1 One Dimensional Problems

1.1 Continuous Problem

We consider the integral equation

\[ g(s) = \int_0^\pi h(s,t)f(t)dt. \]

When \( f(t) = 0 \) outside the interval \([0, \pi]\) this completely defines the formulation. Suppose that \( s \) and \( t \) are discretized on the interval \([0, \pi]\) by

\[ s_i = \frac{(2i - 1)\pi}{2m} \quad t_j = \frac{(2j - 1)\pi}{2m} \quad \Delta s = \frac{\pi}{m} = \Delta t \quad i,j = 1 : m \]
then the integral equation is approximated by the midpoint quadrature rule
\[ g(s_i) \approx \sum_{j=1}^{m} h(s_i, t_j)f(t_j)\Delta t \quad 1 \leq i \leq m \]
and we solve the system of equations
\[ b = Ax \quad (A)_{ij} = \Delta t h(s_i, t_j) \quad x_j = (t_j) \quad b_i = g(s_i). \]

1.2 Solutions may be obtained with respect to a basis

The Singular Value Decomposition (SVD): General A
Suppose, here, \( A \in \mathbb{R}^{N \times N} \):

- \( A = U\Sigma V^T \), \( U \) and \( V \) are orthogonal \( U^T U = I_N = V^T V \),
- \( \sigma_i \) are the singular values of \( A \),
  \[ \Sigma = \text{diag}(\sigma_i) \quad \sigma_1 \geq \sigma_2 \geq \ldots \sigma_N \geq 0. \]
- \( u_i \) (columns of \( U \)) are the left singular vectors of \( A \)
- \( v_i \) (columns of \( V \)) are the right singular vectors of \( A \)
- Assume \( \sigma_N > 0 \), \( A \) is nonsingular, then
  \[ A^{-1} = V\Sigma^{-1}U^T = \sum_{i=1}^{N} \frac{v_iu_i^T}{\sigma_i}, \quad A = \sum_{i=1}^{N} \sigma_iu_iv_i^T \]
- Solution expressed with the SVD components is a weighted linear combination of basis vectors \( v_i \).
  \[ x = \sum_{i=1}^{N} \left( \frac{u_i^T b}{\sigma_i} \right) v_i \quad (1) \]

1.2.1 Impact of Error in the Right Hand Side

- Practically the measured data is contaminated by noise, we denote by \( e \).
  \[ b = b_{\text{exact}} + e. \]
- The spectral decomposition acts on the noise term in the same way it acts on the exact right hand side \( b \). e.g.
  \[ x = \sum_{i=1}^{N} \left( \frac{u_i^T (b_{\text{exact}} + e)}{\sigma_i} \right) v_i = x_{\text{exact}} + \sum_{i=1}^{N} \left( \frac{u_i^T e}{\sigma_i} \right) v_i, \quad \sigma_i \neq 0. \]
• If \( e \) is uniform, we expect \( |u_i^T e| \) to be similar magnitude \( \forall i \).

• Note \( \text{cond}(A) = \sigma_1/\sigma_N, \sigma_N \neq 0 \). Typically, expect \( \sigma_i \) to decay to zero for ill-posed problems. Hence \( \text{cond}(A) \) may be large.

• \( \sigma_i \) small represents high frequency component in the sense that \( u_i, v_i \) have more sign changes as \( i \) increases.

• \( (\frac{u_i^T e}{\sigma_i}) \) is the coefficient of \( v_i \) in the error, the weight of \( v_i \) in the error, where \( v_i \) is associated with a spatial frequency. When \( 1/\sigma_i \) is large the contribution of the high frequency error is magnified due to \( (\frac{u_i^T e}{\sigma_i}) \).

• We note that this applies for any spectral decomposition with respect to the smallest components in the denominator, regardless of the order of the components.

1.2.2 Truncating the Spectral Expansion

Given expansion with respect to a basis SVD shows it may be reasonable to truncate

\[
x_k = \sum_{i=1}^{k} (\frac{u_i^T b}{\sigma_i}) v_i = A_k b
\]

This defines the rank \( k \) matrix \( A_k \).

• The use of the truncated expansion is feasible only if we can first calculate the SVD or decomposition of \( A \) efficiently.

• The limit on the sum \( k \) is regarded as a regularization parameter. We can change \( k \) and obtain different solutions.

• Choice of \( k \) controls the degree of low pass filtering which is applied. i.e. controls the attenuation of the high frequency components.
Figure 1: Truncated SVD Solutions: $|u_i^T b|$. Error in the basis contaminates the solution even though data is noise free. The solution is over or under smoothed dependent on $k$.

1.3 Assignment

1. Download the Regularization toolbox [http://www2.imm.dtu.dk/~pcha/Regutools/](http://www2.imm.dtu.dk/~pcha/Regutools/)

2. Write a Matlab script which solves the problem phillips of size 128 using the SVD for the matrix $A$ from the phillips problem.

3. Add noise to the right hand side data using the Matlab function `randn`: $b_{\text{noisy}} = b + \zeta n$, where $n = \text{randn}(128)$ and $\zeta$ determines the noise level. Try $\zeta = .1$.

4. Plot $b_{\text{noisy}}$ and $b$ to see the noise in the solution.

5. Solve to obtain the solution subject to noise $x_{\text{noisy}}$. Plot both $x$ and $x_{\text{noisy}}$.

6. Obtain the solutions $x_{\text{TSVD}}$ and $x_{\text{TSVD noisy}}$ using truncated expansions for the SVD. Determine the number of terms that you can use to obtain a stable solution.

7. Submit under the assignment in Blackboard.
2 Ill-Posed Problems January 21, 2014

Consider the mapping $A$ which takes the solution $f$ to output data $g$. $Af = g$. Inverse problem: find $f$ given $g$ and $A$.

**Definition 1** (Well-Posed). The problem of finding $f$ from $g$ is called well-posed (by Hadamard in 1923) if all

- **Existence** a solution exists for any data $g$ in data space,
- **Uniqueness** the solution $f$ is unique
- **Stability** continuous dependence of $f$ on $g$: the inverse mapping $g \rightarrow f$ is continuous.

The first two conditions are equivalent to saying that the operator $A$ has a well defined inverse $A^{-1}$.

Suppose there exists $f_1$ such that $Af_1 = 0$, then $Af = g$ and $Af(f + f_1) = g$ so $f$ is not unique.

Moreover, we require that the domain of $A^{-1}$ is all of data space.

**Definition 2** (Ill-Posed: according to Hadamard). A problem is ill-posed if it does not satisfy all three conditions for well-posedness.

Alternatively an ill-posed problem is one in which
1. $g \notin \text{range}(A)$
2. inverse is not unique because more than one image is mapped to the same data, or
3. an arbitrarily small change in the data can cause an arbitrarily large change in the image.

**Example for a Discrete problem - do we know if a solution is good**

Consider the linear system

$$A = \begin{pmatrix} 0.16 & 0.10 \\ 0.17 & 0.11 \\ 2.02 & 1.29 \end{pmatrix}, \quad b = \begin{pmatrix} 0.26 \\ 0.28 \\ 3.31 \end{pmatrix}, \quad x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

The least squares solution yields

$$x_{ls} = [7.0089, -8.3957]^T \|Ax_{ls} - b\|^2 = .00047, \quad \|x - x_{ls}\|^2 = 124.38$$

Perturbing $b$ by $\delta b = [.01, .01, .001]$ yields

$$x_{ls}' = [7.6946, -9.4674]^T, \quad \|Ax_{ls}' - b\|^2 = .00048, \quad \|x - x_{ls}'\|^2 = 154.38$$

- A small residual does not imply a realistic solution
- Even without noise the ill-conditioning of $A$ leads to a poor solution
- Perturbing $b$ leads to a larger perturbation in $x$. 


2.1 Singular Value Expansion (see Hansen Discrete Inverse problems, Chapter 2)

Let $L^2([0, 1] \times [0, 1])$ be space of square integrable functions on $[0, 1] \times [0, 1]$, i.e. $h \in L^2([0, 1] \times [0, 1])$

$$\|h\|_2^2 = \int_0^1 \int_0^1 h(s, t)^2 \, ds \, dt$$

is finite.

The **Singular Value Expansion** is

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

- For the inner product $\langle \phi, \psi \rangle = \int_0^1 \phi(t) \psi(t) \, dt$,
  $$\langle u_i, u_j \rangle = \langle v_i, v_j \rangle = \delta(i - j)$$
  orthonormality

- $\mu_i$ are the *singular values* of $h$, ordered from large to small
  $$\mu_1 \geq \mu_2 \geq \cdots \geq 0.$$

### 2.1.1 Properties of the SVE

1. $h(s, t)$ square integrable implies $\sum_{i=1}^{\infty} \mu_i^2$ must be bounded:

   $$\|h\|^2 = \int_0^1 \int_0^1 \left( \sum_{i,j=1}^{\infty} \mu_i \mu_j u_i(s) u_j(s) v_i(t) v_j(t) \right) \, ds \, dt$$

   $$= \sum_{i=1}^{\infty} \mu_i^2,$$

   by orthonormality of $(v_i(t), u_i(s))$.

   For the sum to be bounded $\mu_i$ must decay faster than $1/\sqrt{i}$.

2. Left and right *singular functions* of $h(s, t)$ are $(u_i, v_i)$:

   - $\{\mu_i^2, u_i\}$ of $\int_0^1 h(s, x) h(t, x) \, dx$ and
   - $\{\mu_i^2, v_i\}$ of $\int_0^1 h(x, s) h(x, t) \, dx$.

   $$\langle \int_0^1 h(s, x) h(t, x) \, dx, u_k(s) \rangle = \langle \sum_{i,j} \mu_i \mu_j u_i(s) u_j(t) \int_0^1 v_i(x) v_j(x) \, dx, u_k(s) \rangle$$

   $$= \langle \sum_i \mu_i^2 u_i(t), u_k(s) \rangle = \mu_i^2 u_k(t).$$

   We note

   $$\langle v_k(t), \int_0^1 h(x, t) h(x, s) \, dx \rangle = \langle v_k(t), \int_0^1 \sum_{i,j} \mu_i \mu_j v_i(t) v_j(s) (u_i(x) u_j(x) \, dx) \rangle$$

   $$= \langle v_k(t), \sum_i \mu_i^2 v_i(t) v_i(s) \rangle = \sum_i \mu_i^2 v_i(s) < v_k(t), v_i(t) > = \mu_k^2 v_k(s).$$
2.1.2 Properties of the Singular Functions

Basis for function Space $u_i$ and $v_i$ provide a basis for $L_2([0,1])$. If $f, g \in L_2([0,1])$ then

$$f(t) = \sum_i <v_i, f>v_i(t) \text{ and } g(s) = \sum_i <u_i, g>u_i(s)$$

Fundamental Mapping $v_i$ to $u_i$:

$$\int_0^1 h(s,t)v_i(t)dt = \int_0^1 \sum_{j=1}^{\infty} (\mu_j u_j(s)v_j(t)) v_i(t)dt$$

i.e. $<h(s,t), v_i(t)> = \mu_i u_i(s), \ i = 1, 2, \ldots$

by the orthonormality of $v_i$.

Smoothing of $v_i$ by kernel $h(s,t)$ to give $\mu_i u_i$.

The Integral Equation $g(s) = \int_0^1 h(s,t)f(t)dt$

Observe by the expansion for $g$ and its relation to the integral equation that

$$\sum_i <u_i, g>u_i(s) = g(s) = \int_0^1 \left( \sum_j \mu_j u_j(s)v_j(t) \right) f(t) dt$$

$$= \int_0^1 \sum_{ij} \mu_j u_j(s)v_j(t) <v_i, f>v_i(t) dt = \sum_i \mu_i <v_i, f>v_i(s)$$

i.e. $\sum_i <u_i, g>u_i(s) = \sum_i \mu_i <v_i, f>v_i(s)$ yields

$$\sum_i <u_i, g>u_i(s) = \sum_i \mu_i <v_i, f>v_i(s)$$

and $f(t) = \sum_i <v_i, f>v_i(t)$ (4)

- By Fundamental relation $\mu_i u_i(s)$ is smoothed $v_i$.
- Hence comparing sums of $f$ and $g$, $g$ is smoother than $f$.
- Also, if $\mu_i = 0$ we can only have solution $f$ of the integral equation if component $<u_i, g>u_i(s)$ is zero.
- For $\mu_i \neq 0, i = 1, 2, \ldots$ with $\mu_i <v_i, f>=<u_i, g>$ (taking inner product with $u_j(s)$ on each side in (4)) yields

$$f(t) = \sum_i \frac{<u_i, g>}{\mu_i} v_i(t)$$
2.2 The Continuous Picard Condition

\( f \in L_2([0,1]) \) only exists if the infinite sum converges.

**Definition 3** (Picard Condition). \( f \) is square integrable if

\[
\|f\|_2^2 = \int_0^1 f(t)^2 \, dt = \sum_{i=1}^{\infty} (\langle v_i, f \rangle)^2 = \sum_{i=1}^{\infty} \left( \frac{\langle u_i, g \rangle}{\mu_i} \right)^2 < \infty
\]

- Right hand coefficients \( \langle u_i, g \rangle \) must decay faster than \( \mu_i \).
- It is necessary that \( \exists N \) s.t. \( \forall i > N, \langle u_i, g \rangle \) decays faster than \( \mu_i \).
- \( g \) is square integrable if \( \langle u_i, g \rangle \) decay faster than \( 1/\sqrt{i} \) (look at \( \|g\|^2 \)), but Picard condition requires faster decay than \( \mu_i/\sqrt{i} \).

**Remark 1.** \( f \) is characterised by \( (\mu_i, u_i, v_i) \)

\[
f(t) = \sum_i \frac{\langle u_i, g \rangle}{\mu_i} v_i(t)
\]

- Decay rate of \( \mu_i \) is fundamental to behavior of ill-posed problem.
- Calculating \( f \) from \( g \) amplifies components \( v_i \) as \( \mu_i \to 0 \).
- High frequency \( v_i \) are amplified by inversion.
- The smaller \( \mu_i \) the greater the oscillations in \( u_i, v_i \).
- If \( h \) has continuous derivatives of order \( 0, \ldots, q \) \( \mu_i \) decay approximately as \( O(i^{-(q+1/2)}) \)
- If \( h \) is infinitely differentiable \( \mu_i \) decay faster, i.e. as \( O(r^i) \) for some \( 0 < r < 1 \). It is this decay fundamental to the kernel that creates the problem with inversion.

**Definition 4** (Ill-posed).

**Degree** if \( \exists \alpha > 0 \) s.t. \( \mu_i = O(i^{-\alpha}) \) then \( \alpha \) is the degree of ill-posedness

- Mildly ill-posed if \( \alpha \leq 1 \).
- Moderately ill-posed if \( \alpha > 1 \).
- Severely ill-posed if \( \mu_i = O(e^{-\alpha i}) \).
Further Details

• Note in discussion we always assume that the equality signs with infinite sums imply uniform convergence for continuous functions. i.e. for solution of the integral equation there exists $N$ such that for all $n > N$

$$\left| f(t) - \sum_{i=1}^{n} \frac{<u_i, g>}{\mu_i} v_i(t) \right| < \epsilon, \quad \forall t \in [0, 1]$$

• If the kernel is discontinuous convergence is with respect to the mean square,

$$\|f(t) - \sum_{i=1}^{n} \frac{<u_i, g>}{\mu_i} v_i(t)\|_2 < \epsilon, \quad \forall t \in [0, 1]$$

Example of an Ill-Posed Problem January 28, 2014

Consider the first kind Fredholm integral equation

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

Consider functions $f_p = \sin(2\pi pt), \quad p = 1, 2, \ldots$. For arbitrary square integrable $h$,

$$g_p = \int_0^1 h(s,t) \sin(2\pi pt) dt \to 0 \text{ as } p \to \infty.$$  

As frequency of $f_p$, amplitude of $g$ decreases due to smoothing by $h$

This is a statement of the Riemann Lebesque Lemma ($g_p$ is smoother than $f_p$)

Finding $f$ from smoother $g$ amplifies high frequencies in $f$.

Ratio $\frac{f}{g_p}$ can become arbitrarily large for $p$ large enough.

Problem is ill-posed because solution $f$ is not continuously dependent on data $g$.

Non existence of a Solution : Ursell

$$\int_0^1 \frac{1}{s + t + 1} f(t) dt = 1, \quad 0 \leq s \leq 1$$

Clearly $h(s,t) = \frac{1}{s + t + 1}$ and $g = 1$.

Defining $g_k(s) = \sum_{i=1}^{k} <u_i, g > u_i(s)$ then

$$\|g - g_k\|_2 \to 0 \quad \text{with} \quad k \to \infty \quad \text{but}$$

$$f_k(t) = \sum_{i=1}^{k} \frac{<u_i, g_k>}{\mu_i} v_i(t) \quad \text{satisfies} \|f_k\|_2 \to \infty.$$  

$f_k$ does not converge to a square integrable solution. (See reference for the plots)
2.2.1 Ambiguity in Inverse Problems: non unique solution

If the integral operator has a null space

\[ \int_0^1 h(s,t)f(t) \, dt = 0, \quad \text{for some } f(t) \]

then \( f \) is called an annihilator for \( h \), as is \( \alpha f \) for any scalar \( \alpha \).

Null space of \( h \) is the space of all annihilators \( f \). Moreover, by the fundamental relation

\[ \text{null}(h) = \text{span}\{v_i | \mu_i = 0\} \]

If there are only a finite number of \( \mu_i > 0 \), the kernel is degenerate.

Another example: degenerate kernel

Consider

\[ \int_{-1}^1 (s + 2t)f(t) \, dt = g(s), \quad -1 \leq s \leq 1. \]

It is possible to find the solution. We set \( h(s,t) = \sum_i \mu_i u_i(s)v_i(t) \) and solve for the constants under assumptions that we have both constant and first order polynomial terms for the basis functions, with normalization. We find

\[ u_1(s) = 1/\sqrt{2} \quad u_2(s) = \sqrt{3}/\sqrt{2}s \quad v_1(t) = \sqrt{3}/\sqrt{2}t \quad v_2(t) = 1/\sqrt{2} \]

\[ \mu_2 = 2/\sqrt{3} \quad \mu_1 = 4/\sqrt{3} \quad \mu_i = 0, i > 2 \]

A solution exists only if

\[ g \in \text{range}(h) = \text{span}\{u_1, u_2\} \quad \text{implies} \]

\[ g = a + bs \quad f = \frac{b}{4} + \frac{3}{2}at \]

But notice that \( f = 3t^2 - 1 \) is an annihilator and hence \( f \) is not unique!

2.2.2 Spectral Properties of the Singular Functions: Brief Overview

Define the integral operator

\[ [Hf](s) = \int_{-\pi}^\pi h(s,t)f(t) \, dt \]

and assume

1. \( h \) is real and continuous on \([−\pi, \pi] \times [−\pi, \pi]\).
2. for simplicity \( \|h(\pi,t) - h(-\pi,t)\|_2 = 0 \).
3. \( h \) is square integrable with singular set \((\mu_i, u_i, v_i)\) such that

\[ [Hv_j](s) = \mu_j u_j(s), \quad [H^*u_j](t) = \mu_j v_j(t) \]
An analysis can be performed to demonstrate that

*Singular functions are similar to Fourier functions*: for small $j$, the large singular values and corresponding singular functions correspond to low Fourier frequencies, but for large $j$ (small singular values) correspond to the high frequencies

i.e. the singular functions are similar to trigonometric functions which explains the increasing oscillations for the smaller singular values.

### 2.3 Noise in the data: for the SVE

Suppose that $g$ is measured and contaminated by errors $g = g_{\text{exact}} + \eta$ where

$$g_{\text{exact}} \in \text{range}(h) \text{ with } \|\eta\|_2 \ll \|g_{\text{exact}}\|_2$$

Immediately from the SVE

$$f = f_{\text{exact}} + \sum_{i=1}^{\infty} \frac{\langle u_i, \eta \rangle}{\mu_i} v_i(t)$$

where the second term is the contribution due to the noise.

- Noise is typically high frequency and we cannot anticipate $\eta$ to satisfy the Picard condition.
- Thus $g \notin \text{range}(h)$.
- Using the infinite sum for obtaining $f$ is unlikely to yield a useful estimate of $f_{\text{exact}}$

#### 2.3.1 Illustration of the Discrete Picard Condition

![Figure 2: Picard condition for a problem size 84. Condition, 9.3873e + 20. Ratios $|u_i^Tb|/\sigma_i$ order 1 with no noise in (a) and with small noise on the weights in (b).](image-url)
Figure 3: In 3(a)-3(c) we illustrate the Picard condition for perfect data. The singular values level out at a noise level of about $10^{-16}$ and $10^{-15}$ in panels 3(a) and 3(b)-3(c), respectively. One can see that addition of noise will then lead to a violation of the Picard condition.

### Summary

1. First kind Fredholm integral equation provides a linear model for inverse problem analysis.
2. For such models the solutions may be arbitrarily sensitive to perturbations of the data.
3. The SVE provides a means to analyse stability and existence of solutions.
4. Picard condition is necessary for existence of solution which is square integrable.
5. Right hand side $g$ must be sufficiently smooth as measured by its SVE coefficients.
6. For more general inverse problems, e.g. Laplace transform, the operator is not compact, but a similar analysis for continuum of singular values can be applied.

### 3 Connecting the Continuous and the Discrete: System of Equations

#### 3.1 Quadrature

**Quadrature - how do we obtain $A$**

Need to understand how we go from integral to matrix.

\[
\begin{align*}
\text{Integral Equation} & \quad < h, f > = g \\
\text{Discrete Form} & \quad Ax = b
\end{align*}
\]
Quadrature to evaluate the integral (finite range \([a, b] \to [0, 1]\))

\[
\int_0^1 p(t) dt = \sum_{j=1}^n \omega_j p(t_j) + E_n(p)
\]

- \(E_n\) is the error which depends on \(n\) and the function \(p\).
- \(t_j\) are the abscissae, \(\omega_j\) are weights for the rule.

For \(<h, f> = g\), \(p(t) = h(s, t)f(t)\). Thus

\[
\sum_{j=1}^n \omega_j h(s_i, t_j)f(t_j) = g(s_i) + E_n(s_i), \quad i = 1 \ldots m.
\]

Notice error depends also on the collocation point \(s_i\).

**Linear Equations**

Neglecting \(E\) and setting \(\tilde{x}\) as the approximation to \(x\) we obtain

\[
\sum_{j=1}^n \omega_j h(s_i, t_j)\tilde{x}_j = g(s_i), \quad i = 1 \ldots m.
\]

Thus defining \(A = HD\), where

- \(D\) is a diagonal matrix \(d_{jj} = \omega_j\)
- \(H_{ij} = h(s_i, t_j)\)

\[
A\tilde{x} = b
\]

We could solve for scaled \(x\) say \(\tilde{x} = D^{-1}\bar{x}\). Note this is right preconditioning.

Given \(b^{(m)}\) i.e. of length \(m\) we obtain \(x^{(n)}\) of length \(n\).

What do we use for the weights \(\omega_j\) and abscissae \(t_j\)?

Trapezium rule etc, are collocation based methods. Give values of \(f\) at discrete \(t_j\).

Expansion methods provide an expression for \(f(t)\).
Expansion Methods

Galerkin Approach

To relate the continuous and discrete we apply the expansion idea for the SVD.

SVE expands \( f \) and \( g \) in terms of basis functions and coefficients \( g_i, f_j \).

\[
g^{(m)} \in \text{span}\{\psi_1(s), \psi_2(s), \ldots, \psi_m(s)\} \quad f^{(n)} \in \text{span}\{\phi_1(s), \phi_2(s), \ldots, \phi_n(s)\}
\]

\[
g^{(m)}(s) = \sum_{i=1}^{m} g_i \psi_i(s), \quad f^{(n)}(t) = \sum_{j=1}^{n} f_j \phi_j(t) \quad \text{integrate}
\]

\[
g(s) = \int_0^1 h(s,t)f(t)dt \approx \int_0^1 h(s,t) \sum_{j=1}^{n} f_j \phi_j(t)dt := \theta(s)
\]

\( g(s) - \theta(s) \) is the residual. Galerkin approach require \( \theta(s) - g(s) \) orthogonal to \( \text{span}\{\psi_1(s), \ldots, \psi_m(s)\} \)

\[
< \psi_i(s), \theta(s) - g(s) > = 0 \quad i = 1 \ldots m
\]

Hence \( < \psi_i, \theta > = < \psi_i, g > \quad i = 1 \ldots m \) gives

\[
b_i = < \psi_i, g > = \int_0^1 \psi_i(s)g(s)ds = < \psi_i, \theta > = \sum_{j=1}^{n} f_j < \psi_i(s), \int_0^1 h(s,t)\phi_j(t)dt >
\]

\[
= \sum_{j=1}^{n} \left( \int_0^1 \int_0^1 h(s,t)\psi_i(s)\phi_j(t)dsdt \right) f_j = \sum_{ij} A_{ij}f_j.
\]

Expansion Quadrature Formula

The integration defines \( A \) and right hand side \( b \) by

\[
A_{ij} = \int_0^1 \int_0^1 h(s,t)\psi_i(s)\phi_j(t)dsdt, \quad b_i = < \psi_i, g >
\]

Requires numerical quadrature for

\[
\int_0^1 \int_0^1 h(s,t)\psi_i(s)\phi_j(t)dsdt \quad \forall(i,j), \quad b_i = \int_0^1 \psi_i(s)g(s)ds, \quad \forall i
\]

If \( h(s,t) \) is symmetric \( h(s,t) = h(t,s) \); use \( \phi_i(s) = \psi_i(s) \). Then \( A \) is symmetric.
Consider the case $\phi_i = \psi_i = \rho_i$ where $\rho_i$ is the top hat:

$$\rho_i(t) = \begin{cases} \frac{1}{\sqrt{\Delta t}} & t \in [(i - 1)\Delta t, i\Delta t] \\ 0 & \text{otherwise} \end{cases}$$

$$A_{ij} = \frac{1}{\Delta t} \int_{(i-1)\Delta t}^{i\Delta t} \int_{(j-1)\Delta t}^{j\Delta t} h(s,t) ds dt$$

$$b_i = \frac{1}{\sqrt{\Delta t}} \int_{(i-1)\Delta t}^{i\Delta t} g(s) ds$$

**Sampling**

$g$ is sampled at $s_i$, thus

$$g(s_i) = \int_0^1 \delta(s - s_i) g(s) ds$$

suggests $\psi_i(s) = \delta(s - s_i)$ and

$$A_{ij} = \int_0^1 \int_0^1 h(s,t) \delta(s - s_i) \phi_j(t) ds dt = \int_0^1 h(s_i,t) \phi_j(t) dt$$

so that the quadrature is reduced to one dimensional.

Sampling can also be implemented with the top hat and then

$$A_{ij} = \frac{1}{\sqrt{\Delta s}} \int_0^1 \phi_j(t) \left( \int_{(i-1)\Delta t}^{i\Delta t} h(s,t) ds \right) dt$$

$$b_i = \int_{(i-1)\Delta s}^{i\Delta s} g(s) ds = \sqrt{\Delta s} g(s_i).$$

### 3.2 Relationship of the Discrete SVD and Continuous SVE (for real kernels and data)

**SVD and SVE for square integrable kernel Hansen 1988**

Idea: calculate an approximate SVE numerically via the SVD.

Given the SVD how are the relevant components $u_j, v_j$ (columns of $U$ and $V$) and $\sigma_j$ related to SVE basis functions $u_i(s), v_i(t)$, singular values $\mu_i$.

Discrete matrix $A$ depends on $\psi_i, i = 1, \ldots m$ and $\phi_j, j = 1 \ldots n$.

Continuous kernel $h$ depends on $\psi_i, \phi_j, (i,j) = 1 \ldots \infty$.

Consider the approximate kernel $\tilde{h}$ which is obtained by using the discrete set $\psi_i, i = 1 \ldots n$ and $\phi_j, j = 1 \ldots n$.

The result relates SVD of $A$ to SVE of $\tilde{h}$. 

15
Singular Expansion of Degenerate Kernel

Suppose that matrix $A$ is calculated using the expansion method with functions $\psi_i, \phi_j$, $i, j = 1, \ldots n$.

Calculate its SVD: $\Sigma = \text{diag}(\sigma_i), U = (u_{ij}), V = (v_{ij})$

Let $\tilde{u}_j^{(n)}(s) := \sum_{i=1}^{n} u_{ij} \psi_i(s), \tilde{v}_j^{(n)}(t) := \sum_{i=1}^{n} v_{ij} \phi_i(t), j = 1 : n$.

**Theorem** $\sigma_j^{(n)}, \tilde{u}_j^{(n)}, \tilde{v}_j^{(n)}$ are exact singular values and functions of degenerate kernel

$$\tilde{h}(s, t) := \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \psi_i(s) \phi_j(t)$$

i.e. we have SVE for an approximate kernel - how does that relate to the exact kernel?

$$h(s, t) = \sum_{i} \mu_i u_i(s) v_i(t)$$

**Limits with $n$, $\sigma_j^{(n)}$**

SVD of $A^{(n)} = U^{(n)} \Sigma^{(n)} (V^{(n)})^T$

Error of the kernel: $\delta_n := \|h - \tilde{h}\|_2 = \|h\|_2 - \|A\|_F^2$

Note $\|\cdot\|_F^2$ is the Frobenious norm $\|A\|_F^2 = \sum_{i,j=1}^{n} a_{ij}^2$

**Singular values converge** $\sigma_i^{(n)} \leq \sigma_i^{(n+1)} \leq \mu_i, i = 1, \ldots, n$.

**Errors are bounded** $0 \leq \mu_i - \sigma_i^{(n)} \leq \delta_n, i = 1, \ldots, n$.

Hence if $\delta_n \to 0$ with $n$ increasing, approximate singular values converge uniformly to true singular values.

**SSE** (Sum of squared error) $\sum_{i=1}^{n} [\mu_i - \sigma_i^{(n)}]^2 \leq \delta_n^2$.

**Estimation** of $\delta_n$ from $\|h\|_2^2$

**Orthonormality** $\tilde{u}_i^{(n)}, \tilde{v}_i^{(n)}$ are orthonormal. Convergence

$$\max\{|\|u_i - \tilde{u}_i^{(n)}\|, |\|v_i - \tilde{v}_i^{(n)}\|\} \leq (\frac{2\delta_n}{\mu_i - \mu_{i+1}})^{1/2}$$

Practically observe that approximate singular values are more accurate than approximate singular functions.
Significance of the Result

\[ \langle \tilde{u}^{(n)}_j, g^{(n)} \rangle \] is important in the Picard condition.

\[
\langle \tilde{u}^{(n)}_j, g^{(n)} \rangle = \int_0^1 \left( \sum_{i=1}^{n} u^{(n)}_{ij} \psi_i(s) \right) \left( \sum_{k=1}^{n} b_k \psi_k(s) \right) ds \\
= \sum_{i,k} u^{(n)}_{ij} b_k \langle \psi_i, \psi_k \rangle = \sum_i u^{(n)}_{ij} b_i = u^T_j b
\]

SVD and approximate inner products are related.

i.e. the exact inner products \( \langle u_j, g \rangle, i = 1, \ldots, \) are approximated by \( \langle u^{(n)}_j, g^{(n)} \rangle \) which is immediately obtained from the SVD for \( A \).

**Discrete Picard Condition** Let \( \tau \) denote the level such that \( \forall j > r, \sigma_j \approx O(\tau), \) due to noise and rounding. The discrete Picard condition is satisfied if for \( j \leq r \) the coefficients \( |(u^{(n)}_j)^T b| \) decay faster than \( \sigma_j \).

Picard condition applies only for \( \sigma_j > O(\tau) \). It is a condition on the size of the inner products \( (u^{(n)}_j)^T b \) for \( j \leq r \).

Discrete Solution approximates Continuous Solution

<table>
<thead>
<tr>
<th>SVE Solution</th>
<th>SVD Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f(t) = \sum_j \frac{\langle u_j, g \rangle}{\mu_j} v_j(t) )</td>
<td>( \tilde{x} = \sum_{j=1}^{n} \frac{\langle u^{(n)}_j, b \rangle}{\sigma_j} v^{(n)}_j )</td>
</tr>
</tbody>
</table>

But \( \langle u^{(n)}_j, b \rangle = \langle \tilde{u}^{(n)}_j, g^{(n)} \rangle \) where \( \tilde{u}^{(n)}_j \) tends to \( u_j \) with increasing \( n \), while \( \sigma^{(n)}_j \) converges to \( \mu_j \) with \( n \).

Equivalently, if the discretization with increasing \( n \) is sufficiently good, the approximate solution obtained from the SVD is essentially independent of the discretization.

For solving the first kind Fredholm integral equation numerically, the coefficients \( (u^{(n)}_j)^T b \) and singular values \( \sigma_j \) reveal important information about the true quantities \( \langle u_j, g \rangle \) and \( \mu_j \).

Summary Approach - see Hansen for more details/examples

For increasing \( n \) until converged

1. Choose the orthonormal basis functions \( \psi_i(s) \) and \( \phi_i(t) \).
2. Calculate matrix \( A \) with entries \( a_{ij} = \langle h(s_i, t), \phi_j(t) \rangle, i, j = 1, \ldots, n. \)
3. Compute SVD of $A$
4. Estimate the singular functions $\tilde{u}_j(s)$ and $\tilde{v}_j(t)$

**Test Convergence** of set of singular values.

**End For**

**Is square integrable required for the theory?**

Consider solving for $f$ from the Laplace transform

$$g(s) = \int_0^\infty e^{-st} f(t) dt$$

Kernel $e^{-st}$ is not square integrable:

$$\int_0^a (e^{-st})^2 ds = \int_0^a e^{-2st} ds = \frac{1 - e^{-2ta}}{2t} \to \frac{1}{2t} \text{ for } a \to \infty$$

But $\int_0^\infty t^{-1}$ is infinite, $\int_0^\infty \int_0^\infty (e^{-st})^2 ds dt$ is infinite. No SVE

Now $f(t)$ bounded for $t \to \infty$ implies $g(s)$ is bounded $\forall s \geq 0$.

Truncation for large $a$ in Laplace transform introduces small error in $g$, and $g$ decays with $s$. We obtain integral equation

$$\int_0^a e^{-st} f(t) dt = g(s), \quad 0 \leq s \leq a.$$ 

Now the kernel is square integrable.

Pick $a$ and increase $n$, the SVD converges.

Pick $n$ and increase $a$, the SVD does not converge. Demonstrates the lack of SVE for the Laplace Transform.

On the other hand, we can see convergence when the domain is not infinite!

Note that the analysis is totally determined by the kernel. But you can look at the coefficients for the expansion in $g$ by picking a function with known Laplace transform!
The Discrete Solution with the SVD Feb 3, 2014

The thin SVD Solution

Consider general overdetermined discrete problem

\[ Ax = b, \quad A \in \mathbb{R}^{m \times n}, \quad b \in \mathbb{R}^m, \quad x \in \mathbb{R}^n, \quad m \geq n. \]

Thin singular value decomposition (SVD) of rectangular \( A \) is

\[ A = U \Sigma V^T = \sum_{i=1}^{n} u_i \sigma_i v_i^T, \quad \Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n). \]

\( U \) of size \( m \times n \), \( V \) and \( \Sigma \) square of size \( n \):

\[ U = [u_1, \ldots, u_n], \quad V = [v_1, \ldots, v_n], \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0 \]

Orthonormal columns in \( U \) and \( V \), left and right singular vectors for \( A \)

\[ u_i^T u_j = v_i^T v_j = \delta(i - j) \rightarrow U^T U = V^T V = VV^T = I_n. \]

If \( A \) has full column rank \( \sigma_n > 0 \)

\[ A^\dagger = V \Sigma^{-1} U^T = A^{-1}, \quad m = n. \]

Moore Penrose Generalized Inverse

1. \( AA^\dagger A = A \)
2. \( A^\dagger AA^\dagger = A^\dagger \)
3. \( (AA^\dagger)^\ast = AA^\dagger \)
4. \( (A^\dagger A)^\ast = A^\dagger A \)

Expansion Solution

1. We can write \( x = VV^T x = \sum_{i=1}^{n} (v_i^T x) v_i \) and

\[ Ax = \sum_{i=1}^{n} (v_i^T x) A v_i \]

2. But \( A v_i = U \Sigma V^T v_i = U \Sigma e_i = \sigma_i u_i \). Thus

\[ Ax = \sum_{i=1}^{n} \sigma_i (v_i^T x) u_i \]

19
3. Similarly, for \( m = n \), \( b = \sum_{i=1}^{n} (u_i^T b) u_i \).

4. Immediately compare coefficients and obtain \( \sigma_i(v_i^T x) = u_i^T b, \ i = 1, \ldots, n \) and

\[
x = \sum_{i=1}^{n} \frac{(u_i^T b)}{\sigma_i} v_i
\]

5. Sensitivity of solutions depends on \( \text{cond}(A) = \sigma_1/\sigma_n \)

**Full SVD: Rectangular \( A \)**

- Let \( U = [U_1, U_2] \) be square of size \( m \), \( \Sigma \) rectangular of size \( m \times n \):

\[
A = U \Sigma V^T = [U_1, U_2] \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} V^T
\]

- The inverse is replaced by the pseudo inverse: if \( A \) has rank \( r \)

\[
A^\dagger = \sum_{i=1}^{r} \frac{1}{\sigma_i} v_i u_i^T
\]

- Solution of the LS problem is given by

\[
x = \sum_{i=1}^{r} \frac{(u_i^T b)}{\sigma_i} v_i
\]

- Sensitivity of solution depends on the condition as measured by \( \sigma_1/\sigma_r \).

- Recall singular values relation to eigenvalues \( \lambda_i \) of \( A^T A \), \( \sigma_i^2 = \lambda_i \)

**4.1 Filtering and Regularization (February 3, 2014)**

**4.2 Solution for Noisy Data**

- Denote noise by \( n \)

- Spectral decomposition acts on the noise term in the same way it acts on the exact right hand side \( b \). e.g.

\[
x = \sum_{i=1}^{r} \frac{(u_i^T (b_{\text{exact}} + n))}{\sigma_i} v_i = x_{\text{exact}} + \sum_{i=1}^{r} \frac{(u_i^T n)}{\sigma_i} v_i
\]

- If \( n \) is uniform, anticipate \( |u_i^T n| \) of similar magnitude \forall i. 
• Can only recover components that arise from $|u_i^Tb|$ greater than the noise level.

• But anticipate $\sigma_i \to 0$. $\sigma_i$ small represents high frequency component in the sense that $u_i, v_i$ have more sign changes as $i$ increases.

• $(u_i^Tn/\sigma_i)$ is the coefficient of $v_i$ in the error image.

• If $1/\sigma_i$ large the contribution of the high frequency error is magnified due to $(u_i^Tn/\sigma_i)$.

**Estimating rank of $A$ for implementing the TSVD**

• Results suggest that we need information on SVD of $A$

• Also need information on the spread of the singular values.

• Ideally information on the noise level in the data is available.

• Practically we need the numerical rank of $A$.

• Practically it is not always viable to find the effective numerical rank

• We turn to other methods to find acceptable solutions.

### 4.3 The Filtered SVD - more general than truncation

The truncated SVD is a special case of spectral filtering.

Recall $x = A^\dagger b = V\Sigma^\dagger U^T b$.

The filtered solution is given by

$$x_{\text{filt}} = \sum_{i=1}^{r} \gamma_i \left( \frac{u_i^T b}{\sigma_i} \right) v_i = V\Sigma_{\text{filt}}^\dagger U^T b, \quad \Sigma_{\text{filt}}^\dagger := \text{diag}(\frac{\gamma_i}{\sigma_i}, 0_{m-r})$$

i.e

$$x = V\Gamma\Sigma^\dagger U^T b,$$

where $\Gamma$ is the diagonal matrix with entries $\gamma_i$.

Notice again the relationship with the SVE - filter out the terms which are noise contaminated.

$\gamma_i \approx 1$ for large $\sigma_i$, $\gamma_i \approx 0$ for small $\sigma_i$

Spectral filtering is used to filter the components in the spectral basis, such that noise in signal is damped.

How to chose filter factors $\gamma_i$?

Truncated SVD takes $\gamma_i = 1, 1 \leq i \leq k$ and 0 otherwise to obtain solution $x_k$. 

4.4 Tikhonov Regularization - Filtering

Consider general overdetermined discrete problem

\[ Ax = b, \quad A \in \mathbb{R}^{m \times n}, \quad b \in \mathbb{R}^m, \quad x \in \mathbb{R}^n, \quad m \geq n. \]

Fit to data functional of the least squares problem is \( \|Ax - b\|_2^2 \)

Define \( x_{LS} = \arg\min_x \|Ax - b\| \) and \( \hat{x} \) by \( A\hat{x} = \hat{b}, i.e. \hat{b} \in \text{Range}(A) \)

We know that \( x_{LS} \) is noise contaminated if \( A \) is ill-conditioned.

Add a penalty term with regularization parameter \( \lambda > 0 \)

\[ x_\lambda = \arg\min_x \{ \|Ax - b\|^2 + \lambda^2 \|x\|^2 \} \]

Regularized solution trades \( \|x_\lambda\|^2 \) against \( \|b - Ax_\lambda\|^2 \).

For small \( \lambda \) the fit to data term is more closely enforced and the solution may be noisy

For larger \( \lambda \) the regularization term is enforced and so the solution is smoothed.

Figure 4: In (a) example: noise in \( b \) is \( n \sim N(0, 10^{-7}) \) (normally distributed, mean 0 and variance \( 10^{-7} \))
Solution for Different Choices of $\lambda$

Figure 5: Solutions $x(\lambda)$
Signal to Noise Ratio for Different Choices of $\lambda$

Figure 6: The Signal to Noise Ratio of the solution $10\log_{10} \|\hat{x}\|/\|\hat{x} - x(\lambda)\|$, true solution $\hat{x}$. Note of course this can’t be calculated for practical data.
4.5 Investigating the Regularized solution

\[ x_\lambda = \arg\min_x \{ \|Ax - b\|^2 + \lambda^2\|x\|^2 \} \]

has equivalent formulation

\[ x_\lambda = \arg\min_x \left\| \begin{pmatrix} A \\ \lambda I_n \end{pmatrix} x - \begin{pmatrix} b \\ 0_n \end{pmatrix} \right\|^2 \]

We need to know how to find \( \lambda \): We can use the heuristic based on the L-Curve: Plot regularization term against the fidelity term for \( \lambda \)

\[ \log(\|x_\lambda\|), \log(\|Ax_\lambda - b\|) \]

Figure 7: On the left a corner and on the right no corner. The L-curve is expensive for general matrices \( A \), but is very general and straightforward. It does not consider any information on the noise structure

4.6 Properties of the regularized solution

**Normal equations** Theoretically the solution solves the equations which are obtained by taking the derivative with respect to \( x \) Make sure you can prove this!

\[ (A^TA + \lambda^2 I)x_\lambda = A^Tb \]

It follows immediately from the equivalent formulation:

\[ x_\lambda = \arg\min_x \left\| \begin{pmatrix} A \\ \lambda I_n \end{pmatrix} x - \begin{pmatrix} b \\ 0_n \end{pmatrix} \right\|^2 \]

**Regularization matrix**

\[ x_\lambda = R(\lambda)b \text{ where } R(\lambda) = (A^TA + \lambda^2 I)^{-1}A^T \]

Matrix \( R \) is also sometimes referred to as the *generalized inverse* or *regularized inverse*. Denoted by \( A^\dagger \) in Vogel.
Model error (not computable)

\[ e(\lambda) = \hat{x} - x_{\lambda} = \hat{x} - R(\lambda)(b) \]
\[ = \hat{x} - R(\lambda)(\hat{b} + n) = \hat{x} - R(\lambda)A\hat{x} - R(\lambda)n \]
\[ = \text{bias} \quad \text{variance} \]
\[ = (I_n - VTV^T)\hat{x} - V\Sigma^\dagger U^Tn \]

Note that exact values \( \hat{x} \) and \( \hat{b} \) not available and thus we investigate the solution through known computable quantities.

**Bias** is the loss of information introduced by regularization - also called the **Regularization error**

Express \( \hat{x} = VV^T\hat{x} \) and rewrite \( (I_n - VTV^T)\hat{x} \) then

\[ (I_n - VTV^T)\hat{x} = V(I_n - \Gamma)V^T\hat{x} = \sum_{i=1}^{n} (1 - \gamma_i)(v_i^T\hat{x})v_i \]
\[ = \left\{ \begin{array}{ll}
\sum_{i=k+1}^{n} (v_i^T\hat{x})v_i & \rightarrow 0 \text{ as } k \rightarrow n \quad \Gamma_{\text{TVD}} = \text{diag}(I_k, 0_{n-k}) \\
\sum_{i=1}^{n} \frac{\lambda^2}{\sigma_i^2 + \lambda^2} (v_i^T\hat{x})v_i & \rightarrow 0 \text{ as } \lambda \rightarrow 0 \quad \Gamma_{\text{Tikh}} = \text{diag}(\gamma_i), \quad \gamma_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} 
\end{array} \right. \]

Now for both Tikhonov and TSVD we observe the form for \( \Gamma \) leads to the filtering of the bias. It is the error due to using \( \Gamma = \Sigma_{\text{filt}} \) in place of \( \Sigma \).

**Variance** is the amplification of the noise in the error and also tends to zero by appropriate choice of \( \lambda \). It is also called the **perturbation error** and is the inverted and filtered noise, consistently zero if \( \Gamma = 0 \). First express the variance of solving for right hand side with noise \( n \) as a sum

\[ V\Sigma^\dagger U^Tn = \sum_{i=1}^{n} (u_i^Tn)\frac{\gamma_i}{\sigma_i}v_i = \sum_{i=1}^{n} (u_i^Tn)\frac{\sigma_i^2}{(\lambda^2 + \sigma_i^2)\sigma_i}v_i \]

Suppose in TSVD \( k \) is chosen by a threshold so that for \( i \leq k \) \( \gamma_i/\sigma_i \leq 1/\lambda \), then equivalent we assume for \( i > k \) \( \gamma_i/\sigma_i = 0 \leq 1/\lambda \).

For the Tikhonov filter observe \( \gamma_i/\sigma_i = (\sigma_i^2/(\sigma_i^2 + \lambda^2))/\sigma_i = (\sigma_i + \lambda^2/\sigma_i)^{-1} \)

- For \( \sigma_i^2 > \lambda^2, \sigma_i > \lambda \) and thus \( \sigma_i + \lambda^2/\sigma_i > \lambda \)
- For \( \sigma_i^2 \leq \lambda^2, 1/\sigma_i > 1/\lambda, \) so \( \lambda^2/\sigma_i > \lambda \) and \( \sigma_i + \lambda^2/\sigma_i > \lambda \).

Hence in each case \( \gamma_i/\sigma_i = (\sigma_i + \lambda^2/\sigma_i)^{-1} \leq 1/\lambda \) and \( \gamma_i \leq \sigma_i/\lambda \) for all \( i \)
Suppose that $\|n\|_2^2 < \delta^2$ then by orthogonality of columns of $V$

$\left\| \sum_{i=1}^{n} (u_i^T n) \gamma_i / \sigma_i v_i \right\|_2^2 = \sum_{i=1}^{n} \left| (u_i^T n) (\gamma_i / \sigma_i)^2 \right| < (\delta / \lambda)^2$.

If $\lambda = \delta^p$ where $p < 1$ then as $\delta \to 0$ variance error goes to zero. Notice that we use $\|U^T n\|^2 = \|n\|^2$ for $U$ orthogonal, $
\lambda$ can be chosen so that the variance goes to zero for both TSVD and Tikhonov

4.6.1 Assignment Feb 4, 2014

This continues the phillips assignment from class using the Regularization toolbox.

1. Write a Matlab script which solves the problem phillips of size 128 using the SVD for the matrix $A$ from the phillips problem.

2. Add noise to the right hand side data using the Matlab function randn: $b_{\text{noisy}} = b + \zeta n$, where $n = \text{randn}(128)$ and $\zeta$ determines the noise level. Try $\zeta = .1$.

3. Explore the solutions of the noisy problem: giving $x_{\text{noisy}}^{\text{Tik}}(\lambda)$ for a range of $\lambda$.

4. Plot your own L-curve for your solutions.

5. Find the corner of the L-curve and plot the optimal solution as compared to the exact solution.

6. Look at the trace of the resolution matrix as a function of $\lambda$. Plot Trace against $\lambda$. Identify on your plot the optimal point as indicated by the L-curve analysis. What do you observe?

$A(\lambda) = AA^T = A(A^T A + \lambda^2 I)^{-1} A^T$

First derive an expression for the Trace of $A(\lambda)$ using the SVD for $A$. You can substitute the expression in your m file using latex and publish.

7. Submit under the assignment in Blackboard

Review February 10, 2014

We review the notation and qualitative measures of the solution of the least squares problem

**Regularization matrix** $R(\lambda) = (A^T A + \lambda^2 I)^{-1} A^T = V \Sigma^T U^T$ and $R(\lambda) A = V \Sigma^T$

**Filter Factors** $\gamma_i = \frac{\sigma_i^2}{\lambda^2 + \sigma_i^2}$ all satisfy $\gamma_i / \sigma_i \leq \lambda$.

**Variance** $R(\lambda)n = V \Sigma^T U^T n$ - amplification of noise in measurement error: called **Perturbation error**. already shown that $\lambda$ can be chosen so that the variance goes to zero.
Solution $x = V\Gamma\Sigma^T U^T b$

Bias $(I_n - R(\lambda)A)\hat{x} = (I_n - VTV^T)\hat{x}$ - loss of information called **Regularization error**. We can estimate the bias using the expression for $x$ and using the orthogonality of $V$ to write $\hat{x} = VV^T \hat{x}$

$$
\| (I_n - VTV^T) x \|_2^2 = \| (I_n - \Gamma) V^T x \|_2^2 = \| (I_n - \Gamma) \Gamma \Sigma^T U^T b \|_2^2
$$

$$
= \sum_{i=1}^n ((1 - \gamma_i) \gamma_i - \sigma_i)^2 = \sum_{i=1}^n \frac{\lambda^2}{\sigma_i^2 + \lambda^2} \frac{\sigma_i^2}{\sigma_i} \left( \frac{u_i^T b}{\sigma_i} \right)^2
$$

We notice that the bias depends on products $(1 - \gamma_i) \gamma_i$ with the weighted expansion coefficients $u_i^T b / \sigma_i$.

- First $|\frac{u_i^T b}{\sigma_i}|$ decays on average by Picard condition
- For small $i$ ($\sigma_i$ big) $\gamma_i \approx 1$, and $(1 - \gamma_i) \gamma_i \approx 0$. Little error from large $\left| \frac{u_i^T b}{\sigma_i} \right|$.
- For large $i$ $(1 - \gamma_i) \approx 1$ and $(1 - \gamma_i) \gamma_i \approx 0$, also provides little damping of smaller $\left| \frac{u_i^T b}{\sigma_i} \right|$.

Indeed the choice of $\Gamma$ controls size of the bias. For Tikhonov regularization we also have

$$
\gamma_i = \begin{cases} 
1 - (\frac{\lambda}{\sigma_i})^2 + O((\frac{\lambda}{\sigma_i})^4) & \sigma_i \gg \lambda \\
(\frac{\sigma_i}{\lambda})^2 + O((\frac{\sigma_i}{\lambda})^4) & \sigma_i \ll \lambda 
\end{cases}
$$

Hence for $\lambda \in [\sigma_r, \sigma_1]$, $\gamma_i \approx 1$ for small $i$, and $\gamma_i \approx (\sigma_i / \lambda)^2$ for large $i$ (small $\sigma_i$).

**Conclude** Parameter $\lambda$ controls the filtering. If $\lambda \approx \gamma_k$, then filtered solution does not include components related to $\sigma_{k+1} \ldots \sigma_r$.

**Moreover** it is sensible to keep $\lambda \in [\sigma_r, \sigma_1]$.

### 4.7 How do we find $\lambda$?

First we introduce some measures of a solution"  

**Predictive error (not computable)** requires $e$

$$
p(\lambda) = Ae(\lambda) = Ax(\lambda) - A\hat{x} = A R(\lambda) b - \hat{b}
$$

Notice that it depends on the matrix $AR(\lambda) := A(\lambda)$

**Define Influence Matrix or Resolution Matrix** $A(\lambda) = AR(\lambda)$.

$$
A(\lambda) = AA^T = A(A^T A + \lambda^2 I)^{-1} A^T
$$

**Regularized Residual or Predictive Risk is computable**

$$
p(\lambda) = Ax(\lambda) - A\hat{x} \approx (Ax(\lambda) - b) = (A(\lambda) - I_m) b := r(\lambda)
$$

It can be used to estimate the error through the unbiased estimator:
4.7.1 Derivation of the Unbiased Predictive Risk for Estimating $\lambda$

Consider the predictive error - cannot be calculated

$$p(\lambda) = AR(\lambda)b - \hat{b} = A(\lambda)(\hat{b} + n) - \hat{b}$$

$$= (A(\lambda) - I_m)\hat{b} + A(\lambda)n$$

$$= \text{deterministic} \quad \text{stochastic}$$

Consider the predictive risk - can be calculated

$$r(\lambda) = (A(\lambda) - I_m)b$$

$$= (A(\lambda) - I_m)\hat{b} + (A(\lambda) - I_m)n$$

$$= \text{deterministic} \quad \text{stochastic}$$

Both expressions use the noise $n$.

Necessary Statistical Results

**Mean-Variance** Suppose random vector $x$ has mean $x_0$, covariance-variance matrix $\Sigma$.

- we say $x \sim (x_0, \Sigma)$
- Then measured $b$ for the exact model $b_0 = Ax_0$, which is noise contaminated by noise $n$ with variance $C_b$ of mean 0 satisfies $b \sim (Ax_0, A\Sigma A^T + C_b)$

**Trace Operator** is linear. $\text{trace}(A + B) = \text{trace}(A) + \text{trace}(B)$ and $\text{trace}(A^T) = \text{trace}(A)$. We note also the cyclic property $\text{trace}(ABC) = \text{trace}(CAB)$, provided that dimensions are consistent.

**Definition 5** (Discrete White Noise Vector). A random vector $n = (\eta_1, \eta_2, \ldots, \eta_n)$ is a discrete white noise vector provided that $E(n) = 0$ and $\text{cov}(n) = \sigma^2 I_n$. i.e.

$$E(\eta_i) = 0, \quad E(\eta_i \eta_j) = \sigma^2 \delta_{ij}$$

$\sigma^2$ is the variance of the white noise

**Lemma 1** (Trace Lemma). Let $y$ be deterministic and $n$ a discrete white noise vector with variance $\sigma^2$. Then

$$E(\|y + An\|^2) = \|y\|^2 + \sigma^2 \text{trace}(A^TA)$$
Obtaining the Estimate

Use the Trace lemma and assume that the noise vector $n$ is a discrete white noise vector. Estimate mean predictive error from $E(\|p(\lambda)\|^2)$ and $E(\|r(\lambda)\|^2)$

$$E(\|r(\lambda)\|^2) = E(\|(A(\lambda) - I_m)\hat{b} + (A(\lambda) - I_m)n\|^2) = \|(A(\lambda) - I_m)\hat{b}\|^2 + \sigma^2\text{trace}((A(\lambda) - I_m)^T(A(\lambda) - I_m))$$

$$E(\|p(\lambda)\|^2) = E(\|(A(\lambda) - I_m)\hat{b} + A(\lambda)n\|^2) = \|(A(\lambda) - I_m)\hat{b}\|^2 + \sigma^2\text{trace}(A(\lambda)^T A(\lambda))$$

$$= E(\|r(\lambda)\|^2) + \sigma^2\text{trace}(A(\lambda)^T A(\lambda)) - \sigma^2\text{trace}((A(\lambda) - I_m)^T(A(\lambda) - I_m))$$

$$= E(\|r(\lambda)\|^2) + \sigma^2(2\text{trace}(A(\lambda)) - m) \text{ by linearity of trace}$$

$$\approx \|r(\lambda)\|^2 + \sigma^2(2\text{trace}(A(\lambda)) - m) := U(\lambda)$$

Notice that $E(U(\lambda)) = E(\|p(\lambda)\|^2)$ so that $U$ is an unbiased estimator.

Thus seek $\lambda$ such that $U$ is minimum:

$$\lambda = \arg \min_{\lambda} U(\lambda).$$

An Example with UPRE

Figure 8: Notice the well-defined minimum. But requires the calculation of the trace
How do we use UPRE: First with the SVD rewrite \( A(\lambda) - I_m \)

\[
A(\lambda) - I_m = A(A^T A + \lambda^2 I_n)^{-1} A^T - I_m \\
= U\Sigma V^T (V\Sigma^T U^T U\Sigma V^T + \lambda^2 V V^T)^{-1} V\Sigma^T U - UU^T \\
= U(\Sigma(\Sigma^T \Sigma + \lambda^2 I_n)^{-1} \Sigma^T - I_m)U^T \\
\text{Hence with } \sigma_i = 0, i > n
\]

\[
\text{trace}(A(\lambda)) = \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \text{ and } (A(\lambda) - I_m)b = \sum_{i=1}^{m} (u_i^T b) \frac{-\lambda^2}{\sigma_i^2 + \lambda^2} u_i. \text{ Thus}
\]

\[
\|r(\lambda)\|^2 = \sum_{i=1}^{m} |u_i^T b|^2 \left( \frac{\lambda^2}{\sigma_i^2 + \lambda^2} \right)^2 \text{ and}
\]

\[
U(\lambda) = \sum_{i=1}^{m} |u_i^T b|^2 \left( \frac{\lambda^2}{\sigma_i^2 + \lambda^2} \right)^2 + \sigma^2 \left( 2 \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} - m \right).
\]

Entries \(|u_i^T b|\) and \(\sigma_i\) are calculated once. Hence given the SVD the cost is basically \(O(n)\) for each \(\lambda\). Basic idea is to bracket the minimum and then minimize within the bracket.


### 4.7.2 Generalized Cross Validation: GCV February 11, 2014

This looks rather like the UPRE but requires minimization of a rational functional:

Based on omitting a data value and testing predictability of solution for this missing value: \(\lambda\) is chosen to minimize

\[
G(\lambda) = \frac{\|r(\lambda)\|^2_2}{(\text{trace}(I_m - A(\lambda)))^2}. 
\]

GCV provides another estimate of the predictive risk: (see e.g. Vogel)

There are the same difficulties of using the trace for large scale problems.

Often \(G\) is relatively flat near the minimum, or has multiple minima and thus difficult to apply.
4.7.3 Why not just use $r(\lambda)$ - rather than deal with the UPRE or GCV: The discrepancy principle

Calculate $r(\lambda) = b - Ax(\lambda)$, noting that

$$b = \sum_i (u_i^T b)u_i,$$

$$Ax(\lambda) = \sum_{i=1}^{m} (\sigma_i^2)/((\sigma_i^2 + \lambda^2))(u_i^T b)u_i$$

$$r(\lambda) = \sum_{i=1}^{m} (\lambda^2)/(\sigma_i^2 + \lambda^2)(u_i^T b)u_i$$

Suppose that $x_\lambda \approx \hat{x}$ then $r(\lambda) \approx b - A\hat{x} = n \in \mathcal{R}^m$ for $b$ of length $m$.

Hence $E(\frac{1}{m}\|Ax_\lambda - b\|^2) \approx E(\frac{1}{m}\|A\hat{x} - b\|^2) \approx E(\frac{1}{m}\|n\|^2) = \sigma^2$, where $n \sim (0, \sigma^2 I)$.

**Discrepancy Principle** Find $\lambda$ such that $\|r(\lambda)\|^2 \approx m\sigma^2$.

Is $x_\lambda \approx \hat{x}$ reasonable? Consider

$$F(\lambda) = \|r(\lambda)\|^2 = \sum_{i=1}^{m} (\lambda^2/(\sigma_i^2 + \lambda^2))^2|u_i^T b|^2$$

Clearly $F(\lambda)$ is continuous, $F(0) = 0$ and $F(\lambda) \to \|b\|^2$ for $\lambda \to \infty$. Hence there exists unique $\lambda$ such that $F(\lambda) = \delta^2$, where $\delta = \|n\|$, provided $\delta < \|b\|$.

We note the regularized functional is given by

$$J(x) = \|Ax - b\|^2 + \lambda^2\|x\|^2$$

has solution $x_\lambda$ as $\text{argmin}$ of the functional that satisfies for any other $x$

$$J(x_\lambda) \leq J(x)$$
Now take $\lambda_*$ such that $F(\lambda_*) = \delta^2$ and note
\[
\delta^2 + \lambda_*^2 \| x(\lambda_*) \|^2 = F(\lambda_*) + \lambda_*^2 \| x(\lambda_*) \|^2 = J(\lambda_*) \leq J(\hat{x}) = \| A \hat{x} - b \|^2 + \lambda_*^2 \| \hat{x} \|^2 = \| A \hat{x} - \hat{b} - n \|^2 + \lambda_*^2 \| \hat{x} \|^2 = \| n \|^2 + \lambda_*^2 \| \hat{x} \|^2 = \delta^2 + \lambda_*^2 \| \hat{x} \|^2
\]
i.e. the solution of that satisfies the discrepancy principle is necessarily smoother than the mean solution (exact solution).

### 4.7.4 Some observations: Parameter Choice is Difficult

<table>
<thead>
<tr>
<th>L-curve</th>
<th>Discrepancy</th>
<th>UPRE</th>
<th>GCV</th>
</tr>
</thead>
<tbody>
<tr>
<td>• No unique solution (depends on finding the L)</td>
<td>• Unique solution</td>
<td>• No unique solution</td>
<td>• Multiple minima or flat</td>
</tr>
<tr>
<td>• Requires multiple solves to find appropriate range for the corner</td>
<td>• Direct solve using Newton method</td>
<td>• Gives good estimates</td>
<td>• Minimization required</td>
</tr>
<tr>
<td>• Does not use noise level</td>
<td>• Requires noise level</td>
<td>• Requires noise level</td>
<td>• Needs trace estimator</td>
</tr>
<tr>
<td>• Easily justified.</td>
<td>• Leads to a smoothed solution</td>
<td>• Minimization required</td>
<td></td>
</tr>
</tbody>
</table>

### Assignment 4 Feb 11, 2014 Due Feb 18

This continues the phillips assignment from class using the Regularization toolbox.

1. Now that you have the Matlab script to solve the phillips problem using the L-curve you can easily modify the script to compare the solution using
   - GCV
   - UPRE
   - Discrepancy principle

2. To obtain the solution by the GCV (UPRE) or the Discrepancy principle sweep through a range of values for $\lambda$ and pick the solution which minimizes the GCV (UPRE) functional and that finds the root for the discrepancy principle.

3. Plot the relevant functionals, use subplot.

4. Plot the solutions obtained from the optimal $\lambda$.

5. Repeat for at least two noise levels.

6. Comment

7. Submit under the assignment in Blackboard
5 Interpretation: Rank and resolution February 18 2014

We need the following basic linear algebra result:

**Theorem 5.1.**

\[
\begin{align*}
\mathbb{N}(A^T) & \perp \mathbb{R}(A) \\
\mathbb{N}(A^T) + \mathbb{R}(A) & = \mathbb{R}^m
\end{align*}
\]

**Proof.** Suppose that \( y \) is any vector in the null space of the matrix \( A^T \): \( A^T y = 0 \). Thus \( a_i^T y = y^T a_i \) where \( a_i \) is the \( i \)th column of \( A \), i.e. \( y \) is perpendicular to all the columns of \( A \). But each \( a_i = A e_i \) is in the range of \( A \) and any element in the range of \( A \) is a linear combination of these columns. Thus \( y \) is perpendicular to the range of \( A \) giving (5).

(6) is also immediate. In particular suppose that \( y \in \mathbb{R}^m \), then there is a unique decomposition \( y = z + x^\dagger \), \( z \in \mathbb{N}(A^T) \) and \( x^\dagger \in \mathbb{R}(A) \). \( \square \)

5.1 The pseudo inverse solution of \( Ax \approx b \)

We return to a brief discussion of the SVD for the solution of the problem

\[
Ax \approx b, \quad A \in m \times n, \quad \text{rank}(A) = p \leq \min\{m, n\}
\]

\[
A = U \Sigma V^T, \quad \Sigma = [\Sigma_p, 0_p \times (n-p); 0_{(m-p) \times n}], \quad \Sigma_p = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p).
\]

The terms of the SVD can be further expanded:

\[
U = [u_i] = [U_p, U_0] \\
V = [v_i] = [V_p, V_0]
\]

from which we obtain by orthogonality

\[
U^T U = I_m = [U_p, U_0]^T [U_p, U_0] = [U_p^T, U_0^T] [U_p, U_0] = \text{diag}(U_p^T U_p, U_0^T U_0) = \text{diag}(I_p, I_{m-p})
\]

\[
U U^T = I_m = [U_p, U_0] [U_p^T, U_0^T] = U_p U_p^T + U_0 U_0^T.
\]

with equivalent expressions for the orthogonal matrix \( V \). Using the decomposition of \( U, V \) the SVD can be written in **compact** form

\[
A = [U_p, U_0] [\Sigma_p, 0_p \times (n-p); 0_{(m-p) \times n}] [V_p, V_0] = U_p \Sigma_p V_p^T.
\]

Thus if \( b \) is a vector in the range of \( A \), for some \( x \)

\[
b = Ax = U_p \Sigma_p V_p^T x = U_p (\Sigma_p V_p^T x) = \sum_{i=1}^p u_i z_i, \quad z = \Sigma_p V_p^T x,
\]
and we see that any such \( b \) is a linear combination of the first \( p \) columns \( u_i \) of \( U \), so that \( U_p \) spans the range of \( A = \mathbb{R}(A) \). Moreover, by Thm. 5.1 \( U_0 \) spans \( \mathbb{N}(A^T) \). Similarly, using \( A^T = V_p \Sigma_p U_p^T \), we obtain that the first \( p \) columns \( v_i \) of \( V \) span the range of \( A^T \) and \( V_0 \) spans \( \mathbb{N}(A) \).

From the compact form of the SVD we obtain

\[
U_p \Sigma_p^T V_p^T x = b \\
V_p^T x = \Sigma_p^{-1} U_p b.
\]

But \( x \in \mathbb{R}^n \) which is spanned by \( \mathbb{N}(A) + \mathbb{R}(A^T) = \text{span} < v_{p+1}, \ldots, v_n > + \text{span} < v_1, \ldots, v_p > \). Thus we may take \( x = V_0 \alpha + V_p \beta = z + x^\dagger \) for some vectors \( \alpha, \beta, z \) and \( x^\dagger \). Therefore

\[
V_p^T x = V_p^T (V_0 \alpha + V_p \beta) = \beta = \Sigma_p^{-1} U_p b \\
V_p (\Sigma_p^{-1} U_p \beta) = V_p \beta = x^\dagger
\]

yielding \( x^\dagger \) as the pseudo inverse solution for the equations and defining the pseudo inverse

\[
A^\dagger = V_p \Sigma_p^{-1} U_p^T.
\]

Clearly the solution is nonuniquely given by

\[
x = x^\dagger + z = (V_p \Sigma_p^{-1} U_p^T) b + z \quad \text{where} \quad z \in \mathbb{N}(A).
\]

(8) leads to the following general conclusions about the solution of (7).

**Square:** \( m = n = p \). The null spaces \( \mathbb{N}(A) \) and \( \mathbb{N}(A^T) \) are trivial. Moreover \( V_p = V, \Sigma_p = \Sigma, U_p = U \) and \( z = 0 \). Thus we obtain the solution of the square system of equations

\[
x = V \Sigma^{-1} U^T b = A^\dagger b.
\]

**Overdetermined full column rank** \( p = n < m \) \( \mathbb{N}(A) \) is trivial, \( z = 0 \) and \( x \) is uniquely determined as the solution of the normal equations:

\[
(A^T A)^{-1} A^T = (V \Sigma^T U^T U \Sigma V^T)^{-1} (V \Sigma^T U^T) = V (\Sigma^T \Sigma)^{-1} \Sigma^T U^T \\
= V (\Sigma_p^T \Sigma_p)^{-1} [\Sigma_p, 0] [U_p^T; U_0] = V \Sigma_p^{-1} U_p = V \Sigma_p^{-1} U_p^T = A^\dagger
\]

**Underdetermined full row rank** \( p = m < n \) \( \mathbb{N}(A^T) \) is trivial, and \( U_p^T = U^{-1} \). Applying \( A \) to \( x^\dagger \)

\[
(U \Sigma V^T)(V_p \Sigma_p^{-1} U_p^T b) = U \Sigma [V_p^T; V_0^T] V_p \Sigma_p^{-1} U_p^{-1} b \\
= U \Sigma [I_p; 0] \Sigma_p^{-1} U_p^{-1} b = U [\Sigma_p, 0] \Sigma_p^{-1}, 0] U_p^{-1} b \\
= U (I_p) U^{-1} b = b.
\]

35
Thus the pseudo inverse gives a solution \( \mathbf{x}^\dagger \) which exactly fits the data, but is non unique because any vector \( \mathbf{z} \in \mathbf{N}(A) \) can be added to the pseudo inverse solution \( \mathbf{x}^\dagger \). Consider the non unique solution \( \mathbf{x} = \mathbf{z} + \mathbf{x}^\dagger \) then

\[
\|\mathbf{x}\|^2 = \|\mathbf{z}\|^2 + \|\mathbf{x}^\dagger\|^2,
\]

because \( \mathbf{z}^T \mathbf{x}^\dagger = 0 \)

\[
\|\mathbf{x}\|^2 \geq \|\mathbf{x}^\dagger\|^2
\]

(9)

and \( \mathbf{x}^\dagger \) is the solution of **minimum length** that satisfies the equations. Moreover, observe that

\[
A^T(AA^T)^{-1} = (V_p \Sigma_p^T U_p^T)(U_p \Sigma_p V_p^T V_p \Sigma_p^T U_p^T)^{-1}
\]

\[
= V_p \Sigma_p^T U^T (U \Sigma_p \Sigma_p^T U^T)^{-1} = V_p \Sigma_p^T (\Sigma_p \Sigma_p^T)^{-1} U^{-1}
\]

\[
= V_p \Sigma_p^{-1} U_p^{-1} = A^\dagger,
\]

from which we see that \( \mathbf{x}^\dagger \) is given by

\[
\mathbf{x}^\dagger = A^T(AA^T)^{-1} \mathbf{b} = A^\dagger \mathbf{b}
\]

**Loss of rank:** \( p < \min\{m,n\} \). It is clear that both null spaces are now non trivial. Consider

\[
A \mathbf{x}^\dagger = (U_p \Sigma_p V_p^T)(V_p \Sigma_p^{-1} U_p^T) \mathbf{b}
\]

\[
= (U_p U_p^T) \mathbf{b} = \mathbf{b}_{P(A)}
\]

where \( \mathbf{b}_{P(A)} \) denotes the projection of \( \mathbf{b} \) onto \( \mathbb{R}(A) \). Hence \( \mathbf{x}^\dagger \) is a least squares solution of (7). Moreover (8) sill applies and again as for (9) the solution has minimum length.

### 5.1.1 Properties of the pseudo-inverse solution

It is clear from the cases with loss of rank, \( p < \min\{m,n\} \) or insufficient sampling \( p = m < n \), the solution \( \mathbf{x}^\dagger \) obtained is not a unique solution of the system of equations.

**Solution covariance** \( \mathbf{x}^\dagger = A^\dagger \mathbf{b} \) and thus when \( \mathbf{b} \sim (0,C_b) \), \( \mathbf{x}^\dagger \sim (0, A^\dagger C_b (A^\dagger)^T) \).

\[
C_{x^\dagger} = A^\dagger C_b (A^\dagger)^T = V_p \Sigma_p^{-1} U_p^T C_b U_p \Sigma_p^{-1} V_p^T
\]

\[
C_b = \sigma^2 I \Rightarrow C_{x^\dagger} = \sigma^2 V_p \Sigma_p^{-1} \Sigma_p^{-1} V_p^T = \sigma^2 \sum_{i=1}^{p} \frac{\mathbf{v}_i \mathbf{v}_i^T}{\sigma_i^2}.
\]

As \( i \) increases \( \sigma_i \) are smaller, contributions to the covariance are larger, and thus if we truncate the solution we reduce its variance. i.e. the solution is smoother.

**Bias** Unless \( p = n \) the solution may have nonzero components due to projections on \( \mathbf{V}_0 \). Thus bias due to not using this information may be larger than that introduced by measurement error in \( \mathbf{b} \).
**Model Resolution** Suppose the true solution is \( \hat{x} \). Then \( \hat{b} = A\hat{x} \) and we have the generalized inverse solution for the exact problem

\[
\hat{x}^\dagger = A^\dagger A\hat{x} = R_x\hat{x} \quad \text{defining} \quad R_x = A^\dagger A
\]  

(10)

the resolution matrix. Clearly \( R_x \) characterizes the effect of using the generalized inverse solution:

\[
R_x = V_p\Sigma_p^{-1}U_p^TU_p\Sigma_p V_p^T = V_p\Sigma_p^{-1}\Sigma_p V_p^T = V_pV_p^T.
\]

(11)

(i) \( N(A) = 0, \ p = n \) and \( R_x = I_n \). The original model \( \hat{x} \) is recovered exactly in the absence of noise. The resolution is perfect.

(ii) \( N(A) \neq 0, \ p = \text{rank}(A) < n \) and \( R \) is symmetric but not the identity. It is an indication of the smearing of the solution \( \hat{x}^\dagger \neq \hat{x} \). The trace\((R_x)\) is a measure of the resolution. If trace\((R_x)\) \( \approx n \) then \( R_x \) is close to the identity. For the expected value

\[
E(\hat{x}^\dagger) = E(A^\dagger b) = A^\dagger E(b) = A^\dagger A\hat{x} = R_x\hat{x}.
\]

Therefore the bias introduced is

\[
E(\hat{x}^\dagger) - \hat{x} = (R_x - I)\hat{x} = (V_pV_p^T - I_n)\hat{x}
\]

\[
\|E(\hat{x}^\dagger) - \hat{x}\| \leq \|R_x - I\| \|\hat{x}\|
\]

and we can deduce that the diagonal elements of the resolution matrix which are close to 1 will yield elements of \( x \) which are resolved well. In contrast, small elements suggest poorly resolved elements in \( x \), significantly biased components.

**Data Resolution** Given a solution \( x \) it is also a question whether the data is well-resolved,

\[
b \approx Ax = AA^\dagger b = R_0b \quad R_b = AA^\dagger
\]

\[
R_b = U_p\Sigma_p V_p^T V_p\Sigma_p^{-1} U_p^T = U_pU_p^T = I_m - U_0U_0^T.
\]

If \( N(A^T) = 0 \) then \( U_0 = 0 \) and the data fit exactly, otherwise, \( p < m \) and the data are not exactly reproduced.

**Summary** Note that the resolution matrices are independent of the data and hence of noise: they are dependent on the model matrix \( A \). Solution depends on

- **Bias** caused by limited resolution - determined by \( A \) - independent of noise.
- **Noise** mapped from data to model.

**Shaw order 0**

http://math.la.asu.edu/~rosie/classes/Spring2013_files/rankdefhtml/regshaw.html We can also examine the resolution due to rank regularization i.e. for the truncated examples. We look at some examples for an underdetermined matrix.
5.1.2 Example of Rank Reduction for APM 520: from Aster Borchers and Thurber

Rosemary Renaut This is Example 3.1 in the book of straight ray path tomography The domain is of size 3 x 3 measured at 8 points yielding an underdetermined system of equations

```matlab
close all, clear all, clc
pos=get(0,'defaultfigureposition');
A=zeros(8,9);row=[1 0 0 1 0 0 1 0 0];
A(1,:)=row;A(2,:)=[row(9) row(1:8)]; A(3,:)=[row(8:9) row(1:7)];
row=[1 1 1 0 0 0 0 0 0];
A(4,:)=row;A(5,:)=[row(7:9) row(1:6)]; A(6,:)=[row(4:9) row(1:3)];
A(7,1)=2^(1/2);A(7,5)=A(7,1);A(7,9)=A(7,1);A(8,9)=A(7,1);
figure(1), subplot(1,2,1),imagesc(A);colorbar;figprops,
title('The Matrix ');
[U,s,V]=svd(A);
% We see that the actual rank is not 8 but 7
subplot(2,1,2),semilogy(diag(s),','*'); figprops,
title('Singular Values of A');
```

Figure 9: Matrix and singular values

```matlab
Rx=V(:,1:7)*V(:,1:7)';Nx(:,1)=V(:,8);Nx(:,2)=V(:,9);
Rb=U(:,1:7)*U(:,1:7)';Nb=U(:,8);
set(0,'defaultfigureposition',[380 320 540 200])
figure(3),subplot(1,2,1),imagesc(Rx);colorbar,figprops,
title(' Model Resolution R_{x} ')
figure(3),subplot(1,2,2),imagesc(Rb),colorbar,figprops,
title(' Data Resolution R_{b} ')
```
Figure 10: **model resolution** matrix $R_x = A^\dagger A = V_p V_p^T$ and the **data resolution** matrix $R_b = A A^\dagger = U_p U_p^T$, $p = 7$

% Elements of null space for $x$ - any multiple can be added to solution
figure(4), subplot(1,3,1), imagesc(reshape(Nx(:,1),3,3)), colorbar, figprops,
title(' null space 
')
figure(4), subplot(1,3,2), imagesc(reshape(Nx(:,2),3,3)), colorbar, figprops,
title(' null space ')
figure(4), subplot(1,3,3), imagesc(reshape(diag(Rx),3,3)); colorbar, figprops

title('Diagonal of $R_x$')

Figure 11: Null Space Vectors and The trace of $R_x$ determines the resolution of $x$

figure(5), subplot(1,2,1), imagesc(reshape([0 0 0 0 1 0 0 0 0],3,3)); colorbar,
figprops, title('Spike model 
')
subplot(1,2,2), imagesc(reshape(Rx(:,5),3,3)); colorbar, figprops,
title('Blurring of spike 
')

% Reset the figure position
set(0, 'defaultfigureposition', pos)
Resolution and Noise

This is Example 3.3 in the book for problem shaw

\[
\text{n}=20; [A,b,x] = \text{shaw}(n); [U,s,V]=\text{svd}(A);
\]
\[
\% \text{ Matlab believes the rank is 18}
\]
\[
\text{subplot}(1,2,1), \text{semilogy(diag(s)},'\ast\prime); \text{figprops,}
\]
\[
\text{title('Singular Values of } A\prime';)
\]
\[
\text{subplot}(1,2,2), \text{plot(V(:,18));hold on, plot(V(:,1)),figprops,}
\]
\[
\text{legend(['V(:, 1)';'V(:,18)']);p=rank(A)};
\]
\[
\text{title(['Basis vectors and rank ',' int2str(p)]})
\]

Figure 13: Some basis vectors and the singular values

\[
\text{model resolution } R_x = A^\dagger A = V_p V_p^T \text{ and data resolution } R_b = A A^\dagger = U_p U_p^T, p = 18
\]
\[
\text{Resolution } = @(V,p) V(:,1:p)*V(:,1:p)';
\]
\[
\text{Nullspace } = @(V,p) V(:,p+1:end);
\]
\[
\text{Pinvmat } = @(U,s,V) V(:,1:p)*\text{diag}(1./s(1:p))*U(:,1:p)';
\]
\[
\text{Rxspike } = @(Rx,sp) Rx(:,sp);
\]
\[
\text{Rx} = \text{Resolution}(V,p)\n\]
\[
\text{Rx} = \text{Resolution}(U,p)\n\]
\[
\text{nullspace} = \text{Nullspace}(V,p)\n\]
\[
\text{nullspace} = \text{Nullspace}(U,p)\n\]
Figure 14: Resolution for different choices of the truncation. The trace of $R_x$ determines the resolution of $x$.
Figure 15: Elements of null space for x - any multiple can be added to solution The trace of $R_x$ determines the resolution of $x$

sp=10;
spike=zeros(1,n);spike(sp)=1;
figure(5), plot(spike,'r*'),legtxt{1}='spike'
figprops,title(['Spike model centered at ',int2str(sp)]),hold on
plot(Rxspike(Resolution(V,p),sp),'go-');
plot(Rxspike(Resolution(V,10),sp),'m+'); figprops
legend(legtxt,'Location', 'NorthWest')

eta=randn(n,1)*10^-12;
figure(6), plot(spike,'r*'),legtxt{1}='spike'
figprops,title(['Spike model centered at ',int2str(sp), ' with noise']),hold on
plot(Rxspike(Resolution(V,p),sp)+Pinvmat(U,diag(s),V,p)*eta,'go-');
plot(Rxspike(Resolution(V,10),sp)+Pinvmat(U,diag(s),V,10)*eta,'m+'); figprops
legend(legtxt,'Location', 'NorthWest')
Figure 16: The model resolution can be seen by looking at $R_x e_{10}$, showing how a spike is impacted. The model resolution under noise $\eta$ in $b$: $x_{\text{noisy}} = A^T b_{\text{noisy}} = A^T (b + \eta) = R_x b + A^T \eta$. Hence $e_{10}$ goes to $R_x e_{10} + A^T \eta$.

Does increasing the sampling assist in such problems? Look at the following example which is with $n = 100$ as compared to $n = 20$ for problem shaw
http://math.la.asu.edu/~rosie/classes/Spring2013_files/rankdefhtml/rankdefshawbig.html

5.1.3 Resolution under Regularization

First of all we recognize that we can immediately write down the regularized resolution matrices using the definition of the regularized inverse

$$A^\dagger(\lambda) = (A^T A + \lambda^2 I)^{-1} A^T \quad R_x(\lambda) = A^\dagger A \quad R_b(\lambda) = A A^\dagger$$

$$R_x(\lambda) = A(\lambda) = VTV^T, \quad R_b(\lambda) = UTV^T \quad \Gamma = \text{diag}(\gamma_i) = \text{diag}(\sigma_i^2 + \lambda^2).$$ (13)

and the filtering impacts the relevant resolution matrices.

5.1.4 Assignment Feb 18, 2014 Due Feb 20, 2014

1. Investigate the provided example for the resolution under truncation, i.e. regularization by truncation.

2. This one again continues the phillips assignment with regularization parameter estimation from class using the Regularization toolbox.

   (a) For the solutions that you have obtained for the ideal regularization parameters in each case, also now plot the resolution matrices $R_x$ and $R_b$ as another way to contrast the solutions.

   (b) What can you observe as to the ability of the different methods to preserve the resolution of the model and the data?

   (c) Investigate the impact on a pulse at the middle of the domain, $e_{n/2}$. 

43
5.2 More on Regularization: February 19 2014

5.2.1 Further Analysis of the Residual: Normalized Cumulative Periodogram

We again look at the residual

\[ r(\lambda) = Ax_\lambda - b = \sum_{i=1}^{m} (\gamma_i - 1) \frac{u_i^T b}{\sigma_i} u_i \]

\[ = \sum_{i=1}^{m} (\gamma_i - 1) \frac{u_i^T b + u_i^T n}{\sigma_i} u_i \]

The filter needs to be chosen so that the contributions from the reliable coefficients \( |u_i^T \hat{b}| > |u_i^T n| \) dominate in \( x_\lambda \).

Equivalently, we want to pick \( \lambda \) so that \( r(\lambda) \) is dominated by contributions from \( |u_i^T n| \)

i.e. we need to test whether \( r(\lambda) \in \mathbb{R}^m \) is a vector of standard normal entries.

Use time series analysis.

**Diagnostics for \( r(\lambda) \) (Rust and O’Leary (08) and Hansen et al (06))**

**Diagnostic 1** Morozov’s discrepancy principle \( 1/\sigma^2 \| r(\lambda) \|^2 \approx m \)? Precisely is

\[ m - 2\sqrt{2m} \leq \frac{1}{\sigma^2} \| r(\lambda) \|^2 \leq m + 2\sqrt{2m}. \]

Is \( 1/\sigma^2 \| r(\lambda) \|^2 \) within two standard deviations of \( 1/\sigma^2 E(\| r(\lambda) \|^2) \)?

**Diagnostic 2** Is histogram of \( r(\lambda)_i \), consistent with \( r(\lambda) \sim N_m(0, \sigma^2) \)?

**Diagnostic 3** Do entries \( r(\lambda)_i \) match the expectation that they are selected from a time series indexed by \( i \) for time?

We use the **periodogram** or **power spectrum** for \( r(\lambda) \).

Let \( \text{dft} \) be the discrete Fourier transform and \( c \) be the vector of entries

\[ c_j = (\text{dft}(r(\lambda)))_j, \quad j = 1, \ldots, \tilde{m} = \lfloor m/2 \rfloor + 1. \]

\( c = Fr(\lambda) \) for \( F \) the Fourier transform matrix with normalization \( 1/\sqrt{m} \).

\( c_1 \) is the 0 (DC) constant frequency component.

\( c_{\tilde{m}} \) is the highest frequency in the signal

For \( r(\lambda) \sim N_m(0, \sigma_r^2 I_m) \), and row \( f_i^T \) of \( F \)

\[ E(\|c_i\|^2) = E(|f_i^T r(\lambda)|^2) = \text{Var}(f_i^T r(\lambda)) = \sigma_r^2 (f_i^T (f_i^T)^*) \rightarrow E(\|c_i\|^2) \propto \sigma_r^2 \]

where the proportionality depends on the normalization in \( F \).
Normalized Cumulative Periodogram

**Diagnostic 1** Parseval’s relation: $\| \text{dft}(r(\lambda)) \|^2 = \| r(\lambda) \|^2$

**Diagnostic 2 & 3** Normalized Cumulative Periodogram (NCP):

Calculate the ratio of the cumulative sum of entries

$$w_j = \frac{\sum_{j=1}^{\tilde{m}} |c_j|^2}{\sum_{i=1}^{m} |c_i|^2}, \quad j = 1, \ldots, \tilde{m},$$

Because $E(|c_i|^2) \propto \sigma_i^2$, for all $i$, the power spectrum is flat: $w_j \approx 2j/m$.

Define $\tilde{j} = (j - 1)/(m - 1), \quad 0 \leq \tilde{j} \leq .5$.

Then $(\tilde{j}, w_j)$ is a straight line $w_j \approx 2\tilde{j}$ of length $\sqrt{5}/2$.

Ideally $w = \tilde{w}, \quad \tilde{w}_j = 2j$

For 5% significance level the NCP is within Kolmogorov Smirnoff limits $\pm 1.35\tilde{m}^{-1/2}$

How to test automatically: various options - Hansen’s NCP test

Minimize the deviation from straight line $\lambda_{NCP} = \arg \min_{\lambda} \| w(\lambda) - \tilde{w} \|_2$

Figure 17: Example NCP Comparing the Residual Periodogram and the Cumulative Residual: Notice the reverse L for the deviation: hence the corner can be found.
5.3 Weighting for the noise February 24, 2014

If the noise if not white, the approach can be extended.

Suppose that \( n \sim (0, C_b) \), i.e. \( C_b \) is symmetric positive definite covariance of the noise in \( n \).

\( C_b \) is SPD, there exists a factorization \( C_b = D^2 \) and \( D \) is invertible. \((C_b^{1/2} = D)\)

To whiten the noise we multiply by \( D^{-1} \) in the equation \( A\hat{x} = \hat{b} = b + n \)

\[
D^{-1}(A\hat{x} - b) = D^{-1}n = \tilde{n}, \quad \text{where } \tilde{n} \sim (0, D^{-1}C_b(D^{-1})^T) = (0, I_m)
\]

Hence rather than solving for an unweighted fidelity term we solve the weighted problem, \( W = C_b^{-1} \)

\[
x(\lambda) = \arg \min_x \{ \|A\hat{x} - \hat{b}\|^2 + \lambda^2\|x\|^2 \}
\]

using the standard notation \( \|A\|^2_W = A^TWA \).

This may be immediately rewritten as before by defining \( \tilde{A} = W^{1/2}A \) and \( \tilde{b} = W^{1/2}b \)

\[
x(\lambda) = \arg \min_x \{ \|\tilde{A}\hat{x} - \tilde{b}\|^2 + \lambda^2\|x\|^2 \}
\]

Does the weighting change the solution we obtain?

Use SVD for the matrix pair \( \tilde{A} \) instead of \( A \), and apply all operations for the weighted fidelity term.

More generally consider the weighted term

\[
x(\lambda) = \arg \min_x \{ \|\tilde{A}\hat{x} - \tilde{b}\|^2 + \|x\|^2_W \}
\]

5.3.1 Statistical Properties of the Augmented Regularized Residual

Consider weighted regularized residual \( r(\lambda)_A, W_x \) is a SPD weighting on \( x \)

\[
x(W_x) = \arg \min_x \left\| \begin{pmatrix} \tilde{A} \\ W_x^{1/2} \end{pmatrix} x - \begin{pmatrix} \tilde{b} \\ 0_n \end{pmatrix} \right\|^2 := \arg \min_x \|r(\lambda)_A\|^2
\]

For a given solution \( x(W_x) \) we can calculate the cost functional

\[J(W_x) = b^T(AW_x^{-1}A^T + W^{-1})^{-1}b, \quad x(W_x) = W_x^{-1}A^T(A^T W_x^{-1}A + W^{-1})^{-1}b \quad \text{(14)}\]

Using

\[
(A^T B A + C)^{-1} A^T B = C^{-1} A^T (AC^{-1} A^T + B^{-1})^{-1}
\]

with \( B = W \) and \( C = W_x \).

Using the factorization \( W_{x}^{-1} = W_{x}^{-1/2} W_{x}^{1/2} \), and the SVD for \( \tilde{A} \) we can obtain

\[J(W_x) = s^T P^{-1} s, \quad s = U^T W_{1/2} b, \quad P = \Sigma V^T W_{x}^{-1} V \Sigma^T + I_m \quad \text{(16)}\]

**Distribution of the Cost Functional** If \( W \) and \( W_x \) have been chosen appropriately functional \( J \) is a random variable which follows a \( \chi^2 \) distribution with \( m \) degrees of freedom:

\[
J(W_x) \sim \chi^2(m)
\]
This is proved by noting that (14) may be written as
\[ J(W_x) = b^T(A^T W_{x}^{-1} A + W_{x}^{-1})^{-1} b = \|b\|_2^2 Z = \left(A^T W_{x}^{-1} A + W_{x}^{-1}\right)^{-1}. \]

Now by \( b \sim (0, A^T W_{x}^{-1} A + W_{x}^{-1}) \) when \( n \sim (0, W_{x}^{-1}) \) and \( x \sim (0, W_{x}^{-1}) \), yields \( Z^{1/2} b \sim (0, I_m) \). Thus \( J \) is a sum of standard normal variables which in the limit is a \( \chi^2 \) distribution with mean \( m \).

Appropriate weighting makes noise on \( b \) and on model \( x \) white.

Of course noise on \( x \) is unknown, but this determines a parameter choice rule for \( W_x = \lambda^2 I \) using the augmented discrepancy.

**5.3.2 \( \chi^2 \) method to find the parameter (Mead and Renaut)**

**Interval** Find \( W_x = \lambda^2 I \) such that
\[ m - \sqrt{2m z_{\alpha/2}} < b^T(A^T W_{x}^{-1} A + W_{x}^{-1})^{-1} b < m + \sqrt{2m z_{\alpha/2}}. \]
i.e. \( E(J(x(W_x)))) = m \) and \( \text{Var}(J) = 2m \).

**Posterior Covariance on \( x \)** Having found \( W_x \) the posterior inverse covariance matrix is
\[ \tilde{W}_x = A^T W A + W_x \]

**Root finding** Find \( \sigma_x^2 = \lambda^{-2} \) such that
\[ F(\sigma_x) = s^T \text{diag}(\frac{1}{1 + \sigma_x^2 \sigma_i^2}) s - m = 0. \] (17)

**Discrepancy Principle** note the similarity
\[ F(\sigma_x) = s^T \text{diag}(\frac{1}{(1 + \sigma_x^2 \sigma_i^2)^2}) s - m = 0. \]

**Homework:** Due Prove the identities (15), (16) and (17).

**Typical \( F \): Two examples**
Newton’s method yields a unique solution (when one exists)

\( F \) is monotonically decreasing

No solution exists if asymptotically \( F > 0 \) as \( \sigma \to \infty \) which implies no regularization is needed.

Likewise, \( F < 0 \) for all \( \sigma_x \) implies that the degrees of freedom is wrongly given, the noise on \( b \) was not correctly identified.

**NCP for \( r(\lambda)_A \)**

The overall residual \( r(\lambda)_A \) is also white noise like provided \( W \) is chosen appropriately. Thus, we can apply exactly the same NCP idea to this residual.

\[ y(\lambda) = \arg \min_y \{ \| Ay - d \|^2 + \lambda^2 \| y - y_0 \|^2 \} \]

**Solution** (written as solution of the normal equations)

\[ y(\lambda) = (A^T A + \lambda^2 I)^{-1} (A^T d + \lambda^2 y) = (A^T A + \lambda^2 I)^{-1} (A^T (b + Ay_0)) + y_0, \quad b = d - Ay_0 \]

**Shifted problem** with \( b = d - Ay_0, \ Ay_0 = d - b \) and \( y - x = y_0 \).

\[ x_\lambda = \arg \min_x \{ \| Ax - b \|^2 + \lambda^2 \| x \|^2 \} \]

**Solution** \( y(\lambda) = x_\lambda + y_0 \).

Figure 18: In the comparison with the standard residual \( r \) is in black and \( r(\lambda)_A \) in magenta. Optimal \( \lambda \) for the augmented residual is smaller than for the discrepancy, so the solution is smoothed less.

### 5.4 Including a Reference Solution or Prior Information

**Background** or reference solution may be known

\[ y(\lambda) = \arg \min_y \{ \| Ay - d \|^2 + \lambda^2 \| y - y_0 \|^2 \} \]

**Solution** (written as solution of the normal equations)

\[ y(\lambda) = (A^T A + \lambda^2 I)^{-1} (A^T d + \lambda^2 y) = (A^T A + \lambda^2 I)^{-1} (A^T (b + Ay_0)) + y_0, \quad b = d - Ay_0 \]

**Shifted problem** with \( b = d - Ay_0, \ Ay_0 = d - b \) and \( y - x = y_0 \).

\[ x_\lambda = \arg \min_x \{ \| Ax - b \|^2 + \lambda^2 \| x \|^2 \} \]

**Solution** \( y(\lambda) = x_\lambda + y_0 \).
5.5 Relating MAP and Tikhonov Regularization

Statistical interpretation of $y_0$: Let $p(x)$ be the probability for $x$ and $p(y|x)$ be the conditional probability of $y$ given $x$

**Maximum Likelihood Estimator (MLE)** for obtaining parameter $y$ given $d$ is the parameter $\hat{y}$ which maximizes the likelihood function $L(y) = p(d; y)$.
MLE maximizes the log likelihood $l(y) = \log p(d; y)$.

**Example** Suppose $d$ is a realization of vector $y \sim \mathcal{N}(y_0, C)$.
Assume $y_0$ is unknown. The probability density function is

$$p(y; y_0, C) \propto \exp\left(-\frac{1}{2}(y - y_0)^T C^{-1} (y - y_0)\right)$$

and

$$l(y) = -\frac{1}{2}(y - y_0)^T C^{-1} (y - y_0) + c,$$

where $c$ is independent of $y_0$, has maximizer $y = y_0$.

**Bayes Law** If $y$ and $d$ are jointly distributed random vectors.

$$p(y|d) = \frac{p(d|y)p(y)}{p(d)}$$

**MAP** Maximum A Posteriori Estimator is the maximizer of $p(y|d)$ with respect to $y$.

To obtain the MAP we use the linear model. Suppose $y \sim \mathcal{N}(y_0, C_y)$, $n \sim \mathcal{N}(0, C)$ and $d$ is a random vector defined by the linear model $Ay + n = d$. Then $d \sim \mathcal{N}(Ay, C)$ implies

$$p(d|y) \propto \exp\left(-\frac{1}{2}(d - Ay)^T C^{-1} (d - Ay)\right)$$

and the prior is

$$p(y) \propto \exp\left(-\frac{1}{2}(y - y_0)^T C_y^{-1} (y - y_0)\right)$$

Combining terms and using Bayes Law the a posteriori log likelihood function is

$$l(y|d) = -\frac{1}{2}(d - Ay)^T C^{-1} (d - Ay) - \frac{1}{2}(y - y_0)^T C_y^{-1} (y - y_0) + c$$

where $c$ is independent of $y$. Then the MAP estimator which maximizes $l(y|d)$ is equivalent to the minimizer of the Tikhonov regularization

$$\|d - Ay\|^2_{C^{-1}} + \|y - y_0\|^2_{C_y^{-1}}$$

In this case $y_0$ is the expected value of $y$.

5.6 Changing the basis by applying a different operator: March 4, 2014

Imposing the regularization for the norm of $x$ is not necessarily appropriate, dependent on what we anticipate for the solution.

Instead we consider the more general weighting $\|Lx\|^2$.

This leads to the general problem

$$x(\lambda) = \arg\min_x \{\|Ax - b\|^2 + \lambda^2 \|Lx\|^2\}$$
Suppose that $L$ is invertible then we can solve for $y = Lx$ noting for the normal equations
\[
(A^TA + L^TL)x = A^Tb \\
(A^TAL^{-1} + L)^Tx = A^Tb \\
L^T((L^T)^{-1}A^TAL^{-1} + I_n)Lx = A^Tb \\
(\tilde{A}^T\tilde{A} + I_n)y = \tilde{A}^Tb \quad \tilde{A} = AL^{-1}
\]
and we see that this solves for the column scaled matrix $\tilde{A}$, with solution $y = Lx$, and $x$ is found in a different basis.

However, typical $L$: $L$ approximates the first or second order derivative
\[
L_1 = \begin{pmatrix}
-1 & 1 \\
\ddots & \ddots \\
-1 & 1
\end{pmatrix} \\
L_2 = \begin{pmatrix}
1 & -2 & 1 \\
\ddots & \ddots & \ddots \\
1 & -2 & 1
\end{pmatrix}
\]
$L_1 \in \mathbb{R}^{(n-1)\times n}$ and $L_2 \in \mathbb{R}^{(n-2)\times n}$. Note that neither $L_1$ nor $L_2$ are invertible.

### 5.6.1 Boundary Conditions: Zero

Operators $L_1$ and $L_2$ provide approximations to derivatives
\[
D_x(u_i) \approx u_{i+1} - u_i \quad D_{xx}(u_i) \approx u_{i+1} - 2u_i + u_{i-1}
\]
Boundary is at $u_1$ and $u_n$.

Suppose zero outside the domain $u_0 = u_{n+1} = 0$
\[
D_x(u_n) = u_{n+1} - u_n = -u_n \\
D_{xx}(u_1) = u_2 - 2u_1 + u_0 = u_2 - 2u_1 \\
D_{xx}(u_n) = u_{n+1} - 2u_n + u_{n-1} = -2u_n + u_{n-1}
\]

\[
L_1^0 = \begin{pmatrix}
-1 & 1 \\
\ddots & \ddots \\
-1 & 1 \\
-1 & 1
\end{pmatrix} \\
L_2^0 = \begin{pmatrix}
-2 & 1 \\
1 & -2 & 1 \\
\ddots & \ddots & \ddots \\
1 & -2
\end{pmatrix}
\]
$L_1^0, L_2^0 \in \mathbb{R}^{n\times n}$. Both $L_1, L_2$ are invertible.

### 5.6.2 Boundary Conditions: Reflexive

Suppose reflexive outside the domain $u_0 = u_2, \ u_{n+1} = u_{n-1}$
\[
D_x(u_n) = u_{n+1} - u_n = u_{n-1} - u_n \\
D_{xx}(u_1) = u_2 - 2u_1 + u_0 = 2u_2 - 2u_1 \\
D_{xx}(u_n) = u_{n+1} - 2u_n + u_{n-1} = -2u_n + 2u_{n-1}
\]

\[
L_1^R = \begin{pmatrix}
-1 & 1 \\
\ddots & \ddots \\
-1 & 1 \\
1 & -1
\end{pmatrix} \\
L_2^R = \begin{pmatrix}
-2 & 2 \\
1 & -2 & 1 \\
\ddots & \ddots & \ddots \\
2 & -2
\end{pmatrix}
\]
Neither $L_1, L_2$ invertible.

### 5.6.3 Boundary Conditions: Periodic

Suppose periodic outside the domain $u_0 = u_n, u_{n+1} = u_1$

\[
D_x(u_n) = u_{n+1} - u_n = u_1 - u_n \\
D_{xx}(u_1) = u_2 - 2u_1 + u_0 = u_n + 2u_2 - u_1 \\
D_{xx}(u_n) = u_{n+1} - 2u_n + u_{n-1} = u_1 - 2u_n + u_{n-1}
\]

\[
L_1^P = \begin{pmatrix} -1 & 1 & 0 \\ \vdots & \ddots & \vdots \\ 1 & -1 & 1 \end{pmatrix}, \quad L_2^P = \begin{pmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ \vdots & \ddots & \vdots \\ 1 & 1 & -2 \end{pmatrix}
\]

$L_1^P, L_2^P \in \mathbb{R}^{n \times n}$. Neither $L_1, L_2$ invertible.

Both are circulant banded. We can use the FFT to form the matrix product.

### 5.6.4 The Generalized Singular Value Decomposition

Introduce generalization of the SVD to obtain an expansion for

\[
x(\lambda) = \arg \min_x \{ \| \tilde{A}x - \tilde{b} \|^2 + \lambda^2 \| L(x - x_0) \|^2 \}
\]

**Lemma 2 (GSVD).** Assume invertibility and $m \geq n \geq p$. There exist unitary matrices $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{p \times p}$, and a nonsingular matrix $X \in \mathbb{R}^{n \times n}$ such that

\[
A = U \begin{bmatrix} \Upsilon & 0_{(m-n) \times n} \end{bmatrix} X^T, \quad L = V[M, 0_{p \times (n-p)}] X^T,
\]

\[
\Upsilon = \text{diag}(v_1, \ldots, v_p, 1, \ldots, 1) \in \mathbb{R}^{n \times n}, \quad M = \text{diag}(\mu_1, \ldots, \mu_p) \in \mathbb{R}^{p \times p},
\]

with

\[
0 \leq v_1 \leq \cdots \leq v_p \leq 1, \quad 1 \geq \mu_1 \geq \cdots \geq \mu_p > 0, \quad v_i^2 + \mu_i^2 = 1, \quad i = 1, \ldots, p.
\]

Use $\tilde{\Upsilon}$ and $\tilde{M}$ to denote the rectangular matrices containing $\Upsilon$ and $M$.

### 5.6.5 Solution of the Generalized Problem using the GSVD

As for the SVD we need the expression for the regularization matrix

\[
R(\lambda) = (A^T A + \lambda^2 L^T L)^{-1} A^T = (X^T)^{-1} (\tilde{\Upsilon}^T \tilde{\Upsilon} + \lambda^2 \tilde{M}^T \tilde{M})^{-1} \tilde{\Upsilon}^T U^T
\]

Notice

\[
(\tilde{\Upsilon}^T \tilde{\Upsilon} + \lambda^2 \tilde{M}^T \tilde{M})^{-1} \tilde{\Upsilon}^T = \text{diag}(\frac{\nu_i}{\nu_i^2 + \lambda^2 \mu_i^2}, 1, \ldots, 1)
\]

Thus

\[
x_\lambda = \sum_{i=1}^{p} \frac{\nu_i}{\nu_i^2 + \lambda^2 \mu_i^2} (u_i^T b) \bar{x}_i + \sum_{i=p+1}^{n} (u_i^T b) \bar{x}_i
\]

51
where $\tilde{x}_i$ is the $i^{th}$ column of $(X^T)^{-1}$ With $\rho_i = \nu_i / \mu_i$ we have
\[
x_\lambda = \sum_{i=1}^{p} \frac{\rho_i^2}{\nu_i (\rho_i^2 + \lambda^2)} (u_i^T b) \tilde{x}_i + \sum_{i=p+1}^{n} (u_i^T b) \tilde{x}_i
\]
\[
= \sum_{i=1}^{p} \frac{\nu_i}{\nu_i} (u_i^T b) \tilde{x}_i + \sum_{i=p+1}^{n} (u_i^T b) \tilde{x}_i, \quad \gamma_i = \frac{\rho_i^2}{\rho_i^2 + \lambda^2}, i = 1, \ldots, p.
\]

Notice the similarity with the filtered SVD solution
\[
x_\lambda = \sum_{i=1}^{r} \gamma_i \frac{\sigma_i}{\sigma_i} u_i v_i, \quad \gamma_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2}.
\]

We can also immediately express the resolution matrix for the mapped regularization.\(^1\)
\[
A(\lambda) = X^{-T} \Gamma X^T, \quad \gamma_i = \frac{\rho_i^2}{\rho_i^2 + \lambda^2}.
\]

Shaw order 1

Shaw order 2
http://math.la.asu.edu/~rosie/classes/Spring2013_files/rankdefhtml/genregshaworder2.html

VSP order 1 and 2

5.6.6 Finding the optimal regularization parameter using the $\chi^2$ test

For the generalized Tikhonov regularization the degrees of freedom for the cost functional are $m - n + p$, $L$ of size $p \times n$.

Basic Newton iteration to solve $F(\sigma) = 0$,
\[y(\sigma^{(k)})\] is the current solution for which
\[x(\sigma^{(k)}) = y(\sigma^{(k)}) + x_0\]

Use the derivative
\[
J'(\sigma) = -\frac{2}{\sigma^3} \|Lx(\sigma)\|^2 < 0.
\]

and line search parameter $\alpha^{(k)}$ to give
\[
\sigma^{(k+1)} = \sigma^{(k)} (1 + \alpha^{(k)} \frac{1}{2} \left( \frac{\sigma^{(k)}}{\|Lx(\sigma^{(k)})\|} \right)^2 (J(\sigma^{(k)}) - (m - n + p))).
\]

\(^1\)Recall that we use the definition $A = U \Sigma X^T$, $L = V M X^T$ for the GSVD. The matlab function cgsvd assumes $A = U \Sigma X^{-1}$, $L = V M X^{-1}$ and returns $[U, sm, X, V, W]$ where $sm(i,:) = [\nu_i, \mu_i]$ and $W = X^{-1}$. Let our definition replace $X^T = Z^T$, $A(\lambda) = Z^{-T} \Gamma Z^T$. Then returned from cgsvd we obtain $W = X^{-1} = Z^T$ and $X = (Z^T)^{-1} = Z^{-T}$. Hence in the formula in terms of the returned entries from cgsvd we have $R_x = A(\lambda) = X T W$. 52
5.6.7 More on the Generalized Tikhonov Solutions

- The parameter estimation techniques extend for the solutions using the GSVD.
- As with the $\chi^2$ method, other regularization techniques can be extended without the GSVD.
- Observation on the shifted solution: for the result with the $\chi^2$ we require that $E(x) = x_0$. i.e we know the expected average solution. When this is not available we can modify the approach for a non central $\chi^2$ distribution.
- For missing data Hansen notes that solution in the $2$− norm leads to solutions which have $x = 0$ if there are missing data but replacing operator by $L$ the first or second derivative operator fills in a solution which more appropriately fills in a realistic solution.

5.6.8 An Observation on shifting and smoothing

Find the solution $y$ which satisfies

$$y(\lambda) = \arg\min_y \{\|Ay - d\|^2\} \quad \text{st} \quad \|L(y - y_0)\|^2 \text{is minimum} \quad (21)$$

Consider the decomposition $V = [V_1, V_2] = [v_1, \ldots, v_k, v_{k+1}, \ldots, v_n]$, where we assume that the columns of $V_1, V_2$ span the effective range and null space of $A$.

Now use the decomposition to separate the solution into two pieces: $y = y_k + \tilde{y} = \sum_{i=1}^{k} \frac{(u_i^T d)}{\sigma_i} v_i + V_2 c$ such that $y_k$ minimizes the fidelity term.

We want $y$ such that $\|L(y - y_0)\|$ is minimum. Equivalently recalling the definition $A^\dagger = (A^T A)^{-1} A^T$ and noting

$$\|L(y - y_0)\| = \|L(y_k - y_0) + \tilde{y}\| = \|L(y_k - y_0) + LV c\|$$

Then the term is minimum for

$$c = -(LV_2)^\dagger L(y_k - y_0).$$

yielding the solution

$$y = y_k - V_2(LV_2)^\dagger L(y_k - y_0) = (I - V_2(LV_2)^\dagger L)y_k + V_2(LV_2)^\dagger L y_0,$$

Hence the solution of the shifted system obtained from this solution for $y$ is given by

$$x = y - y_0 = (I - V_2(LV_2)^\dagger L)(y_k - y_0).$$
On the other hand suppose that we solve the shifted system directly for $x$ using the same decomposition for $V$. We immediately obtain the solution by applying the result for finding $y$ to that for finding $x$ when $x_0 = 0$.

\[
x = (I - V_2(LV_2)^\dagger L)x_k = (I - V_2(LV_2)^\dagger L)\sum_{i=1}^{k} \frac{(u_i^Tb)}{\sigma_i}v_i
\]

\[
= (I - V_2(LV_2)^\dagger L)\sum_{i=1}^{k} \frac{(u_i^T(d - Ay_0))}{\sigma_i}v_i
\]

\[
= (I - V_2(LV_2)^\dagger L)y_k - (I - V_2(LV_2)^\dagger L)\sum_{i=1}^{k} \frac{(u_i^T(Ay_0))}{\sigma_i}v_i
\]

\[
= (I - V_2(LV_2)^\dagger L)y_k - (I - V_2(LV_2)^\dagger L)(V_1V_1^T)y_0
\]

\[
= y - V_2(LV_2)^\dagger Ly_0 - (I - V_2(LV_2)^\dagger L)(V_1V_1^T)y_0
\]

\[
= y - y_0 + (I - V_2(LV_2)^\dagger L)(L-I)V_1^Ty_0
\]

\[
= y - y_0 + (I - V_2(LV_2)^\dagger L)(V_2V_2^T)y_0 \neq y - y_0.
\]

i.e. the same solution is not obtained. Although, note that $L = I$ has $y = x$.

It is important to be careful when shifting in the regularization.

### 5.6.9 Practical Solution Techniques

Are the bases discussed so far relevant for all problems: we will look first at problems in which alternative options arise due to the specifics of the kernel. Fourier and cosine bases.

For practical problems, i.e. of reasonable size, we do not use SVD or GSVD. Rather we use iterative methods - LSQR

Typically use iterative Krylov method with preconditioning

Goal to utilize forward operations for $A$ and $A^T$

We will also look at the randomized SVD
6 One Dimensional Problems March 24 2014

6.1 Continuous Problem

6.1.1 Zero Function Outside the Domain: Dirichlet Condition

We consider the integral equation

\[ g(s) = \int_0^\pi h(s, t)f(t)dt. \]

When \( f(t) = 0 \) outside the interval \([0, \pi]\) this completely defines the formulation. Suppose that \( s \) and \( t \) are discretized on the interval \([0, \pi]\) by

\[ s_i = \frac{(2i - 1)\pi}{2m} \quad t_j = \frac{(2j - 1)\pi}{2m} \quad \Delta s = \frac{\pi}{m} = \Delta t \quad i, j = 1 : m \]

then the integral equation is approximated by the midpoint quadrature rule

\[ g(s_i) \approx \sum_{j=1}^m h(s_i, t_j)f(t_j)\Delta t \quad 1 \leq i \leq m \]

\[ g = Hf \quad (H)_{ij} = \Delta th(s_i, t_j). \]

If \( f(t) \) is not known to be zero outside the given interval then the formulation is modified because the values for \( g(s) \) are influenced by the values outside the interval \([0, \pi]\).

6.1.2 Reflection Outside the Domain

Suppose that \( f(t) \) is obtained by reflection outside the domain

\[ f_R(t) = \begin{cases} f(-t) & -\pi < t < 0 \\ f(t) & 0 \leq t \leq \pi \\ f(2\pi - t) & \pi < t < 2\pi. \end{cases} \]

Thus

\[ g_R(s) = \int_{-\pi}^{2\pi} h(s, t)f_R(t)dt \]

\[ = \int_{-\pi}^0 h(s, t)f(-t)dt + \int_0^\pi h(s, t)f(t)dt + \int_0^{2\pi} h(s, t)f(2\pi - t)dt \]

\[ = \int_0^\pi h(s, -t)f(t)dt + \int_0^\pi h(s, t)f(t)dt + \int_0^{2\pi} h(s, 2\pi - t)f(t)dt \]

\[ = \int_0^\pi (h(s, -t) + h(s, t) + h(s, 2\pi - t))f(t)dt = \int_0^\pi h_R(s, t)f(t)dt \]

55
and we see that this corresponds to an integral equation with a modified kernel. We also see from (24) that we may write
\[ \mathbf{g} = H_{left}\mathbf{f} + H_Z\mathbf{f} + H_{right}\mathbf{f} \]
where the matrices are defined by the kernels in (24).

**Spatial Invariance in the kernel:** If \( h \) is spatially invariant so that \( h(s, t) = h(s - t) \) then
\[ (H_{left})_{ij} = h(s_i, -t_j) = h(s_i + t_j) \quad (H_{right})_{ij} = h(s_i, (2\pi - t_j)) = h(s_i + t_j - 2\pi) \]
both have entries that are functions of \( i + j \),
\[
\begin{align*}
    s_i + t_j &= \frac{(2i - 1)\pi}{2m} + \frac{(2j - 1)\pi}{2m} = \frac{\pi}{2m}(2i + 2j - 2) = \frac{\pi(i + j - 1)}{m} \\
    s_i + t_j - 2\pi &= \frac{\pi(i + j - 1)}{m} - 2\pi = \frac{\pi}{m}(i + j - 1 - 2n).
\end{align*}
\]
Thus by
\[
\begin{align*}
    s_{i+l} + t_{j-l} &= s_i + t_j \quad (H_{left})_{i+l,j-l} = (H_{left})_{ij} \\
    s_{i+l} + t_{j-l} - 2\pi &= s_i + t_j - 2\pi \quad (H_{right})_{i+l,j-l} = (H_{right})_{ij}
\end{align*}
\]
whereas \( H_Z \) depends on \( s_i - t_j \) for which \( s_{i+l} - t_{j+l} = s_i - s_j \) so that
\[
(H_Z)_{i+l,j+l} = (H_Z)_{ij}
\]
In particular the boundary matrices are constant along the off diagonals (Hankel) and the basic matrix is constant along the diagonals (Toeplitz).

**6.1.3 Periodicity**

Suppose that \( f(t) \) is periodic with period \( \pi \), \( f_P(t) = f(t + \pi) \) then \( g \) is again impacted by the values outside the domain
\[
g_P(s) = \int_{-\pi}^{2\pi} h(s, t) f_P(t) dt
= \int_{-\pi}^{0} h(s, t) f(t + \pi) dt + \int_{0}^{\pi} h(s, t) f(t) dt + \int_{\pi}^{2\pi} h(s, t) f(t - \pi) dt
= \int_{0}^{\pi} h(s, t - \pi) f(t) dt + \int_{0}^{\pi} h(s, t) f(t) dt + \int_{0}^{\pi} h(s, t + \pi) f(t) dt
= \int_{0}^{\pi} (h(s, t - \pi) + h(s, t) + h(s, t + \pi)) f(t) dt = \int_{0}^{\pi} h_P(s, t) f(t) dt,
\]
and we can see that we can generate the matrices for the periodic case.

**Spatial Invariance in the kernel:** If \( h \) is spatially invariant and periodic with period \( \pi \)
\[ h(s, t - \pi) = h(s - t + \pi) = h(s - t) = h(s, t) \quad h(s, t + \pi) = h(s - t - \pi) = h(s - t). \]
6.2 Discrete Convolution

**Vectors**: \( f \in \mathbb{R}^m, h \in \mathbb{R}^{2n-1} \) and \( m \geq 2n - 1 \).

\[
f = (f_1, f_2, \ldots, f_m) \quad h = (h_{-(n-1)}, h_{-(n-2)}, \ldots, h_0, h_1, \ldots, h_{n-1})
\]

**Normalization**

\[
\sum_{i=-(n-1)}^{n-1} h_i = 1.
\]

**Convolution**: Defined with the center of \( h \) at \( h_0 \)

\[
g_i = \sum_j h_{i-j} f_j = \sum_{j=-(n-1)}^{n-1} h_j f_{i-j}
\]

\[
= h_0 f_i + h_1 f_{i-1} + h_{-1} f_{i+1} + h_2 f_{i-2} + h_{-2} f_{i+2} + \cdots + h_{n-1} f_{i-(n-1)} + h_{-(n-1)} f_{i+n-1}
\]

\[
g_i = (f * h)_i = \sum_{j \in \Omega(i)} f_j h_{i-j} = \sum_{i=-(n-1)}^{i+(n-1)} f_j h_{i-j}
\]

\( \Omega(i) \) is the set of integers for the sum and may depend on \( i \), depending on how the extent of \( h \) outside the defined range of the signal is implemented, or as given in the second sum, all values are used.

**Matrix Formulation: the complete Toeplitz system**

\[
\begin{pmatrix}
g_1 \\
\vdots \\
g_m
\end{pmatrix} = 
\begin{pmatrix}
\cdots & \cdots & \cdots & \cdots & h_{-(n-1)} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
h_{n-1} & h_{n-2} & \cdots & \cdots & h_{-(n-1)} \\
h_{n-1} & h_{n-2} & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
h_{n-1} & h_{n-2} & \cdots & \cdots & h_{-(n-1)}
\end{pmatrix}
\begin{pmatrix}
f_{-n+2} \\
\vdots \\
f_0 \\
f_1 \\
\vdots \\
f_m \\
f_{m+1} \\
\vdots \\
f_{m+n-1}
\end{pmatrix}
\]

\( g = Hf, \quad g, f, \text{ length } m, \text{ size } H \text{ of size } m \times (m + 2n - 2). \)

This shows how the entries for \( g \) in a given domain depend on entries in \( f \) outside the domain, above and below the horizontal lines. The matrix \( H \) is a Toeplitz matrix, the entries along the diagonals are constant.
Definition 6. A matrix $A$ is Toeplitz if the entries along the diagonals are constant.

$$A = (a_{ij}) = (h_{i-j}) = \begin{pmatrix} h_0 & h_{-1} & \ldots & h_{-(n-1)} \\ h_1 & h_0 & \ldots & h_{-(n-2)} \\ \vdots & \ddots & \ddots & \vdots \\ h_{n-1} & \ldots & h_1 & h_0 \end{pmatrix}$$

To obtain the matrix $H$ in matlab we use function `toeplitz`:

$T = \text{toeplitz}(c,r)$

returns Toeplitz matrix $T$ whose first column is $c$ and whose first row is $r$.

If the first elements of $c$ and $r$ are different, the column element prevails.

$T = \text{toeplitz}(r)$

For a real vector $r$, returns the symmetric Toeplitz matrix formed from vector $r$, where $r$ defines the first row of the matrix.

In the general case forming the padded point spread function defines the first row and

$$\text{row} = h_{\text{pad}} = [h, 0_{m-1}] \quad \text{col} = [h_{n-1}, 0_{m-1}]^T$$

$$H = \text{toeplitz}(\text{col}, \text{row})$$

Zero Boundary Conditions

This amounts to taking the entries outside the domain to be zero,

$$f_{-n+2} = \cdots = f_0 = 0, \quad f_{m+1} = \cdots = f_{m+n-1} = 0.$$

$$\begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix} = \begin{pmatrix} h_0 & h_{-1} & \ldots & \ldots & h_{-(n-1)} & 0 & \ldots & \ldots & 0 \\ h_1 & h_0 & h_{-1} & \ldots & \ldots & h_{-(n-1)} & \ldots & \ldots & 0 \\ h_2 & h_1 & h_0 & \ldots & \ldots & h_{-(n-1)} & \ldots & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ h_{n-1} & h_{n-2} & \ldots & \ldots & \ldots & \ldots & \ldots & h_{-(n-1)} & 0 \\ 0 & h_{n-1} & h_{n-2} & \ldots & \ldots & h_{-(n-1)} & \ldots & \ldots & h_{-(n-2)} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & h_{n-1} & h_{n-2} & \ldots & \ldots & h_{0} \end{pmatrix} \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix}$$

$$\text{row} = [h_0, h_{-1}, \ldots, h_{-(n-1)}, 0_{\tilde{m}}] \quad \text{col} = [h_0, h_1, \ldots, h_{(n-1)}, 0_{\tilde{m}}]^T$$

$$g = H_Z f \quad H_Z = \text{toeplitz}(\text{col}, \text{row}) \quad \text{where} \quad \tilde{m} = m - n.$$
Periodic Boundary Conditions

The entries outside the domain wrap around,

\[ f_{m+1} = f_1, \ldots, f_{m+n-1} = f_{n-1} \quad f_0 = f_m, \ldots, f_{-(n-2)} = f_{m-n+2} \]

Let \( \tilde{m} = m - (2n-1) \)

\[
\begin{pmatrix}
  g_1 \\
  \vdots \\
  g_m
\end{pmatrix} = 
\begin{pmatrix}
  h_0 & h_{-1} & \cdots & h_{-(n-1)} & 0_{\tilde{m}} & h_{n-1} & \cdots & \cdots & \cdots & h_1 \\
  h_1 & h_0 & h_{-1} & \cdots & h_{-(n-1)} & 0_{\tilde{m}} & h_{n-1} & \cdots & \cdots & h_2 \\
  h_2 & h_1 & h_0 & \cdots & h_{-(n-1)} & \cdots & \cdots & \cdots & \cdots & h_3 \\
  \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
  h_{n-1} & h_{n-2} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0_{\tilde{m}} & h_{-(n-1)} \\
  0 & h_{n-1} & h_{n-2} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & h_{-(n-1)} \\
  \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
  0_{\tilde{m}} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & h_{-(n-1)} \\
  h_{-(n-1)} & 0_{\tilde{m}} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & h_{-(n-2)} \\
  \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
  h_{-1} & \cdots & h_{-(n-1)} & 0_{\tilde{m}} & h_{n-1} & \cdots & \cdots & \cdots & \cdots & h_0
\end{pmatrix} 
\begin{pmatrix}
  f_1 \\
  \vdots \\
  f_m
\end{pmatrix}
\]

row = \([h_0, h_{-1}, \ldots h_{-(n-1)}, 0_{\tilde{m}}, h_{n-1}, \ldots h_1]\)

col = \([h_0, h_{-1}, \ldots h_{(n-1)}, 0_{\tilde{m}}, h_{-(n-1)}, \ldots h_{-1}]^T\)

\(g = H_p f \quad H_P = \text{toeplitz}(\text{col}, \text{row}) = \text{circulant}(\text{col})\).

Here the matrix is circulant, each row is a right shift of the preceding with wrap around and completely determined by the first column of the matrix.

**Definition 7.** Toeplitz \( A \) is circulant if rows / columns are circular shifts.

\[
A = (a_{ij}) = 
\begin{pmatrix}
  h_0 & h_{-1} & \cdots & h_{-(n-1)} \\
  h_{-(n-1)} & h_0 & \cdots & h_{-(n-2)} \\
  \vdots & \vdots & \ddots & \vdots \\
  h_{-1} & \cdots & h_{-(n-1)} & h_0
\end{pmatrix}
\]

With discrete periodicity of length \( n \):

\[
A = (h_{i-j})_n = \text{circulant}(h_0, h_{-1}, \ldots h_{-(n-1)})
\]

Reflexive Boundary Conditions March 25, 2014

The entries outside the domain are obtained by reflection from inside the domain. We assume that \( f_1 \) and \( f_m \) are not on the boundary. Thus

\[ f_{m+1} = f_m, \ldots, f_{m+n-1} = f_{m-n}, \quad f_0 = f_1, \ldots, f_{-(n-2)} = f_{n-1}. \]
This matrix is the sum of the standard Toeplitz matrix that arises for the zero boundary conditions \( H_Z \) and a Hankel matrix formed from the reflexive boundary conditions. A **matrix is Hankel when the entries are constant on the anti diagonals.** Indeed in this case the Hankel matrix is a sum of two Hankel matrices, one due to the left boundary and one due to the right boundary: with \( \tilde{m} = m - n + 1 \)

\[
\begin{bmatrix}
    h_0 + h_1 & h_{-1} + h_2 & h_{-2} + h_3 & \ldots & h_{-(n-2)} + h_{n-1} & h_{-(n-1)} & 0 & \ldots & 0 & 0 \\
    h_1 + h_2 & h_0 + h_3 & h_{-1} + h_4 & \ldots & \ldots & 0 & \ldots & 0 & 0 \\
    h_2 + h_3 & h_1 + h_4 & h_0 + h_5 & \ldots & \ldots & \ldots & \ldots & 0 & 0 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    h_{n-2} + h_{n-1} & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & h_{-(n-1)} & 0 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    0 & h_{n-1} & h_{n-2} & \ldots & \ldots & \ldots & \ldots & h_{-(n-1)} & 0 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    0 & 0 & 0 & \ldots & \vdots & h_{n-1} & h_{n-2} + h_{-(n-1)} & \ldots & h_1 + h_2 & h_0 + h_{-1}
\end{bmatrix}
\]

Hence the matrix for the reflexive case is \( H_R = H + H_{left} + H_{right} \). Notice that because of the extent of \( h \) the left and right matrices do not overlap, one is strictly lower triangular and one is strictly upper triangular, both in the sense of anti diagonal triangularity. Also observe that the left boundary matrix depends on the first column of \( H_Z \) shifted up by one and the right boundary’s last row is the first row of \( H_Z \) in reverse order and shifted to the right by one. When the PSF is symmetric the right boundary is the reflection of the left boundary, about the dominant anti diagonal. In Matlab
\( H = \text{hankel}(c, r) \)
returns a Hankel matrix whose first column is \( c \) and whose last row is \( r \). The last element of \( c \) prevails over the first element of \( r \).

We can write this as

\[
H_{\text{right}} = \text{hankel}(\text{col}, \text{row}) : \text{col} = 0_m \quad \text{row} = [0^{T}_{m-n+1}, h_{-(n-1)}, h_{-(n-2)}, \ldots, h_{-1}] \tag{26}
\]

\[
H_{\text{left}} = \text{hankel}(\text{col}, \text{row}) : \text{col} = [h_1, h_2, \ldots, h_{n-1}, 0^{T}_{m-n+1}]^{T} \quad \text{row} = 0^T_m
\]

\[
H_{Z} = \text{toeplitz}(\text{col}, \text{row}) : \text{col} = [h_0, h_1, \ldots h_{(n-1)}, 0^{T}_{m-n}]^{T} \quad \text{row} = [h_0, h_{-1}, \ldots h_{-(n-1)}, 0^{T}_{m-n}]
\]

or we can use the relation to \( H_{Z} \) and write

\[
H_{\text{right}} + H_{\text{left}} = \text{hankel}([H_{Z}(2 : m, 1); 0], [0, H_{Z}(1, m : -1 : 2)]). \tag{27}
\]

### 6.3 Evaluating the Discrete Convolution

#### 6.3.1 The Discrete Fourier Transform (DFT)

Notice in what follows we use indices \( kj \) for row \( k \) and column \( j \), reserving \( i \) for the square root of \(-1\). We first need to introduce the Discrete Fourier Transform (DFT)

- Suppose \( f \) is a sequence of length \( n \) with components \( f_j \). The unitary DFT of \( f \) is \( \hat{f} \) with component \( k \)

\[
\hat{f}_k = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} f_j e^{-2\pi i (j-1)(k-1)/n}
\]

- The inverse DFT is given by

\[
f_j = \frac{1}{\sqrt{n}} \sum_{k=1}^{n} \hat{f}_k e^{2\pi i (j-1)(k-1)/n}
\]

- Notice that the sequences are periodic with period \( n \): \( f_{j+n} = f_j \).

- It may be useful to note that the Fourier transform can be implemented (thought of) as a matrix operation. Let \( \omega = \exp(-2\pi i / n) \) then

\[
\hat{f} = Ff \quad F_{kj} = \frac{1}{\sqrt{n}} e^{-2\pi i (k-1)(j-1)/n} = \frac{1}{\sqrt{n}} \omega^{(k-1)(j-1)}
\]

\[
f = F^{-1}\hat{f} = F^*f \quad F^*_{kj} = \frac{1}{\sqrt{n}} e^{2\pi i (k-1)(j-1)/n} = \frac{1}{\sqrt{n}} \overline{\omega}^{(k-1)(j-1)}
\]

using the fact that the matrix \( F \) is unitary, \( F^*F = I_n \).
6.3.2 Spectral Decompositions

Relating circulant matrices and the DFT (Chapter 5 Vogel)

**Definition 8 (Circulant Right Shift Matrix).** The matrix \( R \) given by

\[
R = \begin{pmatrix}
0 & 0 & \ldots & 0 & 1 \\
1 & 0 & 0 & \ldots & 0 \\
0 & 1 & 0 & 0 & \ldots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \ldots & 0 & 1 & 0 \\
\end{pmatrix}
\]

or \( R_{kj} = \delta([j - k + 1]_n) \) \( \delta(j) = 1 \) only for \( j = 0 \).

satisfies the right shift property \( R(h_1, h_2, h_3, \ldots, h_n)^T = (h_n, h_1, \ldots, h_{n-1})^T \), and \([k]_n\) indicates \( \text{mod } n \).

By induction it is easy to show that

\[
R^j(h_1, h_2, h_3, \ldots, h_n)^T = (h_{n-j+1}, h_{n-j+2}, \ldots, h_{n-j})^T
\]

and

\[
R^j = \begin{pmatrix}
0 & 0 & \ldots & 1 & \ldots & 0 & 0 \\
0 & 0 & 0 & \ldots & 1 & \ldots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \ldots & \ldots & \ldots & \ldots & \ldots & 1 \\
1 & 0 & \ldots & \ldots & \ldots & \ldots & 0 \\
0 & 1 & \ldots & \ldots & \ldots & \ldots & \ldots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ldots \\
\end{pmatrix}
\]

Position of 1 is \( n - j + 1 \)

Row \( j \) .

**Theorem 6.1 (Expressing the circulant matrix in terms of shifts).** Let

\[
A = \begin{pmatrix}
h_1 & h_n & \ldots & h_3 & h_2 \\
h_2 & h_1 & \ldots & h_4 & h_3 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
h_{n-1} & \ldots & h_2 & h_1 & h_n \\
h_n & h_{n-1} & \ldots & h_2 & h_1 \\
\end{pmatrix}
\]

Then, noting \( R^0 = I \),

(i) \( A = \sum_{j=0}^{n-1} h_{j+1} R^j \) (periodicity \( h_{n+p} = h_p \))

(ii) \( A = F^* A F \), \( \Lambda = \sqrt{n} \text{diag}(\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_n) = \sqrt{n} \text{diag}(\hat{h}) \)

(iii) \( (\sqrt{n}\hat{h}_1, F_i^*) \) are the eigenvalue - eigenvector pairs of \( A \).

(iv) \( \lambda_j = (\sqrt{n}FA_1)_j \). Thus \( (FA_1)_j = \hat{h}_j \).
(v) \( A \) is normal.

Proof.

(i) This is easy to see from the structure of the Toeplitz matrix with each diagonal dependent on one entry for \( h \).

(ii) We use \( \omega = \exp(-2\pi i/n) \), then \( \sqrt{n} F_{kj} = \omega^{(k-1)(j-1)} \), \( \sqrt{n} F_{kj}^* = \bar{\omega}^{(k-1)(j-1)} \) and

\[
(F^*F)_{lk} = \frac{1}{n} \sum_{j=1}^{n} \omega^{(j-1)(k-1)} \bar{\omega}^{(l-1)(j-1)}
\]

\[= \frac{1}{n} \sum_{j=1}^{n} \omega^{(j-1)(k-l)} = \delta([k - l]_n), \quad F^* F = FF^* = I_n,
\]

which follows from a well-known equality for the sum of exponentials. Let \( F_l^* \) be the \( l^{th} \) column of \( F^* \). We know

\[
F_l^* = \frac{1}{\sqrt{n}} \begin{bmatrix} 1, \bar{\omega}^{(l-1)}, \bar{\omega}^2(l-1), \ldots, \bar{\omega}^{(n-1)(l-1)} \end{bmatrix}^T
\]

\[
RF_l^* = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^{(n-1)(l-1)}, 1, \bar{\omega}^{(l-1)}, \ldots, \bar{\omega}^{(n-1)(l-1)} \end{bmatrix}^T,
\]

\[
(RF_l^*)^* = \frac{1}{\sqrt{n}} \omega^{l-1} \left( \frac{1}{\omega^{l-1}} \right) \begin{bmatrix} \omega^{(n-1)(l-1)}, 1, \bar{\omega}^{(l-1)}, \ldots, \bar{\omega}^{(n-1)(l-1)} \end{bmatrix}^T
\]

\[= \frac{1}{\sqrt{n}} \omega^{l-1} \begin{bmatrix} \omega^{(l-1)}, \omega^{(n-1)(l-1)}, 1, \bar{\omega}^{(l-1)}, \ldots, \bar{\omega}^{(n-1)(l-1)} \end{bmatrix}^T = \omega^{l-1} F_l^*.
\]

This holds for \( l = 1 : n \), hence that post multiplication by a diagonal matrix scales the columns of the matrix gives

\[
RF^* = F^* \text{diag}(1, \omega, \omega^2, \ldots, \omega^{n-1})
\]

\[
R = F^* \text{diag}(1, \omega, \omega^2, \ldots, \omega^{n-1}) F = F^* \text{diag}(\omega^{l-1}) F.
\]

Immediately we obtain

\[
R_l = F^* \text{diag}(\omega^{l-1})^j F = F^* \text{diag}(\omega^{(l-1)j}) F,
\]

where \( \text{diag}(\omega^{(l-1)j}) \) is the diagonal matrix with entry \( \omega^{(l-1)j} \) in column \( l \). Thus using
(i)
\[
A = \sum_{j=0}^{n-1} h_{j+1} R^j = \sum_{j=0}^{n-1} h_{j+1} F^* \text{diag}(\omega^{(l-1)j}) F
\]
\[
= F^* \text{diag} \left( \sum_{j=0}^{n-1} h_{j+1} \omega^{(l-1)j} \right) F = F^* \text{diag} \left( \sum_{j=1}^{n} h_j \omega^{(l-1)(j-1)} \right) F
\]
\[
= \sqrt{n} F^* \text{diag}(\hat{h}) F = \sqrt{n} F^* \text{diag}(\hat{h}) F
\]
\[
= F^* \Lambda F, \quad \Lambda = \sqrt{n} \text{diag}(\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_n).
\]

(iii) This follows immediately now from (ii) by \( AF^*_l = \lambda_l F^*_l = \sqrt{n} \hat{h}_l F^*_l \).

(iv) Let \((A)_1\) be the first column of \(A\) then observe
\[
FA = \Lambda F \quad F(A)_1 = \Lambda F_1
\]
but \( F_1 = (1/\sqrt{n})(1, 1, \ldots, 1)^T \) hence
\[
(FA_1)_j = \lambda_j/\sqrt{n} = \hat{h}_j.
\]

(v) To show \(A\) is normal note \( AA^* = F^* \Lambda F F \Lambda F = F^*(\Lambda)^2 F = A^*A \).

\[\square\]

Calculating \(Ax\) and \(A^{-1}b\) for \(A\) circulant March 26 2014

Given the decomposition \(A = F^* \Lambda F\), where \(\Lambda\) is determined by \(h\) which defines \(A\), calculate \(Ax\) by observing that \(Fx = \text{dft}(x) = \hat{x}\) and \(F^*(\Lambda Fx) = \text{idft}(\Lambda Fx)\). Thus
\[
Ax = F^* \Lambda Fx = F^* \Lambda \hat{x}
\]
\[
= \sqrt{n} F^*(\hat{h} \ast \hat{x}) = \sqrt{n} F^*(\hat{y}) \text{ where } \hat{y} = \hat{h} \ast \hat{x}
\]
\[
= \sqrt{n} \hat{y}
\]

Hence to form \(\hat{x}\) convolve with \(\hat{h}\) to give \(\hat{y}\), then invert to give \(\sqrt{n} \hat{y}\). Assuming \(\Lambda\) is non-singular, \(A^{-1} = F^* \Lambda^{-1} F\) and
\[
A^{-1}b = F^* \Lambda^{-1} Fb = F^* \Lambda^{-1} \hat{b}
\]
\[
= \frac{1}{\sqrt{n}} F^*(\hat{b} \ast \hat{h}) = \frac{1}{\sqrt{n}} F^*(\hat{y}) \text{ where } \hat{y} = \hat{b} \ast \hat{h}.
\]
\[
= \frac{1}{\sqrt{n}} \hat{y}
\]

We thus apply this for the case of **periodic boundary conditions**: \(H_P = \text{circulant}(\text{col})\),
\[
\text{col} = [h_0, h_1, \ldots, h_{(n-1)}, 0, \ldots, h_{(n-1)}, \ldots, h_{-1}]^T = h_{\text{ext}}
\]

64
where $h_{\text{ext}}$ is $h$ extended to length $m$. Thus the Fourier Transform matrices are of size $m$, and we form $\hat{h}_{\text{ext}}$ to obtain the spectral decomposition: Important: $h_{\text{ext}}$ is the first column of $A$.

**Summary for circulant $A$**

Transform PSF $\hat{h}_{\text{ext}} = F_m h_{\text{ext}} = \text{dft}(h_{\text{ext}})$

Forward operation $A x : \hat{x} = \text{dft}(x), \hat{y} = \hat{x} \ast \hat{h}_{\text{ext}}$. $Ax = \sqrt{m} \text{idft} (\hat{y})$

Inverse Operation $A^{-1} b : \hat{b} = \text{dft}(b), \hat{y} = \hat{x} / \hat{h}_{\text{ext}}$. $A^{-1} b = \frac{1}{\sqrt{m}} \text{idft}(\hat{y})$

6.3.3 General Case

Suppose that there exists a decomposition $A = G^* \Lambda G$, for a transform matrix $G$, and where $\Lambda = \text{diag}(f(A))$ for some function of the entries in $A$. Denote by $\hat{x} = G x$ to indicate the transform of data $x$. Then we immediately obtain as for the Fourier transform case:

$$A x = G^* \Lambda G x = G^* \Lambda \hat{x}$$

$$= G^* (f(A) \ast \hat{x}) = G^* (\hat{y}) \text{ where } \hat{y} = f(A) \ast \hat{x}$$

$$= y$$

where $G^*$ is the inverse transform for transform matrix $G$, and using $G^* G = I$.

Again, also assuming that $f(A) \neq 0$ for any component, then with $A^{-1} G^* \Lambda^{-1} G$ we have

$$A^{-1} b = G^* \Lambda^{-1} G b = G^* \Lambda^{-1} \hat{b}$$

$$= G^* (\hat{b} / f(\hat{A}).) = G^* (\hat{y}) \text{ where } \hat{y} = \hat{b} / f(\hat{A})$$

$$= y$$

We now show that there is an equivalent useful decomposition for the case of reflexive boundary conditions.

6.3.4 The Discrete Cosine Transform

We suppose that $h(t)$ is defined for $0 \leq t \leq \pi$ with periodicity $h(t + \pi) = h(t)$, so that $h(t + p\pi) = h(t)$. Let $t$ be discretized on $[0, \pi]$ by $t_l = (2l - 1)\pi/2n$, $l = 1 : n$, $dt = \pi/n$ and introduce the basis vectors $w_{j+1}$, $j = 0 : n - 1$ with components

$$(w_{j+1})_l = c_j \cos(jt_l) \quad c_j = \begin{cases} \sqrt{1/n} & j = 0 \\ \sqrt{2/n} & \text{otherwise} \end{cases}$$

which discretize the basis functions $\cos(jt)$, and the choice for $c_j$ assure $\|w_{j+1}\|_2^2 = 1$. Moreover the vectors are orthogonal $w_{j+1}^T w_{i+1} = 0$, $i \neq j$, forming the columns of an orthogonal matrix $W$. The **discrete cosine transform** of a vector $x$ is defined by

$$\hat{x} = \text{dct}(x) = W^T x$$
with components

\[ \hat{x}_{j+1} = w_j^T x = c_j \sum_{l=1}^{n} \cos(j t_l) x_l. \]

Relating the Symmetric Toeplitz - Hankel and the DCT (Appendix B Hansen (2010)) First note that for the case of a PSF which is symmetric, \( h_{-l} = h_l, \ l = 1 : n \), \( H_Z \) is symmetric and defined by \((h_0, h_1, \ldots, h_{n-1}, 0_{m-n})\). Thus for the case of the reflexive boundary conditions \( H_R \) in (26) is symmetric Toeplitz - Hankel (STH) and defined by 
\[
\begin{pmatrix}
h_0 + h_1 & h_1 + h_2 & h_2 + h_3 & \ldots & h_{(n-2)} + h_{n-1} & h_{(n-1)} & 0 & \ldots & 0 & 0 \\
h_1 + h_2 & h_0 + h_3 & h_1 + h_4 & \ldots & \ldots & 0 & 0 & \ldots & 0 & 0 \\
h_2 + h_3 & h_1 + h_4 & h_0 + h_5 & \ldots & \ldots & \ldots & \ldots & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
h_{n-2} + h_{n-1} & \ldots & \ldots & \ldots & h_{(n-1)} & 0 & 0 & \ldots & 0 \\
h_{n-1} & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\
0 & h_{n-1} & h_{n-2} & \ldots & \ldots & \ldots & \ldots & \ldots & h_{(n-1)} & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & \ldots & h_{n-1} & h_{n-2} + h_{(n-1)} & \ldots & \ldots & h_1 + h_2 & h_0 + h_1
\end{pmatrix}
\]

Theorem 6.2 (Expressing the STH matrix in terms of shifts). Suppose \( A \) is a STH matrix defined by coefficients \( \alpha_k \), the entries of the first column of \( A \).

(i) \( A = \sum_{k=0}^{n-1} \alpha_k S_k \) for basis matrices \( S_k, S_0 = I \), and otherwise are zero except

\[ (S_k)_{ij} = \begin{cases} 
1 & |i - j| = k \\
1 & i + j = k + 1 \\
1 & i + j = 2n - k + 1
\end{cases} \]
i.e. the \((i, j)\) pairs which define the non zero entries of \(S_k\), \(k \geq 1\), are

\[
\{(k + j, j), (j - k, j), (k + 1 - j, j), (2n - k + 1 - j, j)\}
\]

\[
\{(i, i - k), (i, i + k), (i, k + 1 - i), (i, 2n - k + 1 - i)\}
\]

where here it is assumed that any index less than 1 is invalid. In all cases only two pairs are valid either,

\[
\{(i, i - k), (i, 2n - k + 1 - i)\} \quad \& \quad \{(i, i + k), (i, k + 1 - i)\}.
\]

(ii) The discrete cosine vectors \(w_{j+1}\) are eigenvectors for each of the basis matrices \(S_k\), with eigenvalue \(2 \cos(jk\pi/n)\). Hence they are also eigenvectors of \(A\).

(iii) The decomposition \(A = WAW^T\) follows where the diagonal matrix is \((\Lambda)_j = \lambda_j\).

(iv) \[
A_1 = Ae_1 = WAW^T e_1 \leftrightarrow W^T A_1 = \Lambda W^T e_1.
\]

Consequently, the eigenvalues are \(\lambda_j = \text{dct}(A_1)^j/\text{dct}(e_1)^j\)

(v) \(A\) is normal.

Proof.

(i) For this we note that the matrix is fully defined by the entries in the first column. Thus for a given entry it suffices to find the non zero entries of the particular matrix \(S_k\). It is not difficult to find the mutually exclusive pairs in each case showing that there are two entries in each column, except for the case with \(k = 0\).

(ii) We form the product \(S_k w_{j+1}\). When \(k = 0\) the result follows immediately because \(S_0 = I\). Otherwise, ignoring the scale factor \(c_j\) which will cancel,

\[
(S_k w_{j+1})_l = \sum_{p=1}^{n} (S_k)_{lp} (w_{j+1})_p = (w_{j+1})_{l-k} + (w_{j+1})_{l+k} + (w_{j+1})_{k+1-l} + (w_{j+1})_{2n-k+1-l}
\]

\[
= \cos(jt_{l-k}) + \cos(jt_{l+k}) + \cos(jt_{k+1-l}) + \cos(jt_{2n-k+1-l})
\]

\[
= \cos(jt_{l+k}) + \cos(jt_{k+1-l})
\]

where the last step occurs by noting only two pairs are valid in each case and by the periodicity of \(\cos\) their impact is the same. Now

\[
\cos(jt_{l+k}) + \cos(jt_{k+1-l}) = \cos((2l + k) - 1)j\pi/2n) + \cos((2k - l + 1) - 1)j\pi/2n)
\]

\[
= \cos((2k + (2l - 1))j\pi/2n) + \cos((2k - (2l - 1))j\pi/2n)
\]

\[
= 2 \cos(jk\pi/n) \cos((2l - 1)j\pi/2n) = 2 \cos(jk\pi/n)(w_{j+1})_l.
\]

Thus \(w_{j+1}\) is an eigenvector of \(S_k\), with eigenvalue \(2 \cos(jk\pi/n)\) and hence of \(A\) for all pairs \((k, j)\)
(iii) The decomposition follows immediately by noting $A_{w, j+1} = \lambda_{j+1} w_{j+1}$ yields $w_{i+1}^T A w_{j+1} = \lambda_{j+1} w_{i+1}^T w_{j+1} = \delta_{ij}$ and note $(W^T A W)_{ij} = w_{i+1}^T A w_{j+1}$. Hence $W^T A W = \Lambda$ and the decomposition follows.

(iv) It is immediate that $(A e_1) = A e_1$. But by part (iii) $A W = W \Lambda$ for matrix $W$ formed from the eigenvectors and where $\Lambda$ is the diagonal matrix of eigenvalues. Thus $A e_1 = W \Lambda W^T e_1$ and it follows that $W^T A e_1 = \Lambda W^T e_1$. But $W^T e_1 = \text{dct}(e_1)$ and $W^T A e_1 = \text{dct}(A e_1)$. Thus $\text{dct}(A e_1) = A \text{dct}(e_1)$ and $\lambda_j = \text{dct}(A e_1) / \text{dct}(e_1)$ and the spectrum is determined by the first column of the STH matrix $A$.

(v) $A$ is normal follows as before.

Calculating $A x$ and $A^{-1} b$ for $A$ STH

Given the decomposition $A = W \Lambda W^T$, where $\Lambda$ is determined by $h$ which defines $A$, calculate $A x$ by observing that $W^T x = \text{dct}(x) = \hat{x}$ and $W(\Lambda W^T x) = \text{idct}(\Lambda W^T x)$. Thus

$$A x = W \Lambda W^T x = W \Lambda \hat{x}, \quad \hat{h} = \text{dct}(A e_1) / \text{dct}(e_1)$$

$$= W(\hat{h} * \hat{x}) = W(\hat{y}) \text{ where } \hat{y} = \hat{h} * \hat{x} = y$$

Hence to form $\hat{h}$ convolve with $\hat{x}$ to give $\hat{y}$, then invert to give $y$. Assuming $\Lambda$ is non singular, $A^{-1} = W \Lambda^{-1} W^T$ and

$$A^{-1} b = W \Lambda^{-1} W^T b = W \Lambda^{-1} \hat{b}$$

$$= W(b / \hat{h}) = W(\hat{y}) \text{ where } \hat{y} = b / \hat{h}$$

For the reflexive case therefore we have $A = H_Z + H_{left} + H_{right}$ where for symmetric PSF $h$ we have

$$H_{right} = \text{hankel}(\text{col}, \text{row}) : \text{col} = 0_m \text{ row} = [0^T_{m-n+1}, h_{n-1}, h_{n-2}, \ldots, h_1]$$

$$H_{left} = \text{hankel}(\text{col}, \text{row}) : \text{col} = [h_1, h_2, \ldots, h_{n-1}, 0^T_{m-n+1}]^T \text{ row} = 0_m^T$$

$$H_Z = \text{toeplitz}(\text{col}, \text{row}) : \text{col} = [h_0, h_1, \ldots h_{n-1}, 0^T_{m-n}]^T \text{ row} = [h_0, h_1, \ldots h_{n-1}, 0^T_{m-n}]$$

$$H_Z = \text{toeplitz}(\text{row})$$

We thus apply this for the case of reflexive boundary conditions: $H_R$, formed from $H_Z$ and the left and right boundary condition matrices. Thus the Discrete Cosine Transform matrices are of size $m$, and the spectrum again depends on the first column of $H_R$.

Summary for STH $A = H_R$
Transform first column which depends on the PSF
\[ \hat{h} = \text{dct}(A_1) / \text{dct}(e_1). \] (35)

Forward operation \( Ax : \hat{x} = \text{dct}(x), \hat{y} = \hat{x} \ast \hat{h}. \) \( Ax = \text{idct}(\hat{y}) \)

Inverse Operation \( A^{-1} b : \hat{b} = \text{dct}(b), \hat{y} = \hat{x} / \hat{h}. \) \( A^{-1} b = \text{idct}(\hat{y}) \)

Relating the Toeplitz matrix and the DFT March 31 2014 (Vogel Chapter 5)

To calculate the product \( Ax \) when \( A \) is Toeplitz, embed in a larger circulant matrix. Again suppose that we have a Toeplitz matrix which is determined by the padded PSF
\[ h_{\text{ext}} = [0, h_{-(n-1)}, h_{-(n-2)}, \ldots, h_0, \ldots, h_{n-1}, 0], \quad \tilde{m} = m - n. \]

Note that this is a symmetric and periodic extension of the PSF, under the assumption that \( h_{|l|} \) tends to 0 with increasing \( |l| \). Let \( \text{col} = [h_0, \ldots, h_{n-1}, 0, \tilde{m}], \text{row} = [h_0, h_{-1}, \ldots, h_{-(n-1)}, 0] \) and \( A = \text{toeplitz}(\text{col}, \text{row}) \). Let matrix \( C \) be of the form
\[ C = \begin{bmatrix} A & B \\ B & A \end{bmatrix} \]

and \( B \) chosen to make \( C \) circulant. Then we observe that
\[ C \begin{bmatrix} x \\ 0 \end{bmatrix} = \begin{bmatrix} Ax \\ Bx \end{bmatrix} \]

and so the forward operation \( Ax \) can be calculated using the circulant matrix \( C \), hence by using the FFT algorithm described in (28)-(30). It is not too difficult to see how to build the matrix \( B \) from the matrix \( A \) under the condition that \( C \) is circulant, each row has to be right shifted from the prior row. We obtain matrix \( B \) as the Toeplitz matrix defined by
\[ (h_1, h_2, \ldots, h_{n-1}, 0, h_{-(n-1)}, h_{-(n-2)}, \ldots, h_{-1}) \]

which for the extended PSF becomes
\[ (h_1, h_2, \ldots, h_{n-1}, 0, m_{-(n-2)}, h_{-(n-1)}, h_{-(n-2)}, \ldots, h_{-1}) \]

\( B = \text{toeplitz}(\text{col}, \text{row}) \) \quad \text{row} = [0, 0, h_{-(n-1)}, h_{-(n-2)}, \ldots, h_1] \quad \text{col} = [0, 0, h_{-(n-1)}, \ldots, h_{-1}] \)

The first column of \( C \) which defines the fft is given by
\[ \text{col} = [h_0, \ldots, h_{n-1}, 0, 0, h_{-(n-1)}, h_{-(n-2)}, \ldots, h_{-1}] \]

which for symmetric \( h \) (standard) is
\[ \text{col} = [h_0, \ldots, h_{n-1}, 0, 0, h_{-(n-1)}, h_{-(n-2)}, \ldots, h_{-1}] \]

and the forward matrix product is obtained by the first \( m \) entries in \( Cx \).

Note that we cannot use the inverse \( C^{-1} b \) to obtain \( T^{-1} b \), hence this analysis is useful for defining forward operations which are used in iterative methods. Observe though that the information on the spectrum of the forward operator obtained in this way is still useful.

There is another way to relate the matrix \( C \) and \( T \) which gives explicit information about the spectrum, see Hansen 2010.
6.4 Solutions are obtained with respect to a basis

Note that this discussion is relevant for completing the homework finding the regularization parameter for DCT and DFT expansions for the matrix $A$.

Fourier decomposition for Circulant $A$: Periodic Boundary Conditions

Suppose $A \in \mathbb{R}^{N \times N}$:

- $A = F^* \Lambda F$, $F$ unitary $F^*F = I_N$,
- $\Lambda = \text{diag}(\hat{h}_{\text{ext}})$ for $\hat{h}_{\text{ext}}$ defined by (31).
- The solution for $Ax = b$ can be expressed in terms of the basis vectors $f_j^*$ which are the columns of $F^*$.

$$x = \sqrt{\frac{1}{N}} \sum_{j=1}^{N} \frac{\hat{b}_j}{\hat{h}_j} f_j^* = \sqrt{\frac{1}{N}} \sum_{j=1}^{N} \frac{\gamma_j}{\lambda_j} f_j^* \quad \gamma_j = \hat{b}_j \quad \lambda_j = \hat{h}_j. \quad (36)$$

Cosine decomposition for STH $A$: Reflexive Boundary Conditions

Suppose $A \in \mathbb{R}^{N \times N}$:

- $A = W \Lambda W^T$, $W$ orthogonal $W^T W = I_N$,
- $\Lambda = \text{diag}(\hat{h})$ for $\hat{h}$ defined in (35) which is obtained from the discrete cosine transform.
- The solution for $Ax = b$ can be expressed in terms of the basis vectors $w_j$ which are the columns of $W$ and the discrete cosine transform of $b$, $\hat{b} = \text{dot}(b)$

$$x = \sum_{j=1}^{N} \frac{\hat{b}_j}{\hat{h}_j} w_j = \sum_{j=1}^{N} \frac{\gamma_j}{\lambda_j} w_j \quad \gamma_j = \hat{b}_j \quad \lambda_j = \hat{h}_j. \quad (37)$$

6.4.1 Summary: Spectral Decomposition of the Solution

- For all cases we have a decomposition of the solution as a sum of basis vectors, $g_j$, where $g_j = w_j$ for the DCT case and $g_j = f_j^*$ for the FFT.
- For the Fourier/Cosine cases the order of the spectral components $\lambda_j$ is not defined as it is for the SVD.
- We can examine the solution by examining the spectral components in each case. $\lambda_j$ are equivalent to the weights $\sigma_j$ of the SVD.
- We note that the DFT and DCT provide feasible approaches for finding the spectrum and evaluating the forward and inverse operations. $Ay$ and $A^{-1}y$ for any vector $y$. 70
• Spectral decomposition suggests truncation to remove the components \( \lambda_j \) which lead to contamination of the solution.

**Remark 2** (Relating the DCT expansion and the SVD Expansion). Let \( |\Lambda| = \text{diag}(|\lambda_j|) \) and \( \Omega = \text{diag}(\text{sign}(\lambda_j)) \). Then we may observe

\[
A = W\Lambda W^T = W\Omega|\Lambda|W^T = U^T|\Lambda|W^T
\]

gives an SVD-like expansion for \( A \), in which it is clear that \( U \) is orthogonal, but the spectral entries in \( |\Lambda| \) are not necessarily ordered from large to small as in the SVD.

**Remark 3** (Relating the FFT expansion and the SVD Expansion). In this case we do not have the same result as in Remark 2 because the FFT expansion is not real and we cannot simply use the magnitude of the coefficients in \( \Omega \). However we will still have the same problem with entries in \( \Lambda \) which have small absolute magnitude contributing to sensitivity in evaluating \( x \).

### 7 Extension to 2D: In Matrix Notation March 31 2014

#### 7.1 Image Convolution 2D PSF

Let \( x = \text{vec}(X) \), be the matrix \( X \) with entries stacked as a vector. i.e. we take the entries of \( X \in \mathbb{R}^{m \times n} \) column wise and set \( N = mn \),

\[
\text{vec}(X) = \begin{pmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_n
\end{pmatrix} \in \mathbb{R}^N
\]

Then the convolution operator

\[
G = H \ast X
\]

is rewritten as

\[
Ax = b
\]

where \( A \) represents the PSF \( H \), and entries of \( A \) depend on how this is implemented.

Condition of the problem can still be investigated naively for the linear system. This requires the generation of the matrix \( A \) for a two dimensional problem.

#### 7.1.1 Descriptive Version of 2D Convolution

1. Regard the PSF as a mask with elements in an array.

2. Rotate the PSF array by 180°.
3. To compute convolution for a certain pixel we have to line up the central point of the rotated mask on that pixel.

4. Multiply the corresponding elements within the mask.

5. Sum the elements within the mask.

6. Take account of boundary conditions as for the 1D Case.

### 7.1.2 Simple Example

\[ X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}, \quad H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}, \quad G = \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix} \]

Rotated \( H \), align with \( x_{22} \) and multiply

\[ H^{\text{flip}} = \begin{pmatrix} h_{33} & h_{32} & h_{31} \\ h_{23} & h_{22} & h_{21} \\ h_{13} & h_{12} & h_{11} \end{pmatrix} \quad X \ast H^{\text{flip}} = \begin{pmatrix} x_{11}h_{33} & x_{12}h_{32} & x_{13}h_{31} \\ x_{21}h_{23} & x_{22}h_{22} & x_{23}h_{21} \\ x_{31}h_{13} & x_{32}h_{12} & x_{33}h_{11} \end{pmatrix} \]

where \( X \ast H \) denotes element wise multiplication. We then sum all the entries. In the Vec notation we have

\[ g_{22} = \text{vec}(H^{\text{flip}})^T \text{vec}(X) \]

as an inner product. All entries are obtained as inner products.

### 7.1.3 Calculating all entries for the case of zero boundary conditions

Order the entries by columns.

\[
\begin{pmatrix}
  g_{11} \\
  g_{21} \\
  g_{31} \\
  g_{12} \\
  g_{22} \\
  g_{32} \\
  g_{13} \\
  g_{23} \\
  g_{33}
\end{pmatrix} = \begin{pmatrix}
  h_{22} & h_{12} & h_{21} & h_{11} \\
  h_{32} & h_{22} & h_{31} & h_{21} \\
  h_{12} & h_{11} & h_{11} & h_{21} \\
  h_{23} & h_{13} & h_{22} & h_{12} \\
  h_{33} & h_{23} & h_{32} & h_{22} \\
  h_{13} & h_{12} & h_{12} & h_{22} \\
  h_{23} & h_{13} & h_{22} & h_{12} \\
  h_{33} & h_{23} & h_{32} & h_{22} \\
  h_{13} & h_{12} & h_{12} & h_{22}
\end{pmatrix} \begin{pmatrix}
  x_{11} \\
  x_{21} \\
  x_{31} \\
  x_{12} \\
  x_{22} \\
  x_{32} \\
  x_{13} \\
  x_{23} \\
  x_{33}
\end{pmatrix}
\]

Each block is of Toeplitz form and that as a block matrix the matrix is also Toeplitz.

Matrix is **block Toeplitz with Toeplitz Blocks** BTTB.
7.2 Constructing matrix $A$: Boundary Conditions, April 1, 2014

PSF is usually of finite extent, but for a finite section of an image, convolution with the PSF uses values outside the specific image. If this is not handled correctly, boundary effects will contaminate the image, leading to artifacts. This is true whether for the forward or inverse model.

**Options as for the 1D case**

**Zero Padding:** Embed the image in a larger image, with sufficient zeros to handle the extent of the PSF.

\[
\begin{array}{ccc}
O & O & O \\
O & X & O \\
O & O & O \\
\end{array}
\]

Introduces edges, discontinuities, leads to ringing and artificial black borders. Matrix is **block Toeplitz with Toeplitz Blocks BTTB**.

**Periodicity:** Embed the image periodically in a larger image.

\[
\begin{array}{ccc}
X & X & X \\
X & X & X \\
X & X & X \\
\end{array}
\]

Introduces edges, discontinuities, leads to ringing and artificial black borders. Matrix is **block Circulant with Circulant Blocks BCCB**. As an example recall the earlier case with $3 \times 3$ PSF

\[
\begin{pmatrix}
\begin{array}{cccc}
  h_{22} & h_{12} & h_{32} & h_{21} & h_{11} & h_{31} & h_{23} & h_{13} & h_{33} \\
  h_{32} & h_{22} & h_{12} & h_{31} & h_{21} & h_{11} & h_{33} & h_{23} & h_{13} \\
  h_{12} & h_{32} & h_{22} & h_{11} & h_{31} & h_{21} & h_{13} & h_{33} & h_{23} \\
  h_{23} & h_{13} & h_{33} & h_{22} & h_{12} & h_{32} & h_{21} & h_{11} & h_{31} \\
  h_{33} & h_{23} & h_{13} & h_{32} & h_{22} & h_{12} & h_{31} & h_{21} & h_{11} \\
  h_{13} & h_{33} & h_{23} & h_{12} & h_{32} & h_{22} & h_{11} & h_{31} & h_{21} \\
  h_{21} & h_{11} & h_{31} & h_{23} & h_{13} & h_{33} & h_{22} & h_{12} & h_{32} \\
  h_{31} & h_{21} & h_{11} & h_{33} & h_{23} & h_{13} & h_{32} & h_{22} & h_{12} \\
  h_{11} & h_{31} & h_{21} & h_{13} & h_{33} & h_{23} & h_{12} & h_{32} & h_{22}
\end{array}
\end{pmatrix}
= \begin{pmatrix}
H \circ \text{circulant}(h_1, h_{11}, h_{31}) & H_1 & H_3 \\
H_3 & H_2 & H_1 \\
H_1 & H_3 & H_2
\end{pmatrix}
\]

\[
H_1 = \text{circulant}(h_1, h_{11}, h_{31}) = \text{circulant}(h_{\cdot,1})
\]
\[
H_2 = \text{circulant}(h_{12}, h_{22}, h_{32}) = \text{circulant}(h_{\cdot,2})
\]
\[
H_3 = \text{circulant}(h_{13}, h_{23}, h_{33}) = \text{circulant}(h_{\cdot,3})
\]

and the entire PSF can be obtained from the first column of $A$.  

73
**Reflexive:** Reflect and embed.

\[
\begin{array}{ccc}
X_{lrud} & X_{ud} & X_{lrud} \\
X_{lr} & X & X_{lr} \\
X_{lrud} & X_{ud} & X_{lrud}
\end{array}
\]

Obvious reflections about the central block: no edges Matrix is a sum of BTTB, BTHB, BHTB, BHHB (H for a Hankel block.)

## 7.3 Separable PSF: \( A = cr^T \)

- \( c \) coefficients of blur across columns of the image
- \( r \) coefficients of blur across rows of the image
- \( c = (c_1, c_2, \ldots, c_m)^T \), \( r = (r_1, r_2, \ldots, r_n)^T \).

The PSF matrix \( A \) is a rank one matrix. \( A_{kj} = c_k r_j \).

\[
A = \begin{pmatrix}
c_1 r_1 & c_1 r_2 & c_1 r_3 \\
c_2 r_1 & c_2 r_2 & c_2 r_3 \\
c_3 r_1 & c_3 r_2 & c_3 r_3
\end{pmatrix}
\]

Then, noticing the reflection about the dominant axis, let

\[
A_r = \begin{pmatrix}
r_2 & r_1 & r_2 \\
r_3 & r_2 & r_1 \\
r_3 & r_2 & r_1
\end{pmatrix} \quad A_c = \begin{pmatrix}
c_2 & c_1 \\
c_3 & c_2 \\
c_3 & c_2
\end{pmatrix}
\]

\( A_r \) and \( A_c \) inherit the structure of the 1D matrices dependent on how boundary conditions are applied.

We obtain the overall blur matrix \( A = A_r \otimes A_c \).

### 7.3.1 Kronecker Product

To obtain the two dimensional PSF matrix \( A \) which acts on the image in vec form we map \( A \) using its special structure.
\[ A_r \otimes A_c = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \otimes A_c \]

\[ = \begin{bmatrix} a_{11}A_c & a_{12}A_c & \cdots & a_{1n}A_c \\ a_{21}A_c & a_{22}A_c & \cdots & a_{2n}A_c \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}A_c & a_{m2}A_c & \cdots & a_{mn}A_c \end{bmatrix} \]

7.3.2 Evaluating using the Kronecker Product

Assume \( A_c \in \mathbb{R}^{m \times m} \) and \( A_r \in \mathbb{R}^{n \times n} \), for an image of size \( m \times n \). Note that here we assume that the matrices \( A_c \) and \( A_r \) which are defined by PSFs in each dimension, are appropriately extended to the appropriate dimension, using reflexive, periodic or zero boundary conditions. Then, for the exact blurred image,

\[ G = A_cX A_r^T. \]

Left multiplication by \( A_c \) applies PSF to each column of \( X \), is vertical blur.

Left multiplication of \( X^T \) by \( A_r \) applies PSF to each column of \( X^T \), which are rows of \( X \). Hence the horizontal blur is applied as \((A_rX^T)^T = XA_r^T \)

Let \( A = A_r \otimes A_c \) be the Kronecker product of size \( mn \times mn \), the \( ij \) block of \( A \) is \((A_r)_{ij}A_c\).

Then \( G = A_cX A_r^T \) in vector form is \( \mathbf{b} = \mathbf{Ax} \).

Kronecker Product Relations

1. \( (A \otimes B) \text{vec}(X) = \text{vec}(BX A^T) \)

2. Transpose is distributive \( (A \otimes B)^T = (A^T \otimes B^T) \)

3. When both matrices are invertible \( (A \otimes B)^{-1} = A^{-1} \otimes B^{-1} \)

4. Mixed product property for matrices of appropriate dimensions \( (A \otimes B)(C \otimes D) = (AB \otimes CD) \)

5. The Kronecker product is associative, but not commutative.
7.4 Spectral Decomposition April 2, 2014

7.4.1 The general case for BCCB

\[
A(h) = \begin{pmatrix}
H_2 & H_1 & \ldots & H_4 & H_3 \\
H_3 & H_2 & H_1 & \ldots & H_4 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
H_{ny} & \vdots & H_3 & H_2 & H_1 \\
H_1 & H_{ny} & \ldots & H_3 & H_2 \\
\end{pmatrix}
\]

Each block is circulant, \( H_j = \text{circulant}(h_{.,j}) \), \( h \) of size \( n_x \times n_y \).

\( A \) is of size \( n_x n_y \times n_x n_y \), \( n = n_x n_y \). The first column defines \( h \).

Again \( A \) is normal and we have a spectral decomposition.

\[
A = F^* \text{diag}(\sqrt{n} \text{vec}(\hat{h})) F = F^* \Lambda F
\]

Here \( F = F_r \otimes F_c \) and \( \hat{h} \) is the two dimensional FFT of \( h \) of size \( n_x \times n_y \). \( \hat{h} = F(h) = (F_r \otimes F_c) h. \)

Calculation scheme: \( A \text{vec}(X) = \text{vec}(F^* (\hat{A} \ast \text{vec}(FX))). \)

As for the 1D case to find eigenvalues of \( A \) observe \( FA = \Lambda F \) and \( F_1 = (1/\sqrt{n})(1,1,\ldots,1)^T \), (look at the Kronecker product). Hence \( (\sqrt{n}FA_1)_l = \lambda_l. \)

For the BCCB first column of \( A \) gives \( h \).

Suppose center point of \( h \) in spatial domain is at index center = \( (r_0,c_0) \)

Matlab can be used to find this column \( A_1 \) and \( F \) acting on \( A_1 \) is accomplished as the 2D transform

\[
\text{fft2(circshift}(h, 1 - \text{center})).
\]

i.e. \( \text{circshift} \) performs necessary circular shift of the PSF matrix.

Note that \( \text{fft2} \) in Matlab implements \( \sqrt{n} F \) but also \( \text{ifft2} \) implements \( 1/\sqrt{n} F^{-1}. \)

7.4.2 DCT analysis extends for the 2D case with reflexive BCs

\[
A = WAW^T
\]

where \( W \) is the orthogonal two-dimensional DCT matrix

2D DCT is calculated for array \( X \) by computing DCT of each column, followed by DCT of each row.
Inverse is calculated equivalently \( A = W \Lambda^{-1} W^T \)

Use Matlab functions `dct2` and `idct2` in the image processing toolbox. Alternatively, toolbox for the HNO book!

The spectrum is obtained from \( \text{dct2}(\text{dctshift}(h, \text{center}))/\text{dct2}(\text{e}_1) \)

DCT Implementation uses real arithmetic rather than complex for the FFT.

Note cost equivalent to FFT cost.

### 7.4.3 Summary: Spectral Decomposition for (Most) PSF Matrices: BTTB, BCCB, Double Symmetric Reflexive

Given \( h \) the PSF array, \( \text{center} = (r_0, c_0) \) the central index of the PSF

Eigenvalues of \( A \) are obtained by a transform applied to shifted \( h \)

A spectral decomposition exists

\[
A = U^* \Lambda U
\]

which can be used for forward and inverse operations.

All forward and inverse operations use transforms and convolution.

We do not know anything about the ordering of the eigenvalues in \( \Lambda \) in terms of size.

### 7.5 The General Two Dimensional Spectral Decomposition

With the conversion of the data to a vector, \( x = \text{vec}(X) \), we note that we can do the reverse to convert the vector to an array, \( X = \text{array}(x) \). Thus let \( V_i = \text{array}(v_i) \),

Then

\[
x = \sum_{i=1}^{N} \frac{u_i^T b}{\sigma_i} v_i \quad \text{yields}
\]

\[
X = \sum_{i=1}^{N} \frac{u_i^T b}{\sigma_i} \text{array}(v_i)
\]

\[
= \sum_{i=1}^{N} \frac{u_i^T b}{\sigma_i} V_i
\]

This is a decomposition in the basis images \( V_i \).
7.5.1 Spectral Expansion for DFT : BCCB

$$A = F^* \Lambda F$$

Use $F = F_r \otimes F_c$, the one dimensional FT matrices

$$x = F^* \Lambda^{-1} F b = (F_r \otimes F_c)^* \Lambda^{-1} \text{vec}(F_c G F_r^T)$$

Notice $\text{vec}(F_c G F_r^T)$ is the two dimensional FT of the blurred image $G$.

Let $\text{vec}(\tilde{G}) = \Lambda^{-1} \text{vec}(F_c G F_r^T)$

$$X = F_c^* \tilde{G} (F_c^*)^T = \text{conj}(F_c) \tilde{G} \text{conj}(F_r^T)$$

$$X = \sum_i \sum_j \tilde{G}_{ij} \text{conj}(f_{c,i} f_{r,j}^T)$$

$f_{c,i}$ and $f_{r,j}$ are the $i^{th}$ and $j^{th}$ columns of $F_c$ and $F_r$. This is a decomposition in the basis images $f_{c,i} f_{r,j}^T$.

7.5.2 Spectral Expansion for the Reflexive Case

$$A = W \Lambda W^T$$

Use $W = W_r \otimes W_c$, the one dimensional DCT matrices

$$x = W \Lambda^{-1} W^T b = (W_r \otimes W_c) \Lambda^{-1} (W_r \otimes W_c)^T b$$

$$(W_r \otimes W_c) \Lambda^{-1} (W_r^T \otimes W_c^T) b = (W_r \otimes W_c) \Lambda^{-1} \text{vec}(W_c^T G W_r)$$

$\text{vec}(W_c^T G W_r)$ is the two dimensional DCT of the blurred image $G$.

Let $\text{vec}(\tilde{G}) = \Lambda^{-1} \text{vec}(W_c^T G W_r)$, then

$$X = (W_r \otimes W_c) \text{vec}(\tilde{G}) = \text{vec}(W_c \tilde{G} W_r^T) = \sum_i \sum_j \tilde{G}_{ij} (w_{c,i} w_{r,j}^T)$$

$w_{c,i}$ and $w_{r,j}$ are the $i^{th}$ and $j^{th}$ columns of $W_c$ and $W_r$. This is a decomposition in the basis images $w_{c,i} w_{r,j}^T$.  

78
7.5.3 The Separable PSF: \( A = cr^T \)

Matrix \( A \) is obtained as Kronecker product \( A = A_r \otimes A_c \)
\( A x = A_c X A_r^T \).

SVD of each matrix
\[
A_r = U_r \Sigma_r V_r^T, \quad A_c = U_c \Sigma_c V_c^T
\]

SVD of Kronecker product
\[
(U_r \Sigma_r V_r^T) \otimes (U_c \Sigma_c V_c^T) = (U_r \otimes U_c)(\Sigma_r \otimes \Sigma_c)(V_r \otimes V_c)^T
\]

Extend the one dimensional SVD solution for \( b = Ax \).
\[
X = V_c \Sigma_c^{-1} U_r^T G U_r \Sigma_r^{-1} V_r^T
\]

\( \text{vec}(U_c G U_r^T) = (U_r \otimes U_c)^T \text{vec}(G) \) represents the spectral transform of the blurred image \( G \)

Let \( \tilde{G} \) be the matrix \( \Sigma_c^{-1} U_r^T G U_r \Sigma_r^{-1} \) then
\[
X = V_c \tilde{G} V_r^T = (V_r \otimes V_c) \tilde{G} = \sum \sum \tilde{G}_{ij} (v_{c,i} v_{r,j}^T)
\]

\( v_{c,i} \) and \( v_{r,j} \) are the \( i^{th} \) and \( j^{th} \) columns of \( V_c \) and \( V_r \). This is a decomposition in the basis images \( v_{c,i} v_{r,j}^T \).

Figure 19: Example taken from Hansen, Nagy and O'Leary, Figure 5.7 Example of basis images for Middle Size PSF using SVD

![Figure 5.7. Plots of the first 16 basis images \( V_i \). Green represents positive values in \( V_i \) while red represents negative values. These matrices satisfy \( v_i = \text{vec}(V_i) \), where the singular vectors \( v_i \) are from the SVD of the matrix \( A \) for the middle PSF in Figure 5.4.](image)
Figure 20: Example of basis images: Periodic Boundary Conditions Example taken from Hansen, Nagy and O’Leary, Figure 5.8 Notice the periodicity effect on the basis images.

![Periodic Boundary Conditions Example](image)

**Figure 5.8.** The first 16 basis images $v_i$ for the PSF in Figure 5.7, with periodic boundary conditions in the blurring model.

Figure 21: Example of basis images: Reflexive Boundary Conditions Example taken from Hansen, Nagy and O’Leary, Figure 5.9 Notice the reflexive effect on the basis images.

![Reflexive Boundary Conditions Example](image)

**Figure 5.9.** The first 16 basis images $v_i$ for the PSF in Figure 5.7, with reflexive boundary conditions in the blurring model.
Figure 22: Example of basis images: Periodic BCCB using DFT Example taken from Hansen, Nagy and O’Leary, Figure 5.10 Notice that DFT basis images are quite different.

Figure 5.10. Plots of the real parts of some of the DFT-based basis images $f_{i,j}$ used when the blurring matrix $A$ is a BCCB matrix; blue and red represent, respectively, positive and negative values. The imaginary parts of the basis images have the same appearance. We used $m = n = 256$. 
Figure 23: Example of basis images: Doubly Symmetric PSF using DCT Example taken from Hansen, Nagy and O’Leary, Figure 5.11 Notice that DCT basis images are again also quite different.

7.6 Two Dimensional Problems With Generalized Regularization

Image $X$: rows in $x$- direction and columns in $y$- direction. We know that for regularization in the 1D case we introduce a new basis for the solution when using a generalized regularization. The same should occur in 2D.

Let $L$ be a mapping operator. Mapping in the $y$ direction are achieved by $LX$, and in the $x$- direction by $(LX^T)^T = XL^T$.

If $x = \text{vec}(X)$, $LX = LX I = (I \otimes L)x$ and $XL^T = IXL^T = (L \otimes I)X$.

Regularization terms $\| (I \otimes L)x \|^2$ and $\| (L \otimes I)x \|^2$

We note that

$$\| (I \otimes L)x \|^2 + \| (L \otimes I)x \|^2 = \left\| \begin{pmatrix} I & L \\ L & I \end{pmatrix} x \right\|_2^2$$

is not equivalent to

$$\| ((I \otimes L) + (L \otimes I))x \|^2$$

When $L$ is the first derivative the second expression is the second order Laplacian.
7.6.1 Two Dimensions: Assume image is periodic apply Tikhonov Regularization

When $A$ is the PSF matrix with periodic boundary conditions, and implementable using DFT matrices $F = F_r \otimes F_c$, the regularized Tikhonov solution can be written as

$$x_{\lambda,L} = F^* (|\Lambda_A|^2 (|\Lambda_A|^2 + \lambda^2 \Delta)^{-1} \Lambda_A^{-1}) F b$$

$\Delta$ is the identity

$\Lambda_A$ is matrix of eigenvalues of $A$.

$|\Lambda_A|^2 (|\Lambda_A|^2 + \lambda^2 \Delta)^{-1} \Lambda_A$ is diagonal and $|\Lambda_A|^2 (|\Lambda_A|^2 + \lambda^2 \Delta)^{-1}$ is the filtering matrix.

This corresponds to Wiener filtering.

7.6.2 Two Dimensional Problems using the Fourier Transform

Again suppose $A$ implemented using $F = F_r \otimes F_c$, then derivatives are mapped to Fourier domain

$$(I \otimes L) \rightarrow F^* (I \otimes \Lambda_L) F \quad (L \otimes I) \rightarrow F^* (\Lambda_L \otimes I) F$$

$$
\begin{pmatrix}
I \otimes L \\
L \otimes I
\end{pmatrix}
= 
\begin{pmatrix}
F^* & 0 \\
0 & F^*
\end{pmatrix}
\begin{pmatrix}
I \otimes \Lambda_L \\
\Lambda_L \otimes I
\end{pmatrix}
F
$$

and thus regularization terms can all be expressed using the factorization, yielding

$$x_{\lambda,L} = F^* (|\Lambda_A|^2 (|\Lambda_A|^2 + \lambda^2 \Delta)^{-1} \Lambda_A^{-1}) F b$$

where now $\Delta$ depends on the regularizer.
8 Iterative methods April 14, 2014

8.1 Background

8.1.1 Motivation

For ill-posed problems impossible to construct perfect reconstruction of $\hat{x}$

Direct solution methods demonstrate that approximation to $\hat{x}$ lies in low dimensional subspace of $\mathbb{R}^n$.

SVD can be used to express solution with respect to a basis

For convolution problems the basis may be a cosine or Fourier basis.

The bases capture the low frequency components of $\hat{x}$.

GSVD can be used to express solution with respect to a mapped basis

For large scale problems it is not feasible to find the entire SVD.

Can we find a basis which captures the low frequency information appropriately?

Suppose we are provided with a basis for the solution $x$, e.g. $\mathcal{K}_k = \text{span}\{p_1, \ldots, p_k\}$ which are columns of a matrix $P_k$ of size $n \times k$. Then we may write the approximation $x^{(k)} = P_k y$ for some vector $y$ of size $k$, where we now explicitly assume that the solution $x^{(k)}$ is the $k$th iterative approximation to $\hat{x}$. Therefore finding the solution $x^{(k)}$ which solves the least squares problem over the space spanned by $\mathcal{K}_k$ can be expressed as the constrained least squares problem

$$
\begin{align*}
x^{(k)} &= \arg \min_x \{ \|Ax - b\|_2 \text{ s.t. } x \in \mathcal{K}_k \} \quad \text{implies with } x = P_k y \\
y^{(k)} &= \arg \min_y \{ \|AP_k y - b\|_2 \}
\end{align*}
$$

(38)

(39) is the projected problem obtained by projecting the original problem to a smaller subspace.

Given $P_k$, which is not too large, the solution can be expressed as the solution of the normal equations of size $k \times k$.

Given knowledge of the properties for $x$, one may be able to find an appropriate basis - e.g. for a symmetric solution the basis should be symmetric.

Moreover, the question becomes how to best pick this basis so that we obtain a good approximation for the solution. Notice that we may write

$$
x^{(k)} = \sum_i \alpha_i P_i = x^{(k-1)} + \alpha_k p_k, \quad y^{(k)} = (\alpha_1, \alpha_2, \ldots, \alpha_k)^T
$$
Thus, suppose we have $x^{(k-1)}$ to proceed we need to determine how to obtain the next vector in the underlying space. We want the next vector so that we get a solution which better minimizes (39), but we know how to solve the normal equations given by (39).

8.2 Conjugate Gradients

We note first the connection between finding the minimum of

$$\phi(x) = \frac{1}{2}x^TAx - x^Tb$$

and solving $Ax = b$ which follows from solving $\nabla \phi = 0$ $A$ is SPD.

Suppose we have solution $x^{(k-1)}$ and wish to find an update

$$x^{(k)} = x^{(k-1)} + \alpha_k p_k$$

which minimizes $\phi$ with respect to the scalar $\alpha_k$. In particular, by definition

$$\phi(x^{(k)}) = \phi(x^{(k-1)} + \alpha_k p_k) = \frac{1}{2}(x^{(k-1)} + \alpha_k p_k)^T A (x^{(k-1)} + \alpha_k p_k) - (x^{(k-1)} + \alpha_k p_k)^T b$$

$$= \frac{1}{2} \left( (x^{(k-1)})^T A x^{(k-1)} + 2\alpha_k p_k^T A x^{(k-1)} + \alpha_k^2 p_k^T A p_k \right) - (x^{(k-1)} + \alpha_k p_k)^T b$$

$$= \frac{1}{2} (x^{(k-1)})^T A x^{(k-1)} - (x^{(k-1)})^T b + \alpha_k p_k^T (A x^{(k-1)} - b) + \frac{1}{2} \alpha_k^2 p_k^T A p_k$$

$$= \phi(x^{(k-1)}) - \alpha_k p_k^T r^{(k-1)} + \frac{1}{2} \alpha_k^2 \| p_k \|^2_A, \quad r^{(k)} = b - A x^{(k)}$$

where we use the definition $p_k^T A p_k = \| p_k \|^2_A$, and $r^{(k-1)} = A x^{(k-1)} - b$. We now notice that this depends only on the scalar $\alpha_k$ given the direction $p_k$. Minimizing with respect to $\alpha$, for the function $\phi(\alpha) = b/2\alpha^2 - a\alpha$, the minimum is at $\phi'(\alpha) = b\alpha - a = 0$ or $\alpha = a/b$, with $a = p_k^T r^{(k-1)}$ and $b = \| p_k \|^2_A$ we obtain

$$\alpha_k = \frac{p_k^T r^{(k-1)}}{\| p_k \|^2_A} \quad \text{note by SPD that } \| p_k \|^2_A > 0$$

$$\phi(x^{(k-1)} + \alpha_k p_k) = \phi(x^{(k-1)}) - \frac{1}{2} \left( \frac{p_k^T r^{(k-1)}}{\| p_k \|^2_A} \right)^2$$

$$< \phi(x^{(k-1)}), \quad p_k^T r^{(k-1)} \neq 0.$$  \hspace{1cm} (42)

Hence we know that the basis vector $p_k$ should not be orthogonal to the residual vectors, otherwise there is no decrease in the functional. Thus finding a good solution now depends on finding the suitable basis.

Suppose that we have an initial value $x^{(0)}$ then we know that the iterative solutions are given by

$$x^{(k)} \in x^{(0)} + \operatorname{span}\{p_1, \ldots, p_k\}$$

85
If the search directions $p_k$ are linearly independent and $x^{(k)}$ solves the problem
\[
x^{(k)} = \arg\min_x \{ \phi(x) : x \in x^{(0)} + \text{span}\{p_1, \ldots, p_k\} \}
\]
then the iteration has to converge in at most $n$ steps, because the iteration $x^{(n)}$ minimizes $\phi$ over $\mathbb{R}^n$ and is thus a solution to the system.

The search directions must be easy to compute, i.e. we must be able to compute $x^{(k)}$ easily given $x^{(k-1)}$. Let
\[
x^{(k)} = x^{(0)} + P_{k-1}y + \alpha p_k,
\]
then we first note that
\[
r^{(k-1)} = b - Ax^{(k-1)} = b - A(x^{(0)} + P_{k-1}y) = r^{(0)} - AP_{k-1}y
\]
\[
p_k^T r^{(k-1)} = p_k^T (r^{(0)} - AP_{k-1}y) = p_k^T r^{(0)} - (p_k^T AP_{k-1})y = p_k^T r^{(0)}
\]
if the $p_k$ are $A-$ conjugate. In this case we now see
\[
\phi(x^{(k)}) = \phi(x^{(0)} + P_{k-1}y + \alpha p_k)
= \frac{1}{2}(x^{(0)} + P_{k-1}y + \alpha p_k)^T A(x^{(0)} + P_{k-1}y + \alpha p_k) - (x^{(0)} + P_{k-1}y + \alpha p_k)^T b
= \left( \frac{1}{2}(x^{(0)} + P_{k-1}y)^T A(x^{(0)} + P_{k-1}y) - (x^{(0)} + P_{k-1}y)^T b \right) + \alpha p_k^T A(x^{(0)} + P_{k-1}y) + \frac{1}{2} \alpha (p_k^T A p_k - (p_k^T b)^2)
= \phi(x^{(0)} + P_{k-1}y) - \alpha p_k^T r^{(0)} + \frac{1}{2} \alpha^2 \|p_k\|^2_A + \alpha p_k^T AP_{k-1}y.
\]
and the choice of $A-$ conjugacy for $p_k$, namely that $p_k$ should be chosen in the space orthogonal to $\text{span}\{Ap_1, \ldots, Ap_{k-1}\}$ again simplifies this so that
\[
\phi(x^{(k)}) = \phi(x^{(0)} + P_{k-1}y) - \alpha p_k^T r^{(0)} + \frac{1}{2} \alpha^2 \|p_k\|^2_A
\]
\[
\min_{x^{(k)}} \phi(x^{(k)}) = \min_{y, \alpha} \{ \phi(x^{(0)} + P_{k-1}y + \alpha p_k) \}
= \min_y \{ \phi(x^{(0)} + P_{k-1}y) \} + \min_{\alpha} \left\{ \frac{1}{2} \alpha^2 \|p_k\|^2_A - \alpha p_k^T r^{(0)} \right\}
= \phi(x^{(0)} + P_{k-1}y) - \frac{1}{2} \frac{p_k^T r^{(0)}}{\|p_k\|^2_A}.
\]
and the problem, separates into a pair of uncoupled minimizations.

Suppose that $y$ solves the first problem then $x^{(k-1)}$ minimizes $\phi$ over $x^{(0)} + \text{span}(P_{k-1})$. The solution to the second scalar problem is given by (41) but we note
\[
p_k^T r^{(k-1)} = p_k^T (b - Ax^{(k-1)}) = p_k^T b - p_k^T A(x^{(0)} + P_{k-1}y) = p_k^T b - p_k^T AP^{(0)} = p_k^T r^{(0)},
\]
by the $A-$conjugacy of the vectors $p_k$. We have an iterative scheme to update $x$, $x^{(k)} = x^{(k-1)} + \alpha_k p_k$ as initially desired, provided that we may find

$$p_k \in \text{span}\{Ap_1, \ldots, Ap_{(k-1)}\} \perp p_k^T r^{(k-1)} \neq 0.$$ 

For the latter condition, refer to Lemma 10.2.1, Golub and Van Loan, Edition 3.

### 8.2.1 Choosing the search direction

Suppose that $p_k$ is found that minimizes the distance of $p$ to $r^{(k-1)}$ over all vectors that satisfy the $A$-conjugacy:

$$p_k = \arg \min_{p \in \text{span}\{Ap_1, \ldots, Ap_{k-1}\}} \{\|p - r^{(k-1)}\|_2\}.$$ 

We have to show how to find $p_k$ efficiently. First assume that $p_1 = r^{(0)}$ and we obtain $p_k$ using the recurrence

$$p_k = r^{(k-1)} - AP_{k-1}z_{k-1}, \quad z_{k-1} = \arg \min_z \{\|r^{(k-1)} - AP_{k-1}z\|_2\} \quad (43)$$ 

We note that this choice yields a small $\|p_k\|$, by the definition of $z^{(k-1)}$, which is desirable so that $x^{(k)}$ converges, from the update equation (40). We need to show that this choice satisfies the $A-$conjugacy and is closest to the latest residual. Since $z_{k-1}$ solves the least squares problem, it satisfies the relevant normal equations

$$(AP_{k-1})^T r^{(k-1)} = (AP_{k-1})^T (AP_{k-1})z_{k-1}$$

$$0 = (AP_{k-1})^T (r^{(k-1)} - AP_{k-1}z_{k-1}) = (AP_{k-1})^T p_k.$$ 

Thus this satisfies the necessary $A-$conjugacy. Moreover,

$$p_k = r^{(k-1)} - AP_{k-1}z_{k-1} = (I - (AP_{k-1})(AP_{k-1})^\dagger)r^{(k-1)}$$ 

is by definition the orthogonal projection of $r^{(k-1)}$ into range$(AP_{k-1})^\perp$, which is also therefore the closest vector in this space to $r^{(k-1)}$. Hence we have the correct choice for $p_k$. With the choice for the $\alpha_k$ this therefore leads to finding the effective stable algorithm. We connect the approach with the Krylov approach.

### 8.2.2 Krylov Subspace

**Definition 9 (Krylov Subspace).** Given an $n \times n$ matrix $A$, a vector $b$ and an integer $k$ the Krylov subspace depending on $A$, $b$ and $k$ is denoted by

$$K_k(A, b) = \{b, Ab, A^2b, \ldots, A^{k-1}b\} \quad (45)$$
We want to connect the solution of the CG iteration with the Krylov subspace dependent on the initial residual $r^{(0)}$. First though, why is this important? We observe the solution of the square linear system $Ax = b$ uses the expansion of $A^{-1}$ using its minimal polynomial (see 6.3 Least Squares Data Fitting with Applications, Hansen et al 2012)

$$q(t) = \prod_{j=1}^{k} (t - \lambda_j)^{m_j}$$

where $\lambda_j$ are the distinct eigenvalues of $A$ with multiplicity $m_j$, $\sum_j m_j = n$. Then for coefficients $\gamma_i$

$$A^{-1} = \sum_{i=0}^{n-1} \gamma_i A^i$$

and the solution may be expressed as

$$x = A^{-1}b = \sum_{i=0}^{n-1} \gamma_i A^i b \in \mathcal{K}_n\{A, b\}.$$

The dimension of the space is determined by the degree of $q(t)$ and hence by the spectrum of $A$. If the eigenvalues are clustered with independent eigenvectors then the underlying space is smaller. Hence if we use an element of this subspace to provide a solution to the system of equations, we will have good convergence if the eigenvalues are clustered rather than well-separated.

8.2.3 Connecting CG and Krylov Subspace April 15, 2014

**Theorem 8.1** (CG iterations Golub and Van Loan Edition 3, Thm 10.2.3). The following relations hold:

1. $$r^{(k)} = r^{(k-1)} - \alpha_k A p_k$$ (46)

2. $$P_k^T r^{(k)} = 0$$ (47)

3. $$\text{span}\{p_1, \ldots, p_k\} = \text{span}\{r^{(0)}, \ldots, r^{(k-1)}\} = \mathcal{K}_k(A, r^{(0)})$$ (48)

4. $$(r^{(j)})^T r^{(k)} = 0.$$ (49)

**Proof.** 1. This is easy using the update equation (40) for $x^{(k)}$. Notice that $x^{(k)} = x^{(k-1)} + \alpha_k p_k$ gives $A x^{(k)} = A x^{(k-1)} + \alpha_k A p_k$ and $b - Ax^{(k)} = b - Ax^{(k-1)} - \alpha_k A p_k$ and the result follows.
2. Now use $x^{(k)} = x^{(0)} + P_k y^{(k)}$ where $y^{(k)}$ minimizes $\phi(x^{(0)} + P_k y)$. But

\[
\begin{align*}
\phi(x^{(0)} + P_k y) &= \frac{1}{2}(x^{(0)} + P_k y)^T A(x^{(0)} + P_k y) - (x^{(0)} + P_k y)^T b \\
&= \phi(x^{(0)}) + (P_k y)^T (A x^{(0)} - b) + (P_k y)^T A P_k y \\
&= \phi(x^{(0)}) - (P_k y)^T r^{(0)} + (P_k y)^T A P_k y.
\end{align*}
\]

Given that the first term is fixed, this functional is minimized by minimizing the second term $-(P_k y)^T r^{(0)} + (P_k y)^T A P_k y$. Indeed if this term is zero, then the functional is minimized. But this then means that $y$ solves the system $P_k^T A P_k y = P_k^T r^{(0)}$. Hence

\[
0 = P_k^T (r^{(0)} - A P_k y) = P_k^T (b - A (x^{(0)} + P_k y)) = P_k^T (b - A x^{(k)}) = P_k^T r^{(k)}.
\]

3. To connect the subspaces we first note that by (46)

\[
\{A p_1, \ldots, A p_{k-1}\} \subseteq \{r^{(0)}, \ldots, r^{(k-1)}\}.
\]

Now we have by (44) and by (50)

\[
p_k = r^{(k-1)} - A P_{k-1} z_{k-1} \in \text{span}\{r^{(0)}, \ldots, r^{(k-1)}\}.
\]

Therefore for some upper triangular matrix $T$ we have

\[
P_k = [r^{(0)}, \ldots, r^{(k-1)}] T
\]

and since the search directions are independent $T$ is nonsingular. Hence

\[
\text{span}\{p_1, \ldots, p_k\} = \text{span}\{r^{(0)}, \ldots, r^{(k-1)}\}.
\]

Moreover, by (46)

\[
r^{(k)} \in \text{span}\{r^{(k-1)}, A p_k\} \subseteq \{r^{(k-1)}, A r^{(0)}, \ldots, A r^{(k-1)}\}.
\]

We can then apply induction on to obtain the relationship in (48).

4. Finally by (47) $r^{(k)}$ is orthogonal to any vector in range of $P_k$, which by (48) also contains $r^{(0)}, \ldots, r^{(k-1)}$. The orthogonality of the residuals follows.

This theorem eventually allows us to find that the search directions are obtained by a simple update $p_k \in \text{span}\{p_{k-1}, r^{(k-1)}\}$, for $k \geq 2$.

**Theorem 8.2** (Golub and van Loan Edition 3 Corollary 10.2.4). $p_k \in \text{span}\{p_{k-1}, r^{(k-1)}\}$, for $k \geq 2$. 89
Proof. First note $p_1 = r^{(0)}$ and $p_2 \in \text{span}\{r^{(0)}, r^{(1)}\} = \text{span}\{p_1, r^{(1)}\}$. For $k > 2$ we have

$$p_k = r^{(k-1)} - A p_{k-1} z_{k-1} = r^{(k-1)} - A p_{k-2} w - \mu A p_{k-1}$$

by partition of $z_{k-1}$

$$= r^{(k-1)} - A p_{k-2} w + \frac{\mu}{\alpha_{k-1}} (r^{(k-1)} - r^{(k-2)}) \quad \text{by (46)}$$

$$= \left(1 + \frac{\mu}{\alpha_{k-1}}\right) r^{(k-1)} + s^{(k-1)}$$

where $s^{(k-1)} = -A p_{k-2} w - \frac{\mu}{\alpha_{k-1}} r^{(k-2)}$.

Now we obtain

$$s^{(k-1)} \in \text{span}\{r^{(k-2)}, A p_{k-2} w\}$$

$$\subseteq \text{span}\{r^{(k-2)}, A p_1, \ldots, A p_{k-2}\}$$

$$\subseteq \text{span}\{r^{(k-2)}, \ldots, r^{(0)}\}.$$ 

Now the residuals are mutually orthogonal and so $s^{(k-1)}$ is orthogonal to $r^{(k-1)}$. Hence we note immediately

$$\|p_k\|^2 = \left(1 + \frac{\mu}{\alpha_{k-1}}\right)^2 \|r^{(k-1)}\|^2 + \|s^{(k-1)}\|^2$$

and we can chose $\mu$ so that this norm is minimum for the first term. For the second term we notice that the minimum for $\|s^{(k-1)}\| = \|\mu/\alpha_{k-1} r^{(k-2)} + A p_{k-2} w\|$ is given from (43) by $w$ with

$$s^{(k-1)} = -\frac{\mu}{\alpha_{k-1}} r^{(k-2)} - A p_{k-2} w$$

$$= -\frac{\mu}{\alpha_{k-1}} \left(r^{(k-2)} - A p_{k-2} \frac{-\alpha_{k-1}}{\mu} w\right)$$

$$= -\frac{\mu}{\alpha_{k-1}} \left(r^{(k-2)} - A p_{k-2} z_{k-2}\right), \quad w = -\frac{\mu}{\alpha_{k-1}} z_{k-2}$$

$$= -\frac{\mu}{\alpha_{k-1}} p_{k-1},$$

where the last equality arises from the CG requirement that $p_{k-1}$ minimizes the given residual. Therefore $s^{(k-1)} \propto p_{k-1}$ and we obtain

$$p_k \in \text{span}\{r^{(k-1)}, p_{k-1}\}$$

$\Box$

### 8.2.4 Efficient Update Schemes

We now know that we may write $p_k = r^{(k-1)} + \beta_k p_{k-1}$ and want to calculate this efficiently, which requires the calculation of $\beta_k$. Observe from $A-$conjugacy
\[ 0 = p_{k-1}^T A p_k = p_{k-1}^T A (r^{(k-1)} + \beta_k p_{k-1}) \]
yields
\[ \beta_k = -\frac{p_{k-1}^T A r^{(k-1)}}{p_{k-1}^T A p_k} \]
which requires two matrix-vector multiplications.

Now note that
\[ r^{(k)} = b - Ax^{(k)} = b - A(x^{(k-1)} + \alpha_k p_k) = r^{(k-1)} - \alpha_k A p_k \]
\[ (r^{(k-1)})^T r^{(k-1)} = -\alpha_{k-1} (r^{(k-1)})^T A p_{k-1} \]
\[ 0 = (r^{(k-2)})^T r^{(k-2)} - \alpha_{k-1} (r^{(k-2)})^T A p_{k-1} \]
\[ = (r^{(k-2)})^T r^{(k-2)} - \alpha_{k-1} (p_{k-1} - \beta_{k-1} p_{k-2})^T A p_{k-1} \]
\[ = (r^{(k-2)})^T r^{(k-2)} - \alpha_{k-1} (p_{k-1})^T A p_{k-1} \]
\[ \beta_k = \frac{(r^{(k-1)})^T r^{(k-1)}}{(r^{(k-2)})^T r^{(k-2)}} \]
yields the coefficient by two inner products rather than two matrix vector multiplications.

We have in summary the updates
\[ \beta_k = \frac{(r^{(k-1)})^T r^{(k-1)}}{(r^{(k-2)})^T r^{(k-2)}} = \frac{\|r^{(k-1)}\|^2}{\|r^{(k-2)}\|^2} \] (51)
\[ p_k = r^{(k-1)} + \beta_k p_{k-1} \] (52)
\[ \alpha_k = \frac{p_k^T r^{(k-1)}}{p_k^T A p_k} = \frac{p_k^T r^{(k-1)}}{\|p_k\|^2_A} \] (53)
\[ x^{(k)} = x^{(k-1)} + \alpha_k p_k \] (54)
\[ r^{(k)} = r^{(k-1)} - \alpha_k A p_k \] (55)
which uses only one matrix-vector update per iteration.

**Algorithm 1** CG for SPD $A$ with initial guess $x^{(0)}$.

1: $k = 0$, $r^{(0)} = b - A x^{(0)}$, $p_1 = r^{(0)}$
2: while $r^{(k)} \neq 0$ do
3: \quad $k = k + 1$
4: \quad Update $\beta_k$ by (51)
5: \quad Update $p_k$ by (52)
6: \quad Update $\alpha_k$ by (53)
7: \quad Update $x^{(k)}$ by (54)
8: \quad Update $r^{(k)}$ by (55)
9: end while
Clearly the algorithm 1 requires improved stopping conditions. This is a topic of particular interest for ill-posed problems.

**Theorem 8.3** (Convergence of CG with respect to the weighted $A$ norm $\|x\|_A = \sqrt{x^T Ax}$). See e.g. Hansen, Pereyra, Scherer Section 6.3 or Golub Theorem 10.2.6, Edition 3.

$$\|\hat{x} - x^{(k)}\|_A \leq 1 \|x - x^{(0)}\|_A \left(\frac{\kappa(A) - 1}{\kappa(A) + 1}\right)^k$$

### 8.3 Krylov Methods for the Least Squares Problem April 21 2014

It is immediate from Theorem 8.3 that the convergence of CG depends on the condition of the matrix, hence applying for the normal equations yields potentially slow convergence due to the squaring of the condition number. The obvious common variant of Algorithm 1 that applies for the normal equations depends on noting that the residual for the normal equations is obtained via

$$A^T (b - Ax) = A^T r.$$ 

Thus we can modify the updates (51), (52), and (53) as

$$\beta_k = \frac{(A^T r^{(k-1)})^T A^T r^{(k-1)}}{(A^T r^{(k-2)})^T A^T r^{(k-2)}}$$

$$p_k = A^T r^{(k-1)} + \beta_k p_{k-1}$$

$$\alpha_k = \frac{\|A^T r^{(k-1)}\|^2}{p_k^T A^T A p_k}$$ (58)

in Algorithm 1 in which the initialization replaces $p_1$ by $p_1 = A^T r^{(0)}$ giving the algorithm Conjugate Gradient Normal Equation Residual (CGNR).

From the derivation of CG it is immediate that CGNR generates the iterate $x^{(k)}$ that minimizes

$$\phi_1(x) = \frac{1}{2} x^T (A^T A) x - x^T (A^T b) \text{ over } x^{(0)} + K(A^T A, A^T r^{(0)}).$$ (59)

Since $2\phi_1(x) + \|b\|^2 = \|Ax - b\|^2$, $x^{(k)}$ minimizes the least squares cost functional. Algorithm CGNR is also called CG Least Squares (CGLS) in many places.

It is important to note that the implementation details can drastically impact the stability of the algorithm. Observe for example that equations (56), (57) and (58), suggest that it would be sufficient to directly update $q_k := A^T r^{(k)}$ and throughout use $q_k$ rather than $r^{(k)}$. It has been shown in the literature that this approach is much less stable particularly for large condition for $A$.

For least squares problems a typical starting value is $x^{(0)} = 0$ so that $r^{(0)} = b$. 92
8.3.1 Properties of the CGLS iterates

By Theorem 8.1 for \( x^{(0)} = 0 \) we know that the underlying subspace for the iteration is the Krylov subspace \( K_k(A^T A, A^T b) \), and therefore although not given explicitly as such we may write the solution

\[
x^{(k)} = \sum_{i=0}^{k-1} c_i (A^T A)^i A^T b
\]

for some constants \( c_i \). Use the SVD for \( A \) to observe

\[
A^T A = (V \Sigma^T U^T U \Sigma V^T) = V (\Sigma^T \Sigma) V^T := V \tilde{\Sigma}^2 V^T, \quad \tilde{\Sigma} = \text{diag}(\sigma_i^2)
\]

(60)

\[
(A^T A)^i = (A^T A)(A^T A)^{i-1} = V \Sigma^2 V^T V \Sigma^{2(i-1)} V^T = V \Sigma^{2i} V^T
\]

(61)

yielding

\[
x^{(k)} = \sum_{i=0}^{k-1} c_i V (\tilde{\Sigma}^{2i} \Sigma^T) U^T b
\]

\[
= V \left( \sum_{i=0}^{k-1} c_i \tilde{\Sigma}^{2i} \Sigma^T \right) U^T b
\]

\[
= \sum_{j=1}^{p} \phi_j \frac{u_j^T b}{\sigma_j} v_j \quad \text{assuming rank } p \quad \text{and} \quad \phi_j = \sum_{i=1}^{k} c_{i-1} \sigma_j^{2i}.
\]

Equivalently \( x^{(k)} = V \Phi_k \Sigma^j U^T b \) where \( \Phi_k \) is a filter matrix that depends on step \( k \). Clearly the CGLS solution has a filtered SVD expansion, with filter factors that are polynomial in the squared singular values \( \sigma_i^2 \). Moreover, the contributions due to the small components \( u_j^T b \) are less significant in the iteration. Indeed one sees that there is a regularizing effect of the CGLS through the filter matrix \( \Phi_k \) so that with each iteration the more dominant frequencies will be included at the loss of high frequency (small \( \sigma_i \) ) information.

8.4 Methods based on Lanczos Bidiagonalization

Suppose we are given a rectangular matrix \( A \) of size \( m \times n \) and we transform \( A \) to lower bidiagonal form yielding the factorization, with \( U \) of size \( m \times n + 1 \), \( V \) of size \( n \times n \) and \( B_n \) of size \( n + 1 \times n \),

\[
A = U \begin{pmatrix} B_n \\ 0 \end{pmatrix} V^T, \quad U^T U = I_m, \quad V^T V = I_n U^T A = \begin{pmatrix} B_n \\ 0 \end{pmatrix} V^T
\]

(62)

\[
(i) : \quad A^T U = V \begin{pmatrix} B_n^T, 0 \end{pmatrix} \quad (ii) : \quad AV = U \begin{pmatrix} B_n \\ 0 \end{pmatrix}
\]
\[ B_n = \begin{bmatrix} \alpha_1 & \beta_2 & \alpha_2 & \cdots & \beta_n & \alpha_n & \beta_{n+1} \end{bmatrix} \]

To obtain the entries in \( B_n \) we equate columns in (62) (i) and (ii)

\((i)\) : \( A^T u_j = (V B_n^T) j = ((B_n V^T)_{\text{row} j})^T = \beta_j v_{j-1} + \alpha_j v_j \quad j = 1 : n \quad \beta_1 = \alpha_{n+1} = 0 \)

\((ii)\) : \( A v_j = \alpha_j u_j + \beta_{j+1} u_{j+1} \quad j = 1 : n \)

From which given an initial vector \( u_1 \) and with \( \alpha_j \neq 0, \beta_{j+1} \neq 0 \), we obtain the recursion

\[ v_j = \frac{A^T u_j - \beta_j v_{j-1}}{\alpha_j} \quad j = 1 : n \]

\[ u_{j+1} = \frac{A v_j - \alpha_j u_j}{\beta_{j+1}} \quad j = 1 : n \]

From here we see this does not completely define the entries. We now impose the orthogonality conditions on each of the vectors and replace the updates by

\[ v_j = A^T u_j - \beta_j v_{j-1}, \quad \alpha_j = \|v_j\|_2, \quad v_j = v_j / \alpha_j \quad j = 1 : n \quad (63) \]

\[ u_{j+1} = A v_j - \alpha_j u_j, \quad \beta_{j+1} = \|u_{j+1}\|_2, \quad u_{j+1} = u_{j+1} / \beta_{j+1} \quad j = 1 : m \quad (64) \]

with the assumption that the initial vector is initialized \( \|u_1\|_2 = 1 \). Clearly the vectors satisfy

\[ u_j \in \mathcal{K}_j(A A^T, u_1) \quad v_j \in \mathcal{K}_j(A^T A, A^T u_1). \quad (65) \]

### 8.4.1 LSQR Algorithm (Paige and Saunders - Golub-Kahan bidiagonalization)

Suppose now that we start the algorithm with the initial vector \( u_1 = b / \|b\|_2, \beta_1 = \|b\|_2 \)

then we obtain from (65)

\[ u_j \in \mathcal{K}_j(A A^T, b) \quad v_j \in \mathcal{K}_j(A^T A, A^T b). \quad (66) \]

Thus for the solution \( x^{(k)} \) of the least squares problem we know we want \( x^{(k)} \in \mathcal{K}_k(A^T A, A^T b) = \text{span}\{v_1, \ldots, v_k\} \) and we can look at the projected solution \( x^{(k)} = V_k y^{(k)} \), \( V_k \) of size \( n \times k \).

We obtain, from the definition of \( u_1 \),

\[ A x^{(k)} = A V_k y^{(k)} = U_{k+1} B_k y^{(k)} \]

\[ b - A x^{(k)} = \beta_1 u_1 - U_{k+1} B_k y^{(k)}, \quad U_{k+1} \text{ of size } m \times k + 1, \]

\[ = U_{k+1} w_{k+1} \quad w_{k+1} = \beta_1 e_1 - B_k y^{(k)}. \]
Hence to find the optimal solution that minimizes \( \|Ax - b\| \) over \( x \in \mathcal{K}_k(A^TA, A^Tb) \) we use orthogonality of \( U_{k+1} U_{k+1}^T = I_{k+1} \) to give

\[
\min \|b - Ax^{(k)}\| = \min \|U_{k+1}w_{k+1}\| = \min \|w_{k+1}\| = \min \|\beta_1 e_1 - B_k y^{(k)}\|
\]

\[
y^{(k)} = \beta_1 B_k^T e_1.
\]

The iterative process is the LSQR algorithm of Paige and Saunders, when augmented with a factorization process for the calculation of \( y^{(k)} \). For this note that the matrix \( B_k \) is bidiagonal and so the solution can be found efficiently using the QR factorization, where the resulting \( R \) is upper bidiagonal and efficiently computed with Givens rotations applied to \( B_k \).

We have shown indirectly that to exact precision the following holds.

**Theorem 8.4.** CGLS and LSQR yield the same sequence of iterates to the solution of the least squares problem over iterates that are in the space spanned by the columns of \( V_k \).

### 8.4.2 Projected solution or direct solution April 22, 2014

We have shown that direct methods, e.g. SVD, FFT, obtain the solution in a basis that is determined by the matrix \( A \).

Iterative methods yield a solution with a basis determined by the right hand side, starting with \( A^Tb \).

It is clear still that noise will enter the solution and hence we must regularize the projected solution.

### 8.4.3 Regularizing the Projected Solution

Regularizing the projected problem we solve

\[
\|B_k y^{(k)} - e_1 \beta_1\|^2 + \lambda^2 \|y^{(k)}\|^2
\]

Relating to the original full problem we see

\[
\|Ax - b\|^2 + \lambda^2 \|x\|^2 = \|B_k y^{(k)} - e_1 \beta_1\|^2 + \lambda^2 \|V y\|^2 = \|B_k y^{(k)} - e_1 \beta_1\|^2 + \lambda^2 \|y^{(k)}\|^2,
\]

which follows by the orthogonality of \( V \). Hence regularizing and projecting is the same as projecting with regularization, of course when the regularizer is based on \( \|x\| \).

The regularized problem is of size \( k + 1 \times k \). \( k \ll m \) and can be solved more efficiently than the original problem.

Solving the regularized projected problem leads to a *hybrid* method in which one may only solve the regularized version of the problem every certain number of steps \( k \). For a given regularization parameter choice method, if the regularization parameter converges we assume that the *best* solution has been obtained.
On the other hand, the mechanisms for the regularization choice methods are not immediately applicable.

Nagy uses a weighted GCV - with a fudge factor
the $\chi^2$ principle can be extended but the number of degrees of freedom is not clear.

Noise revealing properties can be utilized.

Generalized Regularization Consider also the case with regularizer $\|Lx\|$, then comparing both forms with size of $L_k$ chosen consistently as size $n \times k$:

$$
\|Ax^{(k)} - b\|^2 + \lambda^2 \|L_n x^{(k)}\|^2 = \|w^{(k)}\|^2 + \lambda^2 \|L_n V_k y^{(k)}\|^2
$$

$$
\|B_k y^{(k)} - e_1\|^2 + \lambda^2 \|L_k y^{(k)}\|^2 = \|w^{(k)}\|^2 + \lambda^2 \|L_k V_k y^{(k)}\|^2,
$$

so that solving the normal equations for $y^{(k)}$ in each case

$$
(B_k^T B_k + \lambda^2 (L_n V_k)^T (L_n V_k)) y^{(k)} = \beta B_k^T e_1
$$

$$
(B_k^T B_k + \lambda^2 (L_k^T L_k)) y^{(k)} = \beta B_k^T e_1
$$

which cannot give the same solution unless we use regularizer $L_k = L_n V_k$ which makes the solution of the projected regularized problem expensive, because the typically sparse $L_n$ is replaced by a dense $L_k$, while the system matrix $B_k$ is sparse!

Standard Form In order to avoid the problem with the calculation of $L_k = L_n V_k$ the underlying fit to data functional for the generalized regularization case is replaced to use a modified matrix $\bar{A}$, obtained using the so-called standard form transformation. In this case one needs not only $L_n$ but also the matrix which spans its null-space, but this is not difficult for typical choices for matrix $L_n$, for example $L_n$ may be used to approximate the first order derivative in which case the constant vector is in the null space. Suppose that $H$ is the matrix formed of the orthonormal basis for the matrix $L_n$, where $L_n$ approximates the first order derivative, then $H$ consists of one vector of length $n$, is constant, with each entry $1/\sqrt{n}$, e.g. $H = 1/\sqrt{n}$ ones(n, 1). For details of implementing the standard form transformation, see for example Hansen, 1998.

8.4.4 Propagation of the noise in the Krylov Solution (Not covered)

Clearly the noise enters the solution in a different manner. Suppose $n \sim (0, C_n)$ where $n$ is the noise in $b$. Then $A^T n \sim (0, A^T C_n A)$, and by (61)

$$
\sum_{i=0}^{k-1} c_i (A^T A)^i A^T n = \sum_{i=0}^{k-1} c_i n_i, \quad n_i = (A^T A) n_{i-1}, \ i = 1 : k - 1 \ n_0 = A^T n
$$

$$
n_i \sim (0, V (\bar{\Sigma}^{2i} \bar{\Sigma}^T) U^T C_n U (\Sigma \bar{\Sigma}^{2i}) V^T), \ n_0 \sim (0, V \Sigma^T U^T C_n U \Sigma V^T)
$$
as compared to the case for the solution in terms of $A$, e.g. with the SVD the noise enters via

$$(A^T A)^{-1} A^T n \sim (0, V(\Sigma^T \Sigma)^{-1} \Sigma^T U^T C_n U \Sigma (\Sigma^T \Sigma)^{-1} V^T)$$

We have to determine how the variance of $\sum_{i=0}^{k-1} c_i n_i$ is obtained to carefully relate the two terms. Note that for two random variables $X$ and $Y$,

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2 \text{Cov}(X, Y)$$

and hence unless $X$ and $Y$ are independent we cannot immediately use linearity. However, we observe from (67) and [http://en.wikipedia.org/wiki/Covariance_matrix](http://en.wikipedia.org/wiki/Covariance_matrix)

$$\text{Cov}(c_i n_i, c_{i-1} n_{i-1}) = \text{Cov}(c_i (A^T A) n_{i-1}, c_{i-1} n_{i-1})$$

$$= c_i (A^T A) c_{i-1} \text{Cov}(n_{i-1}, n_{i-1})$$

$$= c_i c_{i-1} (A^T A) C_{n_{i-1}}.$$ 

Thus

$$\text{Var}(c_i n_i + c_{i-1} n_{i-1}) = \text{Var}(c_i n_i) + \text{Var}(c_{i-1} n_{i-1}) + 2 \text{Cov}(c_i n_i, c_{i-1} n_{i-1})$$

$$= c_i^2 C_{n_i} + c_{i-1}^2 C_{n_{i-1}} + 2 c_i c_{i-1} (A^T A) C_{n_{i-1}}$$

$$= c_i^2 (A^T A) C_{n_{i-1}} (A^T A) + c_{i-1}^2 C_{n_{i-1}} + 2 c_i c_{i-1} (A^T A) C_{n_{i-1}} \quad (69)$$

Suppose that $C_n = \sigma^2 I$ so that we can move this outside the variance term we obtain for the direct method

$$C_{(A^T A)^{-1} A^T n} = \sigma^2 V(\Sigma^T \Sigma)^{-1}(\Sigma^T \Sigma)(\Sigma^T \Sigma)^{-1} V^T = \sigma^2 V(\Sigma^T \Sigma)^{-1} V^T,$$

while for the iterative method

$$C_{n_0} = \sigma^2 V(\Sigma^T \Sigma) V^T = \sigma^2 V(\tilde{\Sigma}^2) V^T \quad \tilde{\Sigma}^2 = \frac{1}{\sigma^2} V^T C_{n_0} V$$

$$C_{n_i} = \sigma^2 V(\tilde{\Sigma}^{2i} \Sigma^T) U^T U (\Sigma \tilde{\Sigma}^{2i}) V^T = \sigma^2 V(\tilde{\Sigma}^{2i} (\Sigma^T \Sigma) \tilde{\Sigma}^{2i}) V^T = \sigma^2 V(\tilde{\Sigma}^{2i+1}) V^T = \left(\frac{1}{\sigma^2}\right)^{2i} \sigma^2 C_{n_0}^{2i+1}$$

$$(A^T A) C_{n_{i-1}} = \frac{1}{\sigma^2} C_{n_0} \left(\frac{1}{\sigma^2}\right)^{2(i-1)} C_{n_0}^{2(i-1)+1} = \left(\frac{1}{\sigma^2}\right)^{2i-1} C_{n_0}^{2i}$$

and in (69)

$$\text{Var}(c_i n_i + c_{i-1} n_{i-1}) = \left(\frac{1}{\sigma^2}\right)^{2(i-1)} (c_i \sigma^{-2} C_{n_0} + c_{i-1})^2 C_{n_0}^{2i-1}$$

$$= \sigma^2 V\left(c_i^2 \tilde{\Sigma}^{2(2i+1)} + c_{i-1}^2 \tilde{\Sigma}^{2(2i-1)} + 2 c_i c_{i-1} \tilde{\Sigma}^{4i}\right) V^T$$

$$= \sigma^2 V \tilde{\Sigma}^{2i} \left(c_i^2 \tilde{\Sigma}^2 + 2 c_i c_{i-1} + c_{i-1}^2 \tilde{\Sigma}^{-2}\right) \tilde{\Sigma}^{2i} V^T$$

$$= \sigma^2 V \tilde{\Sigma}^{2i} \left(c_i \tilde{\Sigma} + c_{i-1} \tilde{\Sigma}^{-1}\right)^2 \tilde{\Sigma}^{2i} V^T$$

97
Clearly the mechanism by which noise enters the projected solution is more complicated, and hence it is not immediate that the parameter estimation techniques follow in exactly the same way as for the full problem. This is a topic for general research.

### 8.4.5 Extending the UPRE for the LSQR Algorithm

Consider again the UPRE algorithm for the LSQR algorithm. Specifically we form the Lanczos procedure to generate bidiagonal matrix $B$ of size $(k+1) \times k$ which satisfies
\[
AV = UB \text{ initialized with } \beta = \|b\|^2 \text{ for which } U^T b = \beta e_1,
\]
where $V$ is of size $n \times k$, $U$ of size $m \times (k+1)$, both of which are column orthonormal $V^T V = I_k$, $U^T U = I_{k+1}$ and $e_1$ is the unit vector of length $k+1$. The solution is approximated by $x = V y$, where $y(W_y)$ solves the projected problem
\[
y(W_y) = \arg\min_y \{ \| B y - \beta e_1 \|_2 + \| y \|_{W_y}^2 \}. \tag{71}
\]
We now examine how to calculate the UPRE from the solution of the projected problem. First observe that from $V^T x = V^T V y$ we have $y_{\text{exact}} = V^T x_{\text{exact}}$. In terms of the projected problem the measurable residual is now given by
\[
A x(W_y) - b = AV y(W_y) - b,
\]
and with
\[
y(W_y) = (B^T B + W_y)^{-1} B^T (\beta e_1) = (B^T B + W_y)^{-1} B^T (U^T b)
\]
we obtain
\[
A x(W_y) - b = (AV (B^T B + W_y)^{-1} B^T U^T - I_m)b = (UB (B^T U^T U B + W_y)^{-1}(UB)^T - I_m)(b_{\text{exact}} + \eta). \tag{72}
\]
Defining $B(W_y) = UB(B^T U^T U B + W_y)^{-1}(UB)^T$, we obtain for the LSQR solution, assuming that $A$ was weighted by the noise in $b$ so that the variance is 1,
\[
E(\| A x(W_y) - b \|^2) = E(\| (B(W_y) - I_m)b_{\text{exact}} \|^2) + \text{trace}((B(W_y) - I_m)^2). \tag{73}
\]
On the other hand, in terms of the projected solution the risk is given by
\[
p(W_y) = A(x(W_y) - x_{\text{exact}})
= B(W_y)(b_{\text{exact}} + \eta) - b_{\text{exact}}
= (B(W_y) - I)b_{\text{exact}} + B(W_y)\eta, \tag{74}
\]
for which
\[
E(\| p(W_y) \|^2) = E(\| (B(W_y) - I)b_{\text{exact}} \|^2) + \text{trace}(B^T(W_y)B(W_y)).
\]
Thus following the derivation for the full problem

\[
E(\|p(W_y)\|^2) = E(\|Bx(W_y) - b\|^2) - \text{trace}((B(W_y) - I)^2) + \text{trace}(B^T(W_y)B(W_y))
\]

\[
= E(\|P(W_y)b\|^2) + 2\text{trace}(B(W_y)) - m,
\]

where \(P(W_y) = I_m - B(W_y)\), which gives the UPRE functional

\[
U_{\text{global}}(W_y) = \|P(W_y)b\|^2 + 2\text{trace}(B(W_y)) - m.
\]

In contrast suppose that we assume that we start with the projected problem (71) and obtain the UPRE functional directly. For this problem

\[
U_{\text{proj}} = \|(B(B^TB + W_y)^{-1}B^T - I_{k+1})\beta e_1\|^2 + 2\text{trace}(B(B^TB + W_y)^{-1}B^T) - (k + 1)
\]

\[
= \|U^T b\|^2 - \|b\|^2 + \|P(W_y)b\|^2 + 2\text{trace}(B(B^TB + W_y)^{-1}B^T) - (k + 1)
\]

(75)

where we use \(U^TU = I_{k+1}\). Thus when calculated using the projected problem without projecting to the full problem we calculate

\[
U_{\text{proj}} = U_{\text{global}} + 2\text{trace}(B(B^TB + W_y)^{-1}B^T) - 2\text{trace}(UB(B^TB + W_y)^{-1}B^TU^T)
\]

\[
+ (m - (k + 1)) + \|U^T b\|^2 - \|b\|^2.
\]

(76)

Notice, however, by (70) that \(\|U^T b\|^2 = \|b\|^2\). We need to compare the eigenvalues of \(B(B^TB + W_y)^{-1}B^T\) with those of \(UB(B^TB + W_y)^{-1}B^TU^T\), where the second matrix is of size \(m \times m\) and the former of size \(k + 1 \times k + 1\). Now suppose there exists \(\nu\) and \(e\) such that

\[
UB(B^TB + W_y)^{-1}B^TU^T e = \nu e,
\]

then

\[
B(B^TB + W_y)^{-1}B^TU^T e = \nu U^T e
\]

so that \(\nu\) is also an eigenvalue of \(B(B^TB + W_y)^{-1}B^T\), with eigenvector \(U^T e\). On the other hand \(U\) is of maximum rank \(k + 1\), and thus \(\text{trace}(B(B^TB + W_y)^{-1}B^T) = \text{trace}(UB(B^TB + W_y)^{-1}B^TU^T)^2\). Thus when seeking to minimize the global predictive risk \(U_{\text{global}}\) with respect to solving the projected problem, it is sufficient to find the minimum of \(U_{\text{proj}}\). Note that still does not minimize the predictive risk for the full problem.

### 8.5 Resolution under Regularization by Iteration

The relationship (10) suggests the general form for the resolution matrix. Suppose that \(x^\sharp\) is an approximation to the set of equations \(Ax = b\) which is obtained as \(x^\sharp = A^\sharp b\) for some approximation \(A^\sharp\) for \(A\). The model resolution relationship is thus

\[
x^\sharp = A^\sharp b = A^\sharp A\hat{x} = R^\sharp \hat{x}, \quad R^\sharp = A^\sharp A
\]

(77)

This also follows by the cyclic property of the trace which holds providing that the matrices cycled are consistent in dimension.
which defines the approximate resolution matrix $R^\sharp$. For example suppose that $A^\sharp$ is obtained from the approximate singular value decomposition of $A$ given by

\[
A \approx U_a \Sigma_a V_a^T \\
A^\sharp = V_a \Sigma_a^{-1} U_a^T \\
R^\sharp = V_a \Sigma_a^{-1} U_a^T A,
\]

which can be evaluated. To apply the spike test requires finding the middle column of $R^\sharp$, i.e. we need to find,

\[
R^\sharp e_{\tilde{n}} = R^\sharp_{\tilde{n}}, \quad \tilde{n} := \lfloor n/2 \rfloor \tag{78}
\]

\[
R^\sharp_{\tilde{n}} \approx V_a \Sigma_a^{-1} U_a^T A_{\tilde{n}}. \tag{79}
\]

Alternatively we also note that if we assume the approximate SVD for $A$ then we also have

\[
R^\sharp \approx V_a V_a^T,
\]

under the assumption of column orthogonality for matrix $U_a$.

For the case of the LSQR algorithm we have, assuming exact arithmetic,

\[
x_k = V_k y_k, \quad B_k y_k \approx \beta e_1 \\
x_k = \beta V_k (B_k^T B_k)^{-1} B_k^T e_1 = V_k (B_k^T B_k)^{-1} B_k^T U_k^T b \\
A_k^\sharp = V_k (B_k^T B_k)^{-1} B_k^T U_k^T = V_k (B_k^T U_k^T U_k B_k)^{-1} (U_k B_k)^T \\
= V_k (B_k^T U_k^T U_k B_k)^{-1} (U_k B_k)^T = V_k ((A V_k)^T (A V_k))^{-1} (A V_k)^T \\
R_k^\sharp = V_k (V_k^T A^T A V_k)^{-1} V_k^T (A^T A) \quad \text{or} \\
R_k^\sharp = V_k (B_k^T B_k)^{-1} B_k^T U_k^T A = V_k (B_k^T B_k)^{-1} B_k^T (A^T U_k)^T \\
\approx V_k (B_k^T B_k)^{-1} B_k^T (V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T)^T = V_k (B_k^T B_k)^{-1} B_k^T (B_k V_k^T + \alpha_{k+1} e_{k+1} v_{k+1}^T) \\
= V_k V_k^T + V_k (B_k^T B_k)^{-1} (\alpha_{k+1} + \beta_{k+1} v_{k+1}^T) v_{k+1}^T,
\]

and we can find the $k^{th}$ column of $(B_k^T B_k)^{-1}$ approximately. Again if the second term is minimal we may use $V_k V_k^T$ to approximate the resolution.
9  Extending the Regularization, April 22 2014

A more general regularization term $R(x)$ may be considered to better preserve properties of the solution $x$:

$$x(\lambda) = \arg \min_x \{ \|Ax - b\|^2_W + \lambda^2 R(x) \} \quad (80)$$

Suppose $R$ is total variation of $x$ (general options are possible)

The total variation for a function $f$ defined on a discrete grid is

$$\text{TV}(x(a)) = \sum_i |x(a_i) - x(a_{i-1})| \approx \Delta \sum_i \left| \frac{dx(a_i)}{da} \right|$$

TV approximates a scaled sum of the magnitude of jumps in $x$.

$\Delta$ is a scale factor dependent on the grid size.

Two dimensional version

$$\text{TV}(x(a,b)) = \sum_{i,j} \sqrt{|x(a_i, b_j) - x(a_{i-1}, b_j)|^2 + |x(a_i, b_j) - x(a_i, b_{j-1})|^2}$$

$$\approx \Delta \sum_{i,j} \sqrt{|\partial x(a_i, b_j)/\partial a|^2 + |\partial x(a_i, b_j)/\partial b|^2}$$

$$\approx \Delta \sum_{i,j} \sqrt{|\nabla x(a_i, b_j)|^2}, \quad \nabla x = [(\partial x(a, b)/\partial a)^T, (\partial x(a, b)/\partial b)^T]^T$$

9.1 One Dimensional Case: Using CVX

Notice $\text{TV}(x(a)) = \sum_i |x(a_i) - x(a_{i-1})| = \|L_{a,1}x\|_1$ where $L_{a,1}$ is the first order derivative matrix operator.

For regularized problem we may want to solve

$$x(\lambda) = \arg \min_x \{ \|Ax - b\|^2 + \lambda^2 \|L_{a,1}x\|_1 \}$$


```matlab
m = 20; n = 10; p = 4;
A = randn(m,n); b = randn(m,1);
C = randn(p,n); d = randn(p,1); e = rand;
cvx_begin
    variable x(n)
    minimize( norm( A * x - b, 2 ) )
    subject to
        C * x == d
        norm( x, Inf ) <= e
cvx_end
```
Using CVX for 1d Problem

\[ \mathbf{x}(\lambda) = \arg \min_{\mathbf{x}} \{ \| A \mathbf{x} - \mathbf{b} \|_2^2 + \lambda \| L_{a,1} \mathbf{x} \|_1 \} \]

In this case use \( L_{a,1} \) as the second order derivative smoothing operator. Notice that this depends on \( \lambda \) still but no explicit formula is available as for the Tikhonov case.

```matlab
cvx_precision('medium');
cvx_quiet(false);
cvx_begin
    variable x(n);
r = A*x-b;
    minimize(lambda*norm(L*x,1) + r'*r);
cvx_end
```

Figure 24: A simple example of examining the solution using TV

### 9.2 Practicalities beyond 1D

For the general problem how can we find the minimum of (80) for \( R(\mathbf{x}) \) defined by a general norm. Consider first that we may write the one dimensional total variation in terms of a matrix operator and an auxiliary function.

Define \( \mathbf{l}_i^T \) to be \( i^{th} \) row of \( L_1 \)

\[ \mathbf{l}_i = [0, \ldots, 0, -1, 1, 0, \ldots, 0], \quad \mathbf{x}(a_i) - \mathbf{x}(a_{i-1}) = \mathbf{l}_i^T \mathbf{x} \]

102
where the pair \([-1, 1]\) is in the \((i - 1, i)\)th position. Then

\[
TV(x) = \sum_{i=2}^{n} \psi((I_{i-1}^T x)^2), \quad \psi(t) = \sqrt{t}, \quad \psi'(t) = \frac{1}{2\sqrt{t}}
\]

and we see that the derivative is singular at \(t = 0\).

However we know that if we wish to solve a minimization problem we need the gradient of the functional (see background in APM 505).

We consider alternative definitions for the functional \(\psi(t)\) and introduce approximations for the total variation of the data.

Suppose that we introduce a perturbation \(\beta > 0\)

\[
\psi(t) = \sqrt{t + \beta^2}, \quad \psi'(t) = \frac{1}{2\sqrt{t + \beta^2}}. \tag{81}
\]

Then the partial derivative \(\psi'(t)\) is nonsingular and we approximate the total variation through

\[
|x| \approx \sqrt{|x|^2 + \beta^2}, \quad 1 \gg \beta > 0.
\]

Another choice is the Huber function given by

\[
\psi(t) = \begin{cases} 
  t/\epsilon > 0 & t \leq \epsilon^2 \\
  2\sqrt{t} - \epsilon > 0 & t > \epsilon^2
\end{cases}
\]

\(\tag{82}\)

Notice that the choice here for \(\epsilon\) makes both \(\psi\) and \(\psi'\) continuous.

Moreover the two definitions for \(\psi\) are quite close practically.

In particular viable implementation using nonlinear minimization approximates (using the first definition for \(\psi\))

\[
TV(x(a, b)) \approx \Delta \sum_{i,j} \sqrt{|\nabla x(a_i, b_j)|^2 + \beta^2} = \sqrt{||\nabla x||^2 + \beta^2}
\]

i.e. \(\nabla x \approx [(L_{a,1} x)^T, (L_{b,1} x)^T]^T\) where \(L_{a,1}, L_{b,1}\) denote first order derivatives in \(a\) and \(b\).

9.2.1 Review: Newton’s Method April 22, 2014

We minimize the local quadratic approximation in the neighborhood of the minimum of a function \(F(x)\) with respect to a search direction \(\delta\): 

\[
G(\delta) \approx F(x + \delta) \approx F(x) + (\nabla F)^T \delta + \frac{1}{2} \delta^T H \delta,
\]

where \(H\) is the Hessian of function \(F\).
For this approximation $\nabla G(\delta) = \nabla F(x) + H\delta$, which is zero when $H\delta = -\nabla F(x)$. This determines an alternative search direction to the steepest descent. The convergence rate is typically quadratic, but requires that the initial step is sufficiently close to the solution. A line search is not required as with steepest descent, because the quadratic model should provide an appropriate direction and step length along that direction, but away from the solution it can be useful to use line search.

**Newton’s Method:** For given $x_k$ solve $H(x_k)\delta_k = -\nabla F(x_k)$, and take

$$x_{k+1} = x_k + \alpha_k\delta_k, \quad \alpha_k = 1.$$  

**Damped Newton’s Method:** For given $x_k$ solve $H(x_k)\delta_k = -\nabla F(x_k)$, and take

$$x_{k+1} = x_k + \alpha_k\delta_k, \quad \text{where } \alpha_k \text{ is chosen to minimize } G(\alpha_k) = F(x_{k+1}).$$

Note here this is now a scalar minimization for $\alpha$.

**Secant-Updating- BFGS (Broyden, Fletcher, Goldstein, Shanno)**

There are many proposed secant updating formulae for the Hessian, which preserve symmetry and PD, and guarantee that the resulting quasi-Newton step yields a descent step. In designing a scheme, it is important to not only preserve symmetry and PD, but also control error aspects of the calculation. Mostly the methods are initiated with $B_0 = I$ so that the initial step uses steepest descent. Let $y_k = \nabla F(x_{k+1}) - \nabla F(x_k)$, then the BFGS update is given by

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T \delta_k} - \frac{B_k \delta_k \delta_k^T B_k}{\delta_k^T B_k \delta_k} \approx H_{k+1}.$$

We obtain the algorithm:

**Algorithm: BFGS to minimize $F(x)$.

Given expression for $F(x)$, and an initial estimate $B_0(x)$:**

For $k = 0, 1, 2, \ldots$

- Solve $B_k \delta_k = -\nabla F(x_k)$ for $\delta_k$ !Search Direction
- $x_{k+1} = x_k + \delta_k$ !Update solution
- $y_k = \nabla F(x_{k+1}) - \nabla F(x_k)$
- $B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T \delta_k} - \frac{B_k \delta_k \delta_k^T B_k}{\delta_k^T B_k \delta_k}$ !Update approximate Hessian

**9.2.2 TV Regularization by Newton’s method (Vogel Chapter 8)**

For $R(x) = TV(x)$ in a nonlinear minimization we need the gradient (and Hessian) of $R(x)$. 

104
Differentiating
\[ \nabla R(x) = L^T \text{diag}(\psi'(x))Lx = \Psi(x)x, \quad \Psi(x) = L^T \text{diag}(\psi'(x))L \]
diag(\psi'(x)) := D_1 denotes diagonal matrix with \(i^{th}\) entry \(\psi'((1^T_i x)^2)\), and the entries are positive for (81) -(82).

\( L \) is SPD provided \(\psi'(t) > 0\) whenever \(t > 0\).

Differentiating again
\[ \nabla^2 R(x) = \Psi(x) + \Psi'(x)x, \quad \Psi'(x)x = L^T \text{diag}(2(Lx)^2 \psi''(x))L = -L^TD_2L \]
\(D_2 = -\text{diag}(2(Lx)^2 \psi''(x))\) denotes diagonal matrix with \(i^{th}\) entry \(-2((1^T_i x)^2 \psi''((1^T_i x)^2)\)

For the functional \(J(x) = \frac{1}{2} \|Ax - b\|^2 + \lambda^2 R(x)\) we obtain
\[ \nabla J(x) = A^T(Ax - b) + \lambda^2 \Psi(x)x = A^T(Ax - b) + \lambda^2 L^TD_1Lx \]
\[ \nabla^2 J(x) = A^TA + \lambda^2 \Psi(x) + \lambda^2 \Psi'(x)x = A^TA + \lambda^2 L^TD_1L - \lambda^2 L^TD_2L \]

Notice that once again we need to solve a set of normal equations
\[
\begin{align*}
(A^TA + \lambda^2 \Psi(x) + \lambda^2 \Psi'(x)x)\delta &= -(A^T(Ax - b) + \lambda^2 \Psi(x)x) \\
(A^TA + \lambda^2 (L^T(D_1 - D_2)L)\delta &= -[A^T(r) + \lambda^2 L^TD_1Lx].
\end{align*}
\]

where \(r = Ax - b\).

For equivalent formulations for 2D see the results in Vogel page 152-153.

### 9.2.3 Iteratively Reweighted Norm (Rodriguez and Wohlberg 2007 and 2009)

Goal: replace \(l_1\) norm by quadratic \(l_2\) and iterate to avoid Newton-type approaches

Generally in 2D consider derivative approximations \(L_{1,a}x = u\) and \(L_{1,b}x = v\).

\[ R(x) = \|
\sqrt{\| \nabla x \|^2}
\|_q = \sum_l (\sqrt{u_l^2 + v_l^2})^q = \sum_l (u_l^q + v_l^q)^{q/2} \quad 0 < q \leq 2 \]

Suppose that \(W\) is a diagonal matrix with entries \(z_l = (u_l^2 + v_l^2)^{(q-2)/2}\), then
\[ \|\text{diag}(W^{1/2}, W^{1/2})[u^T, v^T]^T\|_2^2 = \sum_l (z_l(u_l^2 + v_l^2)) = \sum_l (u_l^2 + v_l^2)^{q/2}. \]

Hence
\[
R(x) = \| \left( \begin{array}{cc}
W^{1/2} & 0 \\
0 & W^{1/2}
\end{array} \right)
\left( \begin{array}{c}
L_{1,a} \\
L_{1,b}
\end{array} \right)x \|_2^2 = \|W^{1/2}Lx\|_2^2,
\]
\[ W = \text{diag}(W, W) \quad L = \text{diag}(L_{1,a}, L_{1,b}) \]

105
Replacing $W$ by $W^{(l)}$ for iteration $l$, indicating that the elements are calculated using $L_{1,0}x^{(k)} = u^{(k)}$ and $L_{1,b}x^{(k)} = v^{(k)}$, $z_l^{(k)} = ((u_l^{(k)})^2 + (v_l^{(k)})^2)^{q-2}/2$ yields

$$R^{(k)}(x) = \|Lx\|_W^{2(k)}$$

To avoid singularity again introduce scaling parameter:

$$W_l = \tau(z_l)(z_l), \quad \tau(z) = 1, \quad z \geq \epsilon, \quad \tau(z) = 0, \quad z < \epsilon.$$  

### 9.2.4 IRN Algorithm

**Input:** $\lambda, \epsilon, b, A, L$, TOL

1. Initialize: $k = 0$, $x^{(k)} = (A^T A + \lambda^2 L^T L)^{-1}A^T b$
2. while $\|x^{(k)} - x^{(k-1)}\|_2 > TOL$ do
3. Calculate $(Lx^{(k)})^T = [u, v]$, $z_l = (u_l^2 + v_l^2)^{(q-2)/2}$
4. Update the weighting $W_l = \tau(u_l^2 + v_l^2)(z_l)$
5. Update solution $x^{(k+1)} = (A^T A + \lambda^2 L^T W L)^{-1}A^T b$ (normal equations again)
6. $k = k + 1$
7. end while


### 9.3 Augmented Lagrangian Approaches: Split Bregman for TV

The main reference is here with software:


Many developments have been made since that time:


Above reference discusses relationship of SB to Augmented Lagrangian and Peaceman-Rachford alternating direction
9.3.1 SB The Main Idea of GO Paper: for Regularization $R(x)$

For $R(x) = Lx$ for $L \in \mathbb{R}^{q \times n}$: Introduce $d = Lx$

Rewrite $R(x) = \frac{\lambda}{2} \|d - Lx\|_2^2 + \mu \|d\|_1$

Anisotropic Formulation:

$$(x, d) = \arg \min_{x,d} \left\{ \frac{1}{2} \|Ax - b\|_2^2 + \frac{\lambda}{2} \|d - Lx\|_2^2 + \mu \|d\|_1 \right\}$$

Solve using an alternating minimization which separates minimization for $d$ from $x$

Derivation uses the Bregman distance and so called Bregman proximal point iteration.

Various versions of the iteration can be defined. Focus on basic formulation here. Iterate over

S1: $x^{(k+1,l)} = \arg \min_x \left\{ \frac{1}{2} \|Ax - b\|_2^2 + \frac{\lambda}{2} \|Lx - (d^{(k+1,l-1)} - g^{(k)})\|_2^2 \right\}$ \hspace{1cm} (85)

S2: $d^{(k+1,l)} = \arg \min_d \left\{ \frac{\lambda^2}{2} \|d - (Lx^{(k+1,l)} + g^{(k)})\|_2^2 + \mu \|d\|_1 \right\}$ \hspace{1cm} (86)

S3: $g^{(k+1)} = g^{(k)} + Lx^{(k+1)} - d^{(k+1)}$. \hspace{1cm} (87)

9.3.2 SB Unconstrained algorithm:

Update for $x$:

$$x = \arg \min_x \left\{ \frac{1}{2} \|Ax - b\|_2^2 + \frac{\lambda^2}{2} \|Lx - h\|_2^2 \right\}, \quad h = d - g$$

$$= \arg \min_x \left\{ \|\tilde{A}x - \tilde{b}\|_2^2 \right\}, \quad \tilde{A} = (A^T, \lambda L^T)^T, \quad \tilde{b} = (b^T, \lambda h^T)^T.$$  

Standard least squares update using a Tikhonov regularizer.

Update for $d$:

$$d = \arg \min_d \left\{ \mu \|d\|_1 + \frac{\lambda^2}{2} \|d - c\|_2^2 \right\}, \quad c = Lx + g$$

$$= \arg \min_d \left\{ \|d\|_1 + \frac{\gamma}{2} \|d - c\|_2^2 \right\}, \quad \gamma = \frac{\lambda^2}{\mu}.$$  

This is achieved using soft thresholding.
9.3.3 Thresholding for \(d\)

If \(d\) has \(q\) components \((d)\) componentwise solution:

\[(d)_i = \frac{c_i}{|c_i|} \max(|c_i| - \frac{1}{\gamma}, 0) \quad i = 1 : q\]

If \(d\) is two dimensional it contains components \(d_a\) and \(d_b\).

\(L\) is defined for two dimensions. \(d_a = L_a x\), \(d_b = L_b x\)

Threshold is applied for each component of \((d_a^T, d_b^T)^T\): we use

\[\|d\|_1 = \|d_a\|_1 + \|d_b\|_1\]

The TV norm for the two dimensional case can be written

\[\|d\|_{TV} = \left(\sum_{i=1}^{n} ||d_i||_2\right) \quad q = 2n\]

We evaluate for example \(\nabla = [\nabla_a^T, \nabla_b^T]^T\) at each point \(i\) in space, there are two components at each point.

Intrinsically TV is still local.

\[(d_{TV})_i = \frac{c_i}{\|c_i\|_2} \max(\|c_i\|_2 - \frac{1}{\gamma}, 0) \quad i = 1 : n\]

SB Algorithm

**Input:** \(\lambda, \tau, A, L, \sigma^2\)

**Output:** \(x, d\).

1: Initialize: \(k = 0, d^{(k)} = g^{(k)} = x^{(k)} = 0, b^{(0)} = b\).

2: while \(\|Ax - b\|_2 > \sigma^2\) (DISCREPANCY) do

3: while \(\|x^{(k)} - x^{(k-1)}\|_2 > \tau\) do

4: Update \(x\) : solve equation (85) where \(b = b^{(k)}\)

5: Update \(d\) : solve equation (86)

6: Update \(g\) : equation (87)

7: end while

8: Update \(b\) : equation \(b^{(k+1)} = b + b^{(k)} - Ad^{(k+1)}\)

9: end while

In the unconstrained case the outer loop is removed. No update for \(b\) are used.

The costly step is that for \(x\). Overall cost is basically repeats of quadratic Tikhonov
9.3.4 Matlab for SB in One D

function \[x_k,k\]=SB(A, L, b,\lambda,\mu,tol,\text{itk})
update_x=inline('(A\text{t}A)\text{t}b','A','A\text{t}','b');
update_d=inline('c./abs(c).*\text{max}(abs(c)-1/gamma,0)','gamma','c');
gamma=\lambda/\mu;
[n,p]=size(L);
gk=zeros(n,1);\text{dk}=gk;\text{bk}=b;\text{hk}=gk;\text{xold}=bk;\text{testx}=\text{xold};
Atilde=[\text{A};\sqrt{\text{\lambda}}\text{*L}];k=0;
while ((\text{norm(testx,2)}\text{>tol}) & (k<\text{itk})),
\text{btilde}=[b;\sqrt{\text{\lambda}}\text{*hk}];
\text{xk}=\text{update}_x\text{(Atilde,Atilde',btilde});
\text{c}=L*\text{xk}+gk;
\text{dk}=\text{update}_d\text{(gamma, c)};
\text{gk} = c - \text{dk};\text{hk}=dk-gk;\text{testx}=\text{xk-xold};\text{xold}=\text{xk};
k=k+1;
end

9.3.5 Solutions with SB

Figure 25: Solution using the Split Bregman algorithm programmed directly
9.3.6 Further Discussion on the SB

First we note that the basic step of the SB is a Tikhonov regularization update

\[ \mathbf{x}^{(k+1)} = \underset{\mathbf{x}}{\text{arg min}} \left\{ \frac{1}{2} \| \mathbf{A} \mathbf{x} - \mathbf{b} \|_2^2 + \frac{\lambda^2}{2} \| \mathbf{L} \mathbf{x} - (\mathbf{d}^{(k)} - \mathbf{g}^{(k)}) \|_2^2 \right\} \]

We can use the GSVD to examine the solution: at the first step \((\mathbf{d}^{(k)} - \mathbf{g}^{(k)}) = 0\) and so

\[ \mathbf{x}_\lambda = \sum_{i=1}^{p} \frac{\nu_i}{\nu_i^2 + \lambda^2 \mu_i^2} (\mathbf{u}_i^T \mathbf{b}) \tilde{z}_i + \sum_{i=p+1}^{n} (\mathbf{u}_i^T \mathbf{b}) \tilde{z}_i \]

\(\tilde{z}_i\) is the \(i^{th}\) column of \((\mathbf{Z}^T)^{-1}\). \((\mathbf{A} = \mathbf{U} \mathbf{\Gamma} \mathbf{Z}^T, \mathbf{L} = \mathbf{V} \mathbf{M} \mathbf{Z}^T)\)

Equivalently with filter factor \(\phi_i\)

\[ \mathbf{x}_\lambda = \sum_{i=1}^{p} \frac{\phi_i}{\rho_i} \mathbf{u}_i^T \mathbf{b} \tilde{z}_i + \sum_{i=p+1}^{n} (\mathbf{u}_i^T \mathbf{b}) \tilde{z}_i, \]

\[ \mathbf{L} \mathbf{x}_\lambda = \sum_{i=1}^{p} \frac{\phi_i}{\rho_i} \mathbf{u}_i^T \mathbf{b} \mathbf{v}_i, \quad \phi_i = \frac{\rho_i^2}{\rho_i^2 + \lambda^2}, \quad \rho_i = \frac{\nu_i}{\mu_i} \]

At subsequent steps the right hand side is changed: Let \(\mathbf{h}^{(k)} = (\mathbf{d}^{(k)} - \mathbf{g}^{(k)})\)

Update for \(\mathbf{x}\): using the GSVD : basis \(\mathbf{Z}\)

\[ \mathbf{x}^{(k+1)} = \sum_{i=1}^{p} \left( \frac{\nu_i \mathbf{u}_i^T \mathbf{b}}{\nu_i^2 + \lambda^2 \mu_i^2} + \frac{\lambda^2 \mu_i \mathbf{v}_i^T \mathbf{h}^{(k)}}{\nu_i^2 + \lambda^2 \mu_i^2} \right) \mathbf{z}_i + \sum_{i=p+1}^{n} (\mathbf{u}_i^T \mathbf{b}) \mathbf{z}_i \]

A weighted combination of basis vectors \(\mathbf{z}_i\): weights

\[ (i) \frac{\nu_i^2}{\nu_i^2 + \lambda^2 \mu_i^2} \frac{\mathbf{u}_i^T \mathbf{b}}{\nu_i} \quad (ii) \frac{\lambda^2 \mu_i^2}{\nu_i^2 + \lambda^2 \mu_i^2} \frac{\mathbf{v}_i^T \mathbf{h}^{(k)}}{\mu_i} \]

Notice (i) is fixed by \(\mathbf{b}\), but (ii) depends on the updates \(\mathbf{h}^{(k)}\)

i.e. (i) is iteration independent

If (i) dominates (ii) solution will converge slowly or not at all
\[ \lambda \text{ impacts the solution and must not over damp (ii)} \]

\[
x^{(k+1)} = \sum_{i=1}^{p} \left( \phi_i \frac{u_i^T b}{\nu_i} + (1 - \phi_i) \frac{v_i^T h^{(k)}}{\mu_i} \right) z_i + \sum_{i=p+1}^{n} (u_i^T b) z_i
\]

\[
L x^{(k+1)} = \sum_{i=1}^{p} \left( \phi_i \frac{u_i^T b}{\rho_i} + (1 - \phi_i) v_i^T h^{(k)} \right) v_i.
\]

The stability depends on coefficients

\[
\frac{\phi_i}{\rho_i} \quad 1 - \phi_i \quad \frac{\phi_i}{\nu_i} \quad 1 - \phi_i \quad \frac{1}{\mu_i}
\]

with respect to the inner products

\[
u_i^T b \quad v_i^T h^{(k)}
\]

The literature indicates that the choice for \( \lambda \) is problem dependent, and can be picked to be constant for a given application.

Let's look at what happens as we iterate to see whether it makes sense to keep \( \lambda \) constant.

We can obtain results on determination of optimal choices for the parameter \( \lambda \). For example we can use the UPRE to obtain estimates of the optimal choice for the parameter.

**Lemma 3.** Suppose noise in \( h^{(k)} \) is stochastic and both data fit \( Ax \approx b \) and derivative data fit \( Lx \approx h \) are weighted by their inverse covariance matrices for normally distributed noise in \( b \) and \( h \); then optimal choice for \( \lambda \) at all steps is \( \lambda = 1 \).

**Remark 4.** Can we expect \( h^{(k)} \) is stochastic?

**Lemma 4.** Suppose \( h^{(k+1)} \) is regarded as deterministic, then UPRE applied to find the the optimal choice for \( \lambda \) at each step leads to a different optimum at each step, namely it depends on \( h \).

**Remark 5.** Because \( h \) changes each step the optimal choice for \( \lambda \) using UPRE will change with each iteration, \( \lambda \) varies over all steps.
Figure 26: Coefficients of $x^{(k)}$ and $Lx^{(k)}$ on left and $|u_i^Tb|$ and $|v_i^Tv_i^T|$, on right, for $\lambda = .001$, .1 and 10, at iterations $k = 1$, 10 and Final.
Figure 27: Solutions: $\lambda = .001$, .1 and 10
Noise level 0.0015536

Data 4.804          LS 19.664

SB UPRE $\lambda$  20.2618          SB UPRE $\lambda=1$  20.3218

Noise level 0.0062143

Data 4.8021          LS 13.806

SB UPRE $\lambda$  15.9083          SB UPRE $\lambda=1$  15.4281