}\{m\} \cdot \mathcal{F}\{k\} \]

- deconvolution:
  \[ \Rightarrow \mathcal{F}\{m\} = \frac{\mathcal{F}\{o\}}{\mathcal{F}\{k\}} \]

- problems
  - division by zero
  - Gibbs phenomenon
    (ringing artifacts)
A One-Dimensional Example – Deconvolution Spectral

- most common: \( \mathcal{F}\{k\} \) is a low pass filter

\[
\frac{1}{\mathcal{F}\{k\}}
\]

- \( \frac{1}{\mathcal{F}\{k\}} \), the inverse filter, is high pass

- \( \frac{1}{\mathcal{F}\{k\}} \) amplifies noise and numerical errors
A One-Dimensional Example – Deconvolution Spectral

- reconstruction is noisy even if data is perfect!
  - Reason: numerical errors in representation of function

![Signal and Kernel](image1)

![Observation and Reconstructed Signal](image2)
A One-Dimensional Example – Deconvolution Spectral

- spectral view of signal, filter and inverse filter
A One-Dimensional Example – Deconvolution Spectral

- solution: restrict frequency response of high pass filter (clamping)

\[
\mathcal{F}\{g\} := \begin{cases} 
\frac{1}{\mathcal{F}\{k\}} & \text{if } \frac{1}{\mathcal{F}\{k\}} < \gamma \\
\frac{\mathcal{F}\{k\}}{\gamma |\mathcal{F}\{k\}|} & \text{else}
\end{cases}
\]

\[
\mathcal{F}\{m\} = \mathcal{F}\{o\} \cdot \mathcal{F}\{g\}
\]
A One-Dimensional Example - Deconvolution Spectral

- reconstruction with clamped inverse filter
A One-Dimensional Example – Deconvolution Spectral

- spectral view of signal, filter and inverse filter

![FFT of signal](image1)

![FFT of kernel](image2)

![FFT of kernel](image3)

![FFT of est. kernel](image4)
A One-Dimensional Example – Deconvolution Spectral

- Automatic per-frequency tuning: Wiener Deconvolution
  - Alternative definition of inverse kernel
  - Least squares optimal
  - Per-frequency SNR must be known

\[
\mathcal{F}\{g\}(\omega) := \frac{1}{\mathcal{F}\{k\}(\omega)} \left| \frac{\mathcal{F}\{k\}(\omega)}{\mathcal{F}\{k\}(\omega) \left| \mathcal{F}\{k\}(\omega) \right|^2 + \frac{1}{\text{SNR}(\omega)}} \right|^2 \]

Inverse Problems - Deconvolution

Deconvolution

-- Algebraic Solution --
A One-Dimensional Example - Deconvolution Algebraic

- alternative: algebraic reconstruction
- convolution

\[ o(x) = \int_{-\infty}^{\infty} m(t)k(x - t) \, dt \]

- discretization: linear combination of basis functions

\[ m(t) = \sum_{i=0}^{N} m_i \phi_i(t) \]
A One-Dimensional Example – Deconvolution Algebraic

- discretization:
  - observations are linear combinations of convolved basis functions
  - linear system with unknowns $m_i$
  - often over-determined, i.e. more observations $o$ than degrees of freedom ($\#$ basis functions)

\[
o(x) &= \{m \otimes k\} (x) \\
&= \int_{-\infty}^{\infty} m(t)k(x - t)dt \\
&= \int_{-\infty}^{\infty} \sum_{i=0}^{N} m_i \phi_i(t)k(x - t)dt \\
&= \sum_{i=0}^{N} m_i \int_{-\infty}^{\infty} \phi_i(t)k(x - t)dt \\
&= \sum_{i=0}^{N} m_i \{\phi_i \otimes k\} (x)
\]

\[o = Mm \quad \text{linear system}\]
discretization:

- observations are linear combinations of convolved basis functions
- linear system with unknowns $m_i$
- often over-determined, i.e. more observations $o$ than degrees of freedom (# basis functions)

\[
o(x) = \{m \otimes k\}(x) = \int_{-\infty}^{\infty} m(t)k(x-t)dt = \int_{-\infty}^{\infty} \sum_{i=0}^{N} m_i \phi_i(t)k(x-t)dt = \sum_{i=0}^{N} m_i \int_{-\infty}^{\infty} \phi_i(t)k(x-t)dt = \sum_{i=0}^{N} m_i \{\phi_i \otimes k\}(x)
\]

\[o = \mathbf{Mm}\]

linear system
A One-Dimensional Example – Deconvolution Algebraic

- normal equations

\[
\min_x \| Ax - b \|_2^2 = \min_x (Ax - b)^T (Ax - b) = \min_x f(x)
\]

\[
\nabla f = 2A^T Ax - 2A^T b = 0
\]

→ solve \( A^T Ax = A^T b \) to obtain solution in a least squares sense

→ apply to deconvolution

solution is completely broken!
A One-Dimensional Example – Deconvolution Algebraic

- Why?
- analyze distribution of eigenvalues
- Remember:

\[
\det A = \prod_{i=0}^{N} \lambda_i \quad \text{and} \quad \det A = 0 \Rightarrow \text{Matrix is under-determined}
\]

- we will check the singular values
  - Ok, since \( A^T A \) is SPD (symmetric, positive semi-definite)
  - \( \Rightarrow \) non-negative eigenvalues
- Singular values are the square root of the eigenvalues
A One-Dimensional Example – Deconvolution Algebraic

- matrix $M^T M$ has a very wide range of singular values!
- more than half of the singular values are smaller than machine epsilon ($10^{-16}$) for double precision

Log-Plot!

$Ivo Ihrke$ - “Optimization Tech
A One-Dimensional Example – Deconvolution Algebraic

- Why is this bad?
- Singular Value Decomposition: U, V are orthonormal, \( D \) is diagonal

\[
M = UDV^T
\]

- Inverse of \( M \):

\[
M^{-1} = (UDV^T)^{-1} = V^{-T}D^{-1}U^{-1} = VD^{-1}U^T
\]

- singular values are diagonal elements of \( D \)
- inversion:

\[
D^{-1} = \text{diag} \left( \frac{1}{D_{i,i}} \right)
\]
A One-Dimensional Example – Deconvolution Algebraic

- computing model parameters from observations:
  \[ \mathbf{m} = \mathbf{M}^{-1} \mathbf{o} = \mathbf{V} \mathbf{D}^{-1} \mathbf{U}^T \mathbf{o} \]
- again: amplification of noise
- potential division by zero

Log-Plot!

\[ D^{-1} = \text{diag} \left( \frac{1}{D_{i,i}} \right) \]
- inverse problems are often ill-conditioned (have a numerical null-space)
- inversion causes amplification of noise

\[ 10^2 \]
\[ 10^{-8} \]
\[ 10^{-16} \]
Well-Posed and Ill-Posed Problems

- Definition [Hadamard1902]
  - a problem is well-posed if
    1. a solution exists
    2. the solution is unique
    3. the solution continually depends on the data
Well-Posed and Ill-Posed Problems

- Definition [Hadamard1902]
  - a problem is ill-posed if it is not well-posed
    - most often condition (3) is violated
    - if model has a (numerical) null space, parameter choice influences the data in the null-space of the data very slightly, if at all
    - noise takes over and is amplified when inverting the model
Condition Number

- measure of ill-conditionedness: condition number
- measure of stability for numerical inversion
- ratio between largest and smallest singular value

\[ \rho(A) = \frac{\sigma_0}{\sigma_N}, \quad \sigma_0 > \ldots > \sigma_N \] are the singular values of \( A \)

- smaller condition number \( \rightarrow \) less problems when inverting linear system
- condition number close to one implies near orthogonal matrix
Truncated Singular Value Decomposition

- solution to stability problems: avoid dividing by values close to zero
- Truncated Singular Value Decomposition (TSVD)

\[ d^+ = \begin{cases} \frac{1}{D_{i,i}} & \text{if } D_{i,i} > \epsilon \\ 0 & \text{else} \end{cases} \]

\[ D^+ = \text{diag} (d^+) \]

\[ M^+ = VD^+U^T \]

- \( \epsilon \) is called the regularization parameter
Minimum Norm Solution

- Let $K[A]$ be the null-space of $A$ and $X_K \in K$

$$\Rightarrow AX_K = 0$$

$$\Rightarrow AX = A(X_{K\perp} + X_K)$$

$$= AX_{K\perp} + AX_K$$

$$= AX_{K\perp} + 0$$

$$= AX_{K\perp}$$

$$= b$$

- $X_{K\perp}$ is the minimum norm solution

\[\begin{align*}
\text{Linear Variety Containing Solutions of } A\hat{x} = b \\
\hat{X} \text{ is non-optimum} \\
\hat{X}_{K\perp} \in K(A)^\perp \text{ is optimum (minimum norm)} \\
\hat{X} = \hat{X}_{K\perp} + \hat{X}_K \text{ where } \hat{X}_K \in K(A)
\end{align*}\]
Regularization

- countering the effect of ill-conditioned problems is called regularization

- an ill-conditioned problem behaves like a singular (i.e. under-constrained) system

- family of solutions exist
  → impose additional knowledge to pick a favorable solution

- TSVD results in minimum norm solution
Example – 1D Deconvolution

- back to our example – apply TSVD
- solution is much smoother than Fourier deconvolution

unregularized solution

TSVD regularized solution $\epsilon = 10^{-6}$
Large Scale Problems

- consider 2D deconvolution
- 512x512 image, 256x256 basis functions
  → least squares problem results in matrix that is 65536x65536!
- even worse in 3D (millions of unknowns)
- problem: SVD is \( \mathcal{O}(N^3) \)

<table>
<thead>
<tr>
<th>system size</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
<th>4096</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVD time (in s)</td>
<td>0.27</td>
<td>1.75</td>
<td>12.54</td>
<td>96.28</td>
</tr>
</tbody>
</table>

Intel Xeon 2-core (E5503) @ 2GHz (introduced 2010)

- today impractical to compute for systems larger than \( > 16384^2 \)
  (takes a couple of hours)
- Question: How to compute regularized solutions for large scale systems?
Explicit Regularization

- Answer: modify original problem to include additional optimization goals (e.g. small norm solutions)

\[
\min_x \quad \alpha \|Ax - b\|^2_2 + (1 - \alpha)\|Rx\|^2_2 = \\
\min_x \quad \alpha (Ax - b)^T (Ax - b) + (1 - \alpha)x^TR^TRx = \\
\min_x \quad \hat{f}(x)
\]

- minimize modified quadratic form

\[
\nabla \hat{f}(x) = 2\alpha A^T Ax - 2A^T b + 2(1 - \alpha)R^TRx = 0
\]

- regularized normal equations:

\[
(\alpha A^T Ax + (1 - \alpha)R^TR)x = A^T b
\]
Modified Normal Equations

- include data term, smoothness term and blending parameter

\[
\left( \alpha A^T Ax + (1 - \alpha)R^T R \right)x = A^T b
\]

- blending (regularization) parameter

Prior information (popular: smoothness)
Tikhonov Regularization

- setting $R = I$ and $\lambda = \frac{1 - \alpha}{\alpha}$ we have a quadratic optimization problem with data fitting and minimum norm terms

\[
\min_x (Ax - b)^T (Ax - b) + \lambda x^T x
\]

- large $\lambda$ will result in smooth solution, small $\lambda$ fits the data well

- find good trade-off
Tikhonov Regularization - Example

- reconstruction for different choices of $\lambda$
- small lambda, many oscillations
- large lambda, smooth solution (in the limit constant)
Tikhonov Regularization - Example
L-Curve criterion [Hansen98]

- need automatic way of determining
- want solution with small oscillations
- also want good data fit
- log-log plot of norm of residual (data fitting error) vs. norm of the solution (measure of oscillations in solution)
L-Curve Criterion

- video shows reconstructions for different $\lambda$
- start with $\lambda = 10^{-12}$

L-Curve

regularized solution
L-Curve Criterion

- compute L-Curve by solving inverse problem with choices of $\lambda$ over a large range, e.g. $\lambda \in [10^{-12}, 10^7]$

- point of highest curvature on resulting curve corresponds to optimal regularization parameter

- curvature computation

$$\kappa = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{3/2}}$$

- find maximum $\kappa$ and use corresponding $\lambda$ to compute optimal solution
L-Curve Criterion – Example
1D Deconvolution

- L-curve with automatically selected optimal point
- optimal regularization parameter is different for every problem

\[ \lambda = 10^{-12} \]

\[ \lambda = 0.0429 \]

\[ \lambda = 10^7 \]
L-Curve Criterion – Example 1D Deconvolution

- regularized solution (red) with optimal $\lambda = 0.0429$
Solving Large Linear Systems

- we can now regularize large ill-conditioned linear systems
- How to solve them?
  - Gaussian elimination: $O(N^3)$
  - SVD: $O(N^3)$
- direct solution methods are too time-consuming
- Solution: approximate iterative solution
stationary iterative methods [Barret94]

- Examples
  - Jacobi
  - Gauss-Seidel
  - Successive Over-Relaxation (SOR)

- use fixed-point iteration
  \[ x^{t+1} = Gx^t + c \]
  - matrix G and vector c are constant throughout iteration
  - generally slow convergence
  - don't use for practical applications
Iterative Solution Methods for Large Linear Systems

- non-stationary iterative methods [Barret94]
  - conjugate gradients (CG)
    - symmetric, positive definite linear systems (SPD)
  - conjugate gradients for the normal equations
    short CGLS or CGNR
    - avoid explicit computation of $A^TA$
  - CG – type methods are good because
    - fast convergence (depends on condition number)
    - regularization built in!
    - number of iterations = regularization parameter
    - behave similar to truncated SVD
Iterative Solution Methods for Large Linear Systems

- iterative solution methods require only matrix-vector multiplications
- most efficient if matrix $A$ is *sparse*
- sparse matrix means lots of zero entries
- back to our hypothetical $65536 \times 65536$ matrix
- memory consumption for full matrix:

  $$2^{16} \times 2^{16} \times 8 \text{ bytes} = 32 \text{ Gbyte}$$

- sparse matrices store only non-zero matrix entries
- **Question:** How do we get sparse matrices?
Iterative Solution Methods for Large Linear Systems

- answer: use a discretization with basis functions that have local support, i.e. which are themselves zero over a wide range

- for deconvolution the filter kernel should also be locally supported

\[
\text{discretized model: } \quad o = \sum_{i=0}^{N} m_i \{ \phi_i \otimes k \}
\]
Iterative Solution Methods for Large Linear Systems

- answer: use a discretization with basis functions that have local support, i.e. which are themselves zero over a wide range

- for deconvolution the filter kernel should also be locally supported

\[
\begin{array}{l}
\text{discretized model: } \quad o = \sum_{i=0}^{N} m_i \{ \phi_i \otimes k \}
\end{array}
\]

will be zero over a wide range of values
Iterative Solution Methods for Large Linear Systems

sparse matrix structure for 1D deconvolution problem
Inverse Problems – Wrap Up

- inverse problems are often ill-posed
- if solution is unstable – check condition number
- if problem is small $< 4000^2$ use TSVD and Matlab
- otherwise use CG if problem is symmetric (positive definite), otherwise CGLS
- if convergence is slow try Tikhonov regularization – it's simple
  - improves condition number and thus convergence
- if problem gets large $> 15000^2$ make sure you have a sparse linear system!
- if system is sparse, avoid computing $A^T A$ explicitly – it is usually dense