AMSC 600/CMSC 760
Fall 2007
Numerical Solution of Ill-Posed Problems
Part 1

Dianne P. O'Leary

Numerical Solution of Ill-Posed Problems

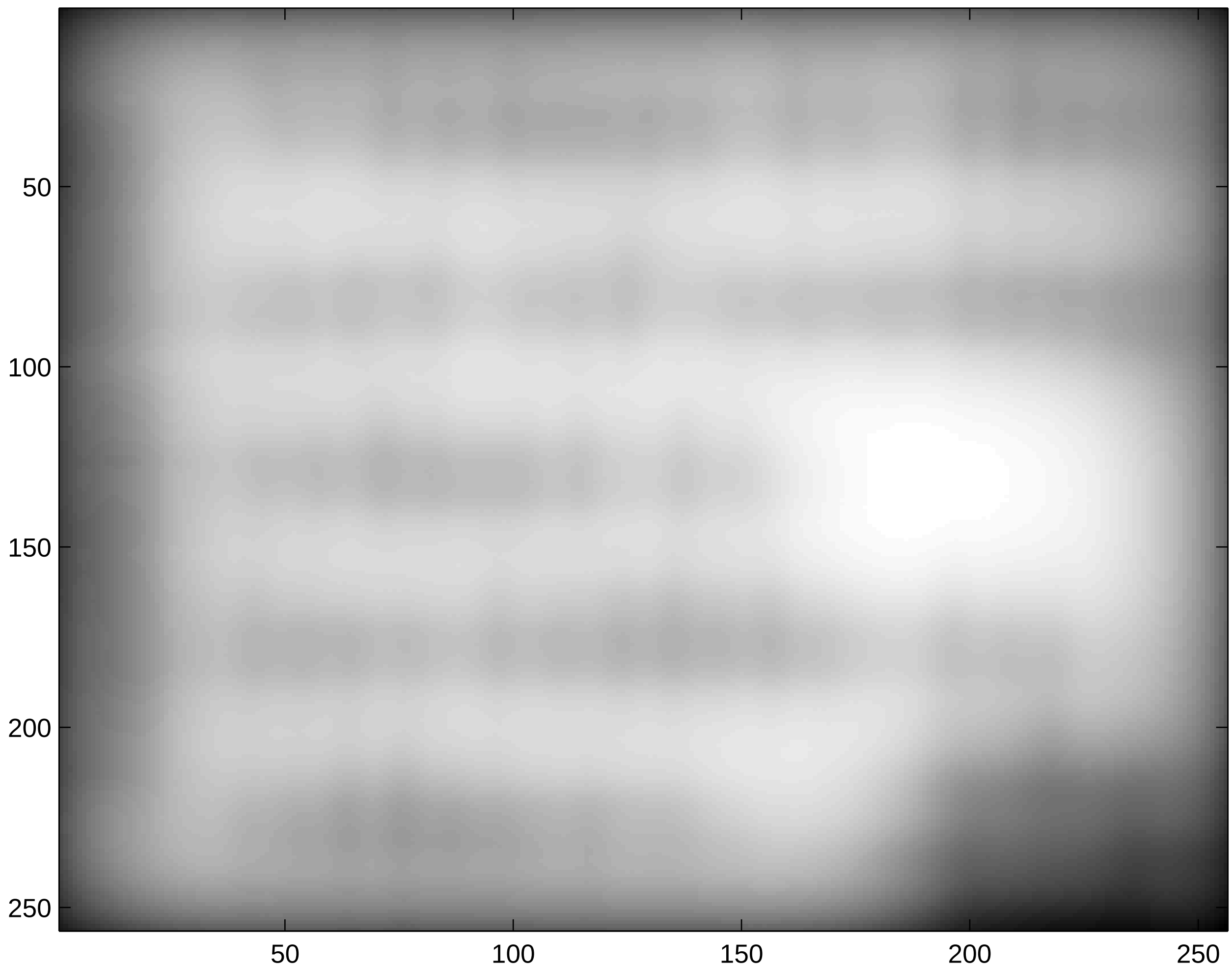
Inverse problems are among the most challenging computations in science and engineering.

They involve determining the parameters of a system that is only observed indirectly.

Examples:

- Given data from a mass spectrometer, determine the chemical species that produced it, as well as their relative proportions.
- Given sonar measurements of a containment tank, decide whether it has a hidden crack.
- Given a blurred image, reconstruct the original.

Can you deblur this image?



The plan

- The mathematical origins of the problem
- From ill-posed to ill-conditioned
- Method 1: Tikhonov Regularization
- Efficient algorithms for solving the Tikhonov problem
- Method 2: Truncated SVD
- Extending these methods to very large problems: Kronecker Product Structure

Reference: James G. Nagy and Dianne P. O'Leary,
"Image Deblurring: I Can See Clearly Now,"
Computing in Science and Engineering.

Project: Vol. 5, No. 3, May/June 2003, pp. 82-85.
Solution: Vol. 5, No. 4, July/August 2003, pp. 72-74.

The mathematical origins of the problem

Let f be the true image. Then f is actually a function over some 2-dimensional domain that we call Ω. The function values are the intensities of the image at each coordinate (s_1, s_2) in the domain.

Let g be the recorded image. Again, g is actually a function over the 2-dimensional domain, but we only have a few samples of this function, perhaps an $n_r \times n_c$ array of pixel values which we may assume are measured at points $\mathbf{s}_{jk} = (j/n_r, k/n_c)$ for $j = 1, \ldots, n_r$, $k = 1, \ldots, n_c$.

Kernels and convolutions

The recorded image g is the result of the convolution of the true image f with a recording device specified by a kernel function k so that

$$g(\mathbf{s}) = \int_\Omega k(\mathbf{s}, \mathbf{t}) f(\mathbf{t}) d\mathbf{t}.$$

If $k(\mathbf{s}, \mathbf{t}) = \delta(\|\mathbf{s} - \mathbf{t}\|)$ where δ is the Dirac δ function, then $g(\mathbf{s}) = f(\mathbf{s})$; this is the ideal case, and k is nonzero at only one point.

In practical situations, k is not this nice, although it often has small support, so that $k(\mathbf{s}, \mathbf{t})$ is zero when $\mathbf{t}$ and $\mathbf{s}$ are not close to each other.

In this case, the value of the integral is a weighted average of values of f in a neighborhood of $\mathbf{s}$.

Discretize

We obtain the matrix equation

$$\mathbf{b} = \mathbf{A}\mathbf{x}$$

by discretizing the integral. The row of this equation corresponding to s_{jk} approximates the relation

$$g(\mathbf{s}_{jk}) = \int_\Omega k(\mathbf{s}_{jk}, \mathbf{t}) f(\mathbf{t}) d\mathbf{t} \approx \sum_{\ell=1}^{n_r} \sum_{p=1}^{n_c} w_{\ell p} k(\mathbf{s}_{jk}, \mathbf{t}_{\ell p}) f(\mathbf{t}_{\ell p}),$$

where the values $w_{\ell p}$ are chosen to make the approximation as accurate as desired.

Example: Choosing $w_{\ell p} = 1/(n_r n_c)$ for all values of ℓ and p gives a rectangle rule for integration.

If we use our sample values of $\mathbf{s}$ as sample values for $\mathbf{t}$, then the entry in the row of $\mathbf{A}$ corresponding to $\mathbf{s}_{jk}$ and the column corresponding to $\mathbf{s}_{\ell p}$ is $w_{\ell p} k(\mathbf{s}_{jk}, \mathbf{s}_{\ell p})$, and this defines our matrix problem.

A serious limitation: determining k

Usually it is either

- modeled by some mathematical function. For example, for the Hubble space telescope, a mathematical function was used to model the incorrect grinding of the lenses.
- measured. For example, we aim the camera at a point source – a picture that is black except for a single white pixel – and the blurred image defines k at that pixel. By moving that white pixel and repeating the measurement – or by assuming that the image is unchanged except for translation as we move the white pixel – we can approximately determine all of the values $k(\mathbf{s}_{jk}, \mathbf{s}_{\ell p})$.

In either case, there is error in the matrix, but for now we assume it is negligible compared with error in the right-hand side.

From ill-posed to ill-conditioned

Consider a linear system of equations

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

where $\mathbf{A}$ is an $n \times n$ matrix, and $\mathbf{x}$ and $\mathbf{b}$ are vectors.

Suppose $\mathbf{A}$ is scaled so that its largest singular value is $\sigma_1 = 1$.

If the smallest singular value is $\sigma_n \approx 0$, then $\mathbf{A}$ is ill-conditioned. We distinguish two types of ill-conditioning:

- The matrix $\mathbf{A}$ is considered numerically rank deficient if there is a j such that
$$\sigma_j \gg \sigma_{j+1} \approx \cdots \approx \sigma_n \approx 0\,.$$
That is, there is an obvious gap between large and small singular values.
- If the singular values decay to zero with no particular gap in the spectrum, then we say the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is a discrete ill-posed problem.

The effects of noise

It is very difficult to compute accurate approximate solutions of discrete ill-posed problems, especially because in most real applications, the right-hand side vector $\mathbf{b}$ is not known exactly. Rather, it is more typical that the collected data has the form:

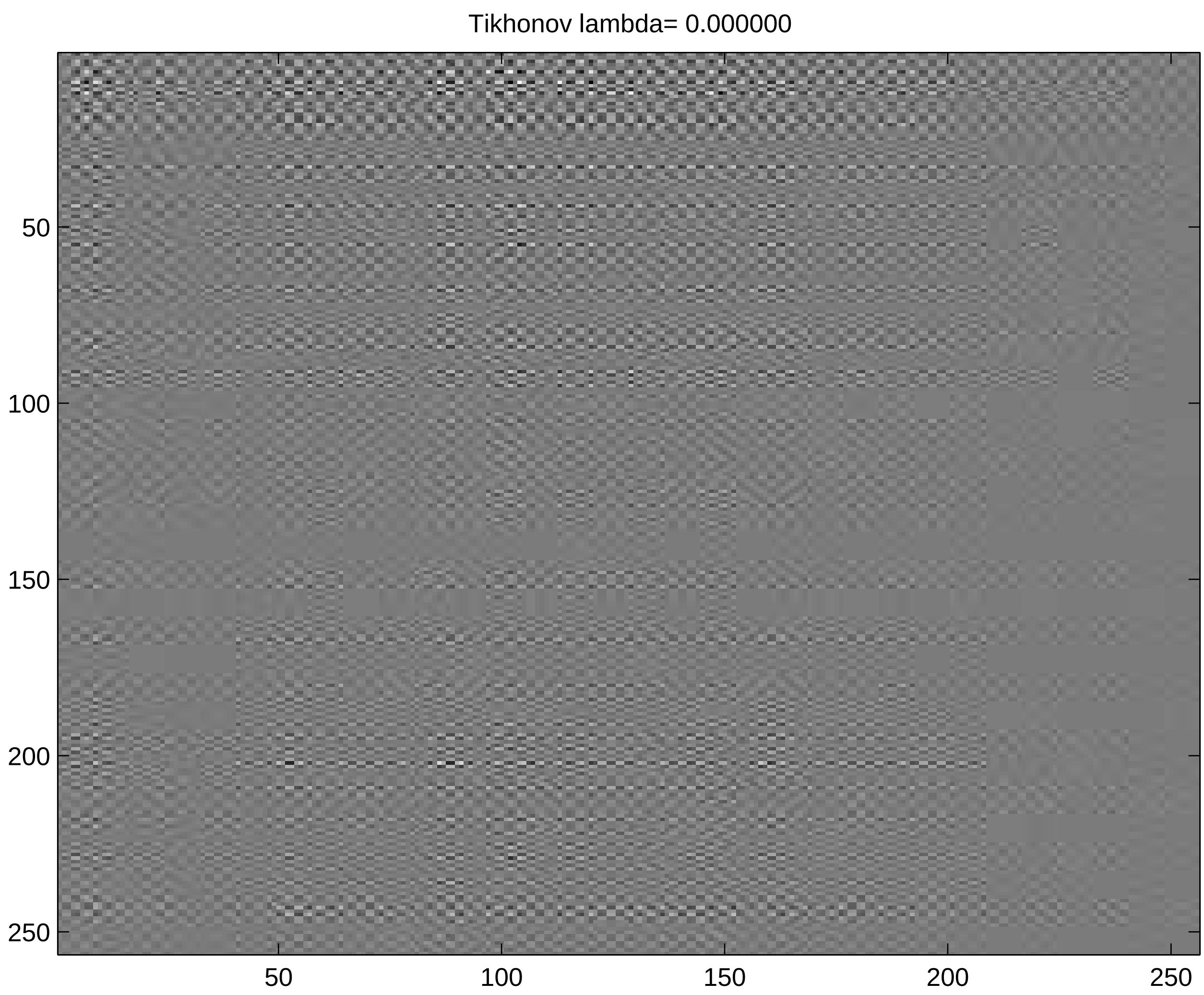
$$\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta},$$

where $\boldsymbol{\eta}$ is a vector representing (unknown) noise or measurement errors.

The goal, then, is: Given an ill-conditioned matrix $\mathbf{A}$ and a vector $\mathbf{b}$, compute an approximation of the unknown vector $\mathbf{x}$.

Naïvely solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ usually does not work, since the matrix $\mathbf{A}$ is so ill-conditioned.

Result of computing $\mathbf{A}^{-1}\mathbf{b}$ for our blurred image



Diagnosis and cure of the sick matrix $\mathbf{A}$

- The "x-ray machine" that shows us the defects in $\mathbf{A}$ is the singular value decomposition
$$\mathbf{A} = \mathbf{U\Sigma V}^T.$$
- "Surgery" on the matrix is performed using regularization.

Method 1: Tikhonov Regularization

The best known regularization procedure, called Tikhonov regularization, computes a solution of the damped least squares problem:

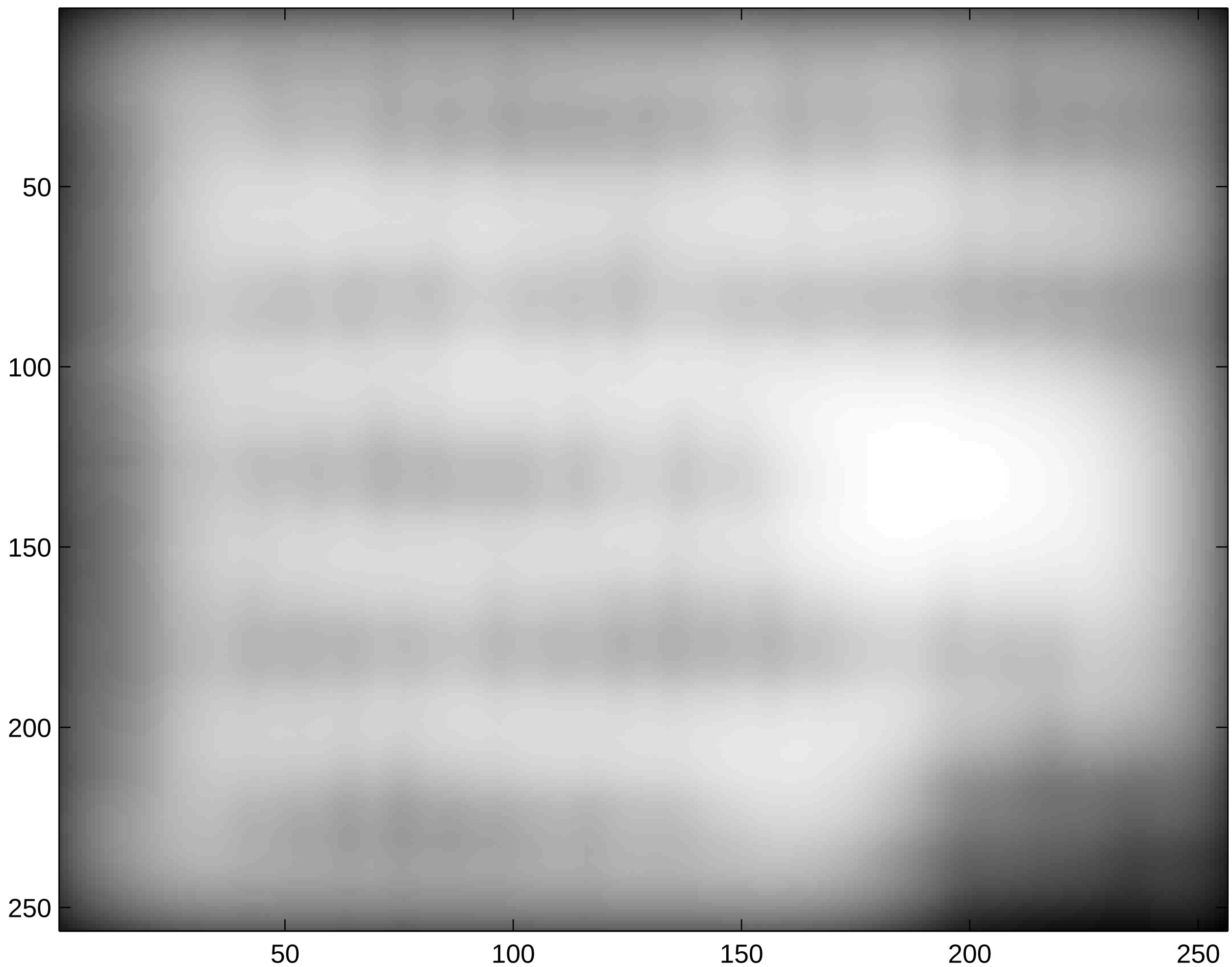
$$\min_{\mathbf{x}}\{||\mathbf{b} - \mathbf{Ax}||_2^2 + \alpha^2||\mathbf{x}||_2^2\}$$

- $\alpha^2||\mathbf{x}||_2^2$ imposes a penalty for making the norm of the solution too big, and this means that the effects of small singular values are reduced.
- The regularization parameter α controls the degree of smoothness of the solution:
  - $\alpha = 0$ implies no regularization, and we just solve the linear system of equations, getting a noisy solution.
  - If α is large, then the computed solution cannot be a good approximation of the exact $\mathbf{x}$.
- It is difficult to choose an appropriate value for α.

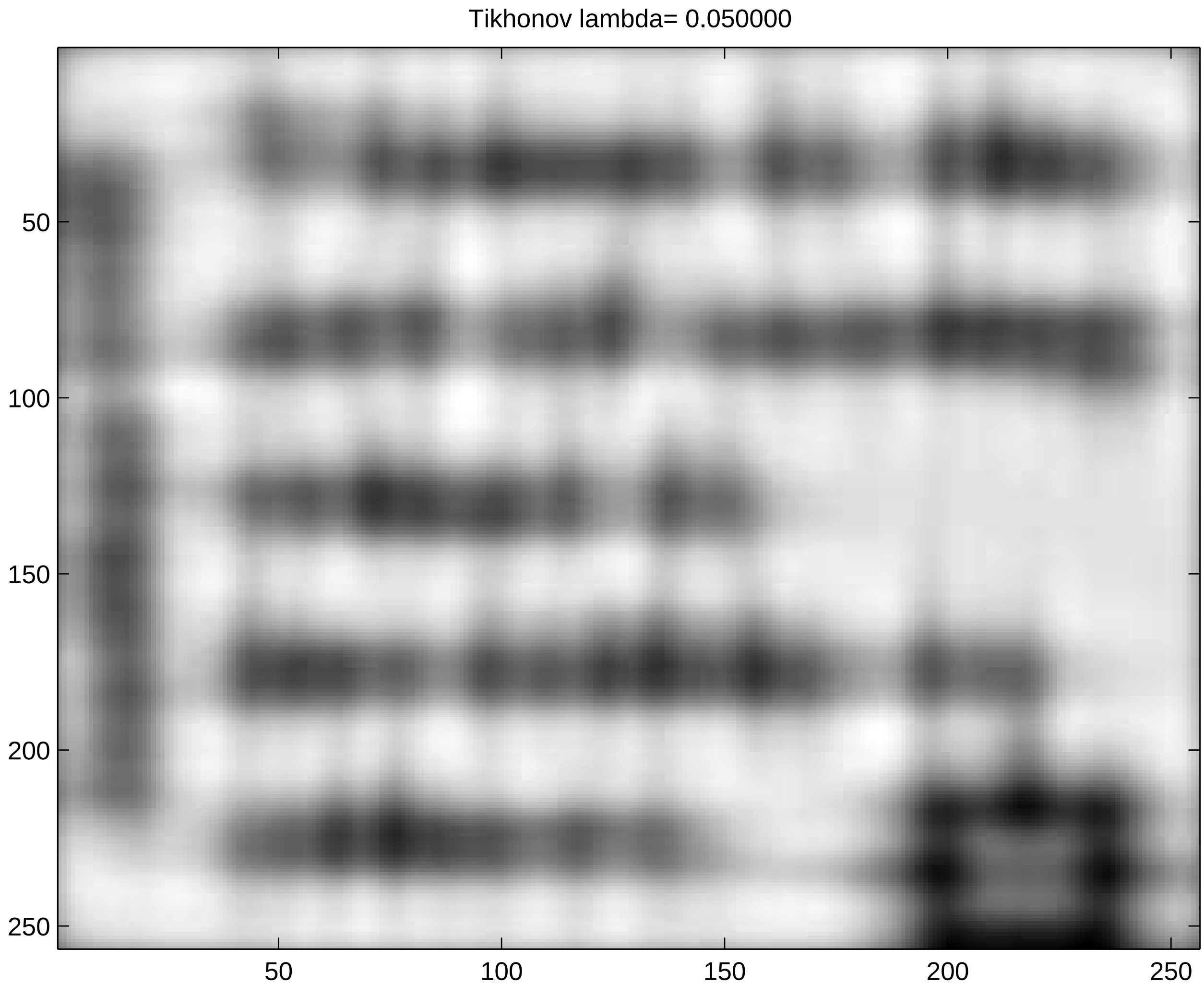
Unquiz: Show that Tikhonov regularization is equivalent to the linear least squares problem
$$\min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{A} \\ \alpha\mathbf{I} \end{bmatrix} \mathbf{x} \right\|_2^2 .$$

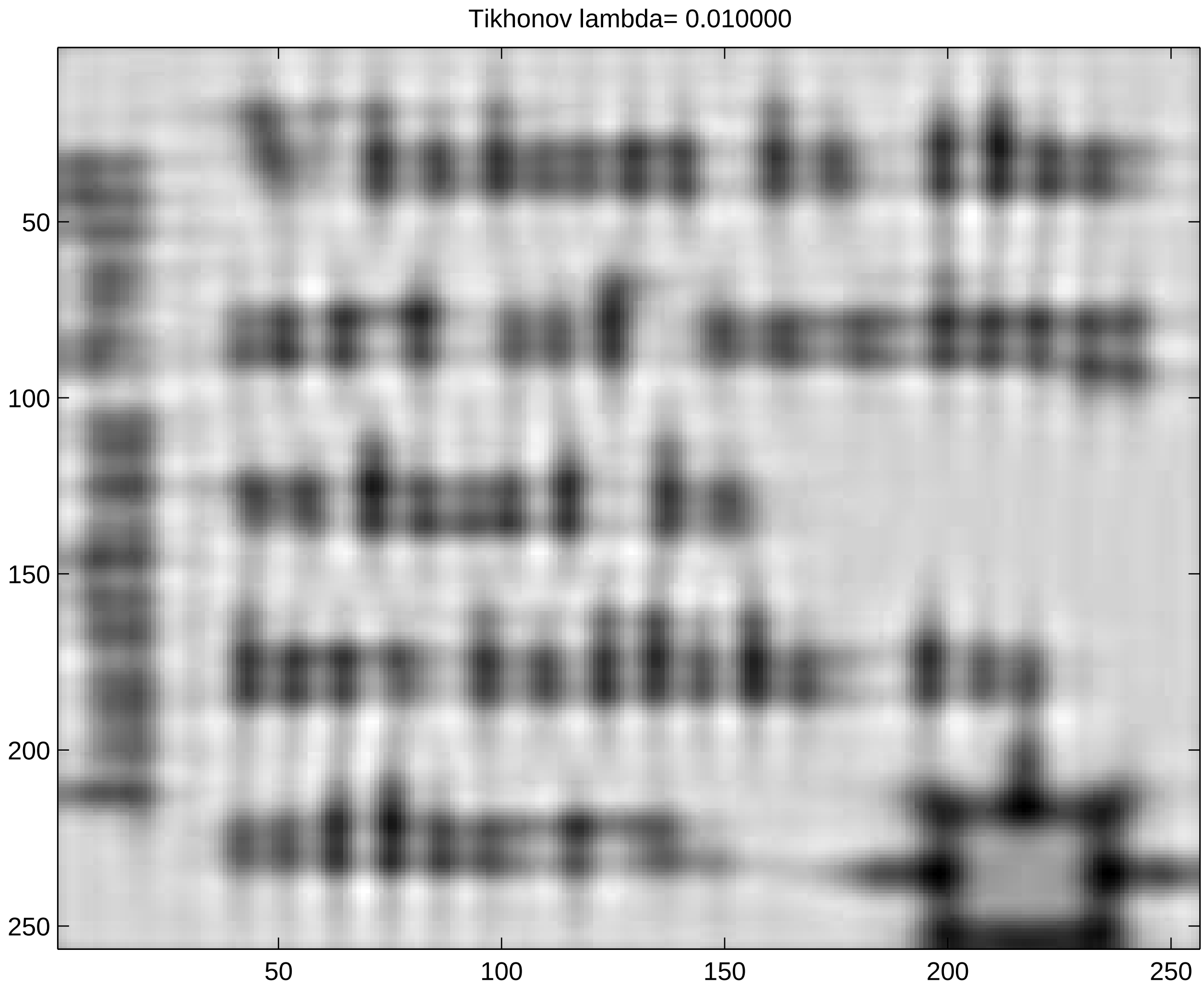
Result of Tikhonov regularization

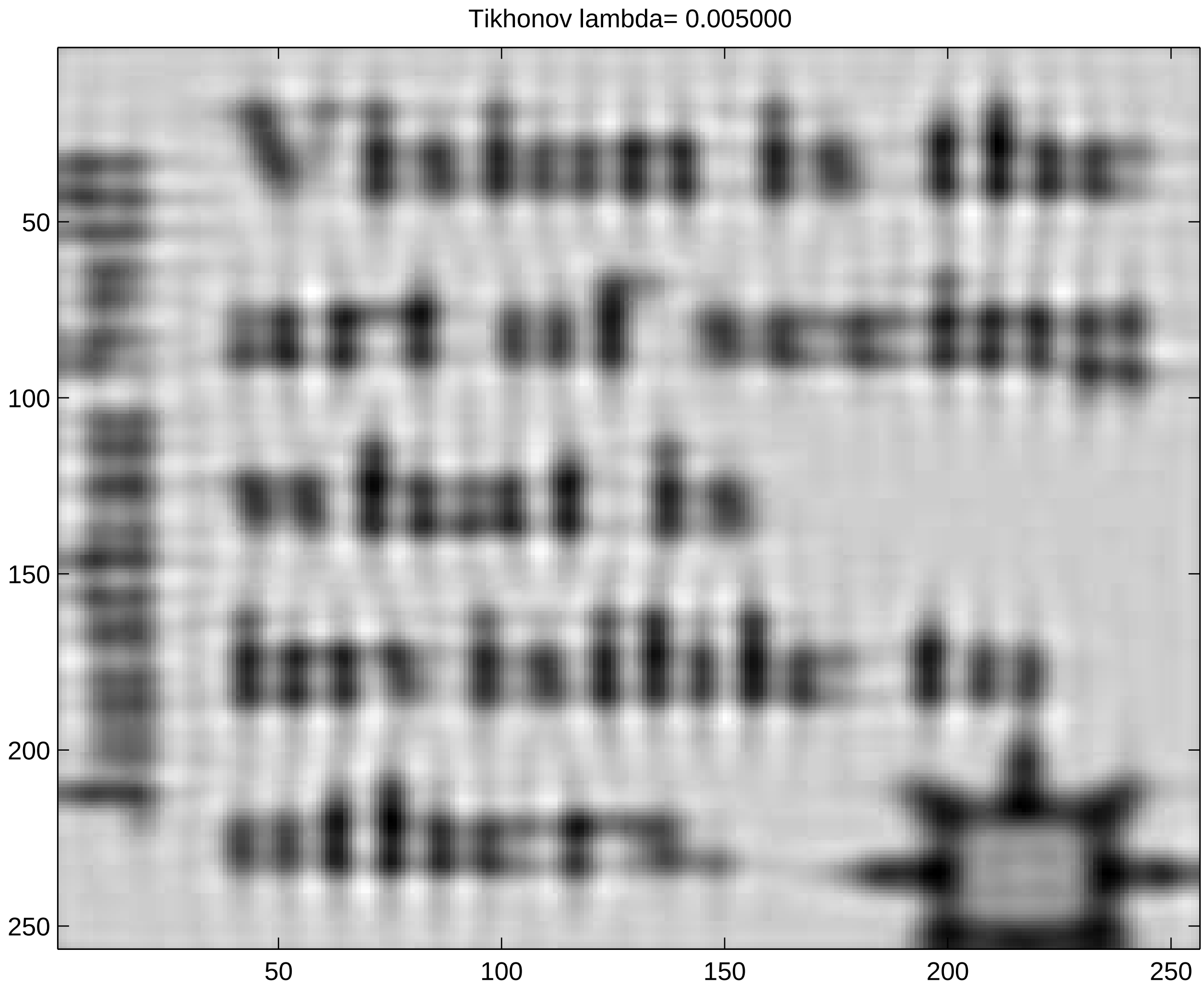
An example: Can we deblur this image?

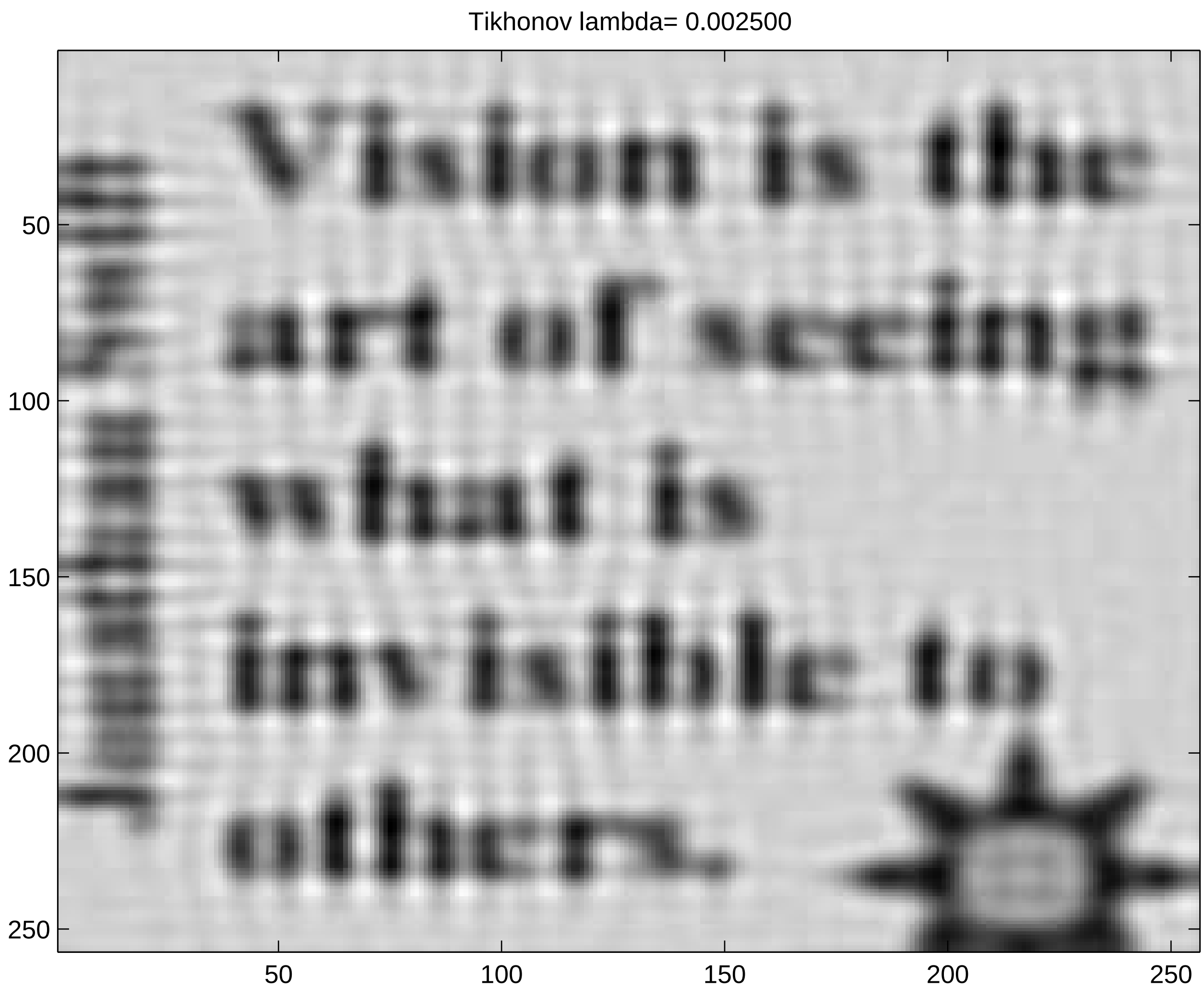


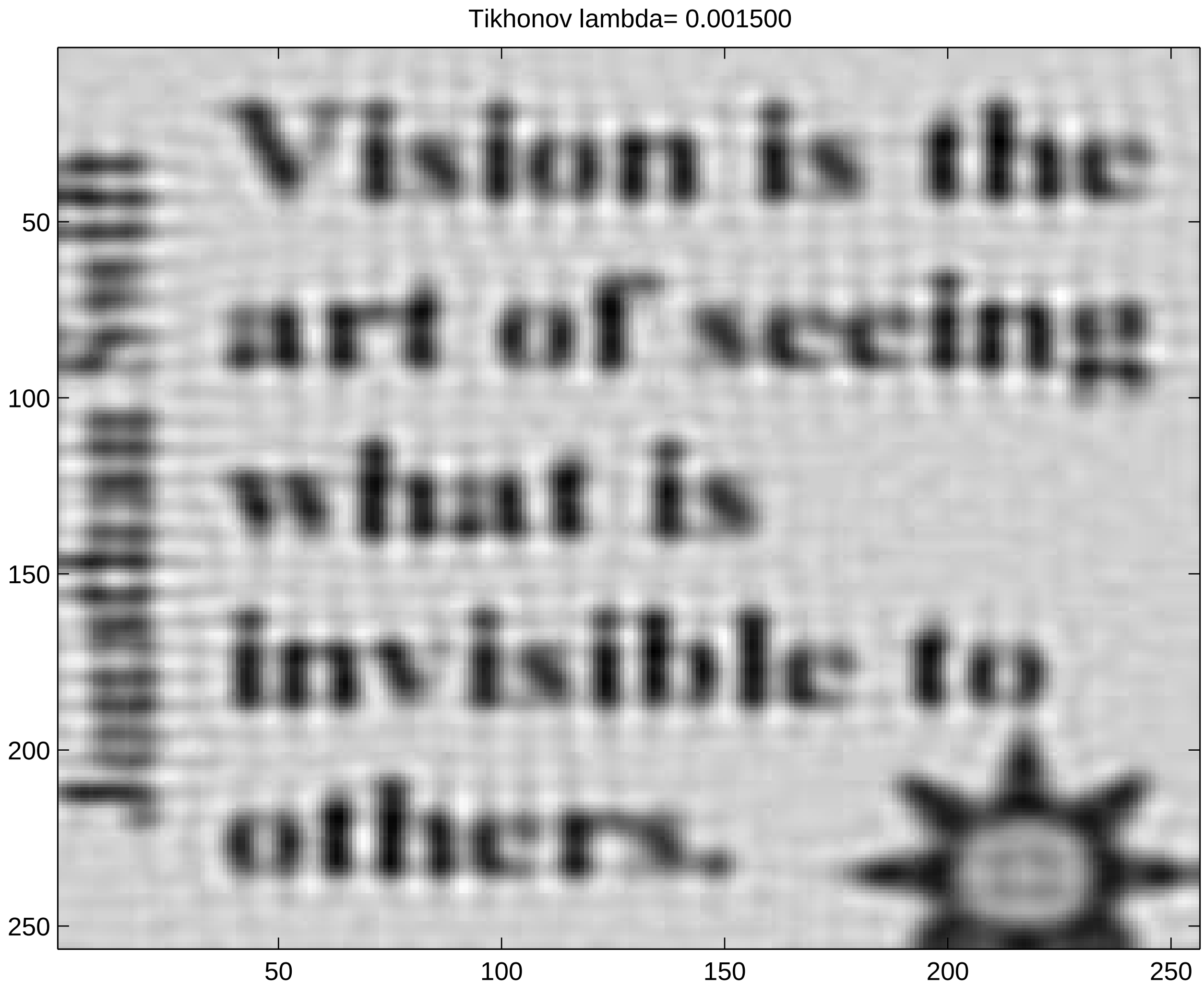
50
100
150
200
250
50
100
150
200
250

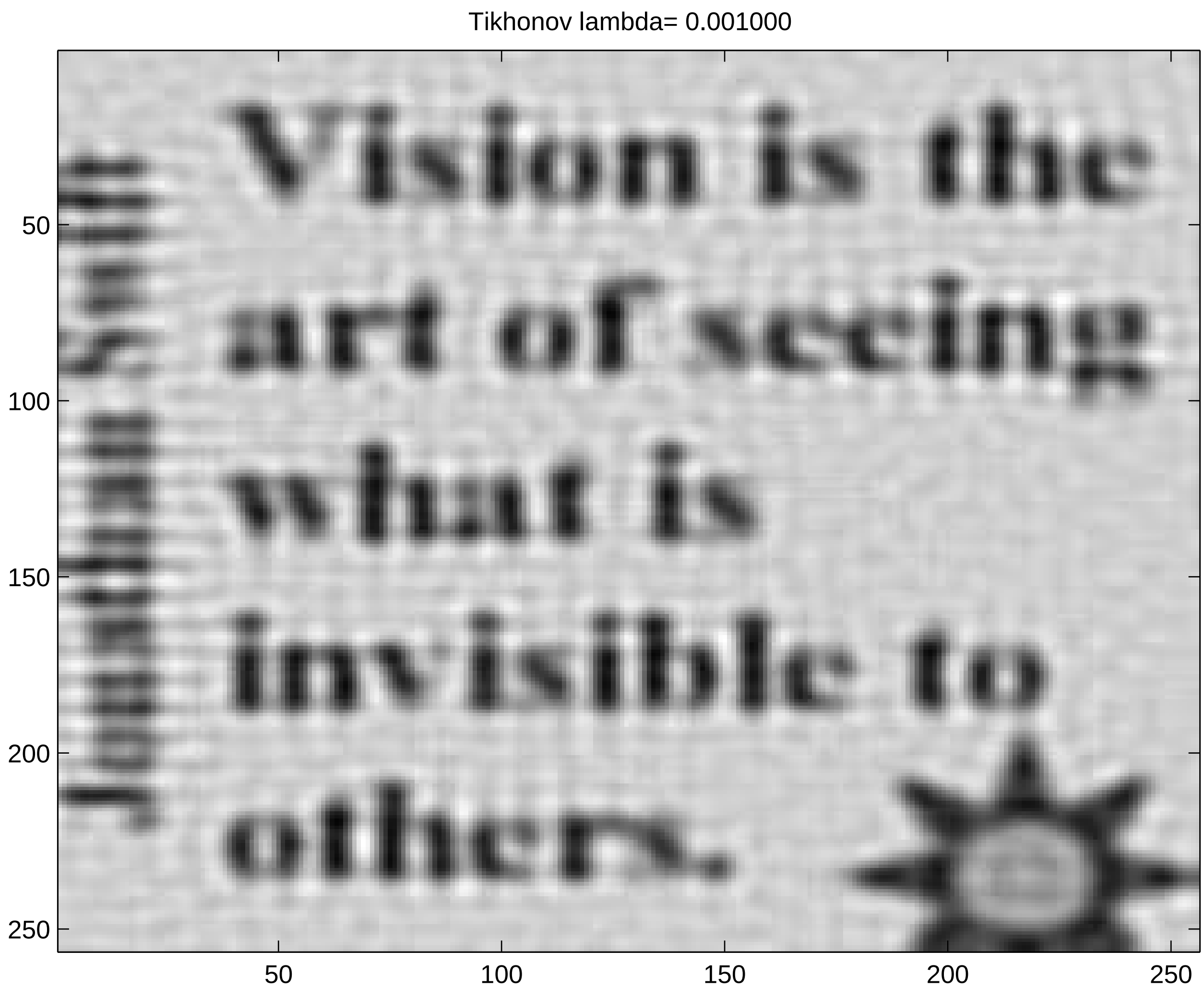

Tikhonov lambda= 0.050000
50
100
150
200
250
50
100
150
200
250

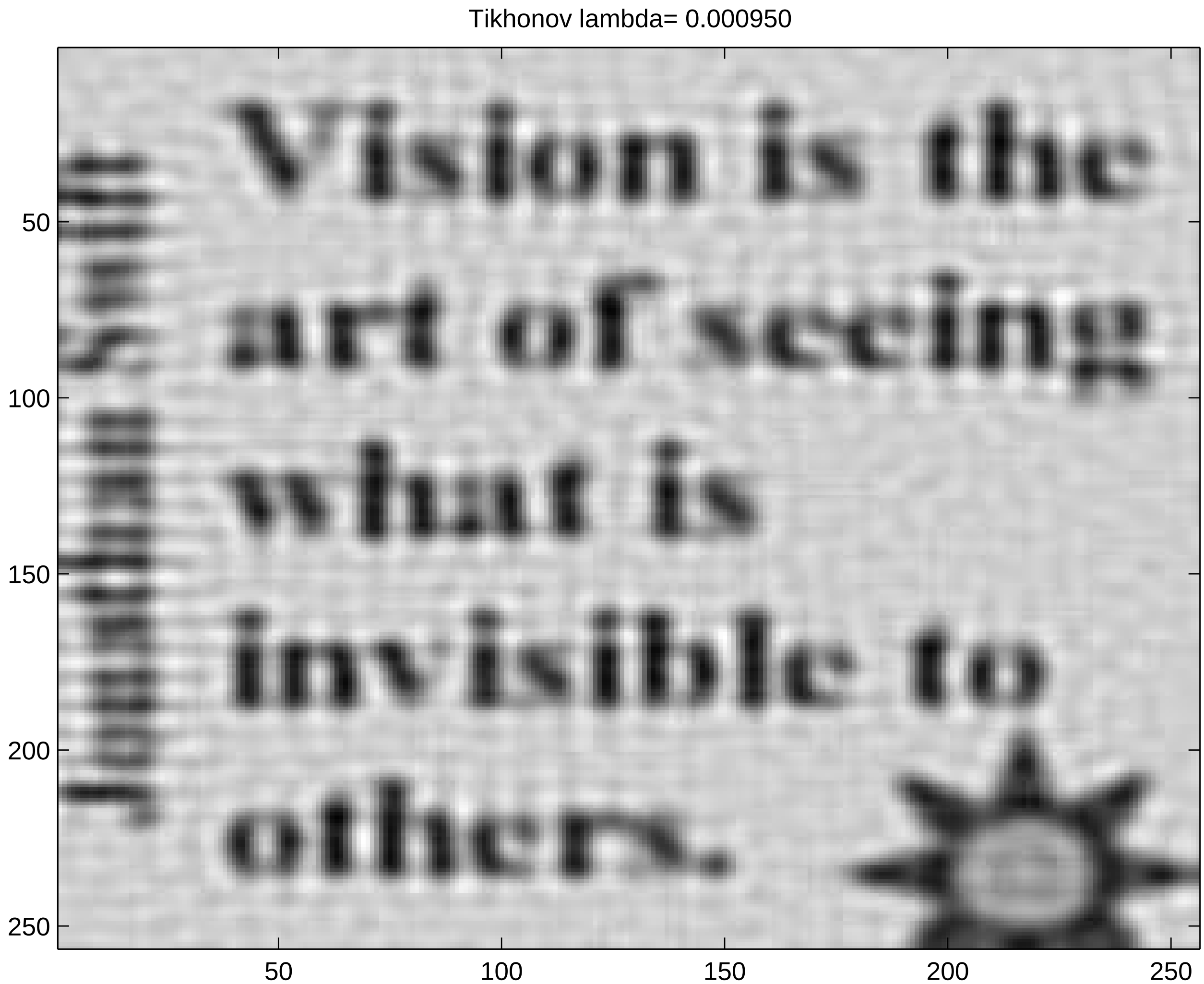

Tikhonov lambda= 0.010000
50
100
150
200
250
50
100
150
200
250

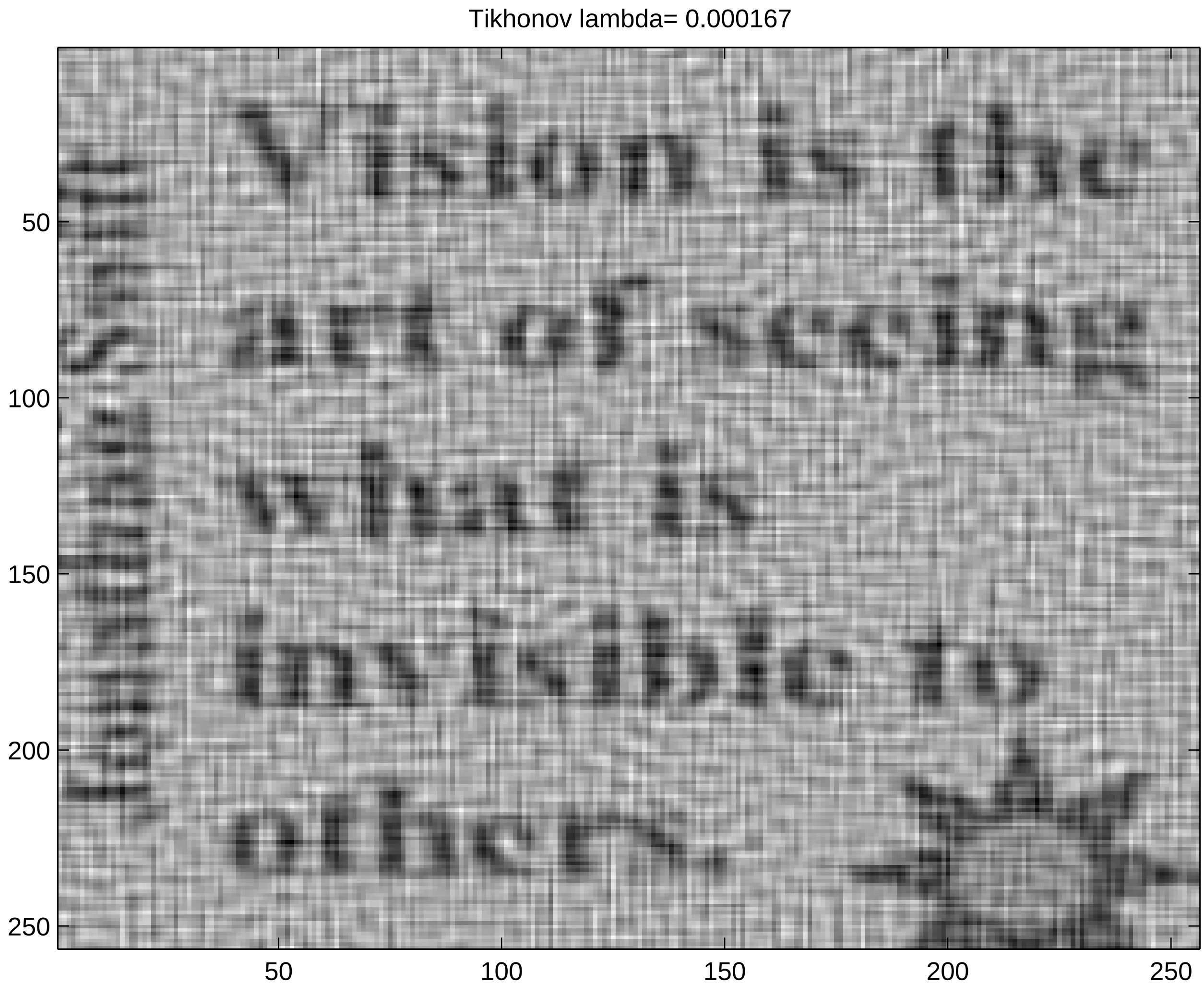

Tikhonov lambda= 0.005000
50
100
150
200
250
50
100
150
200
250

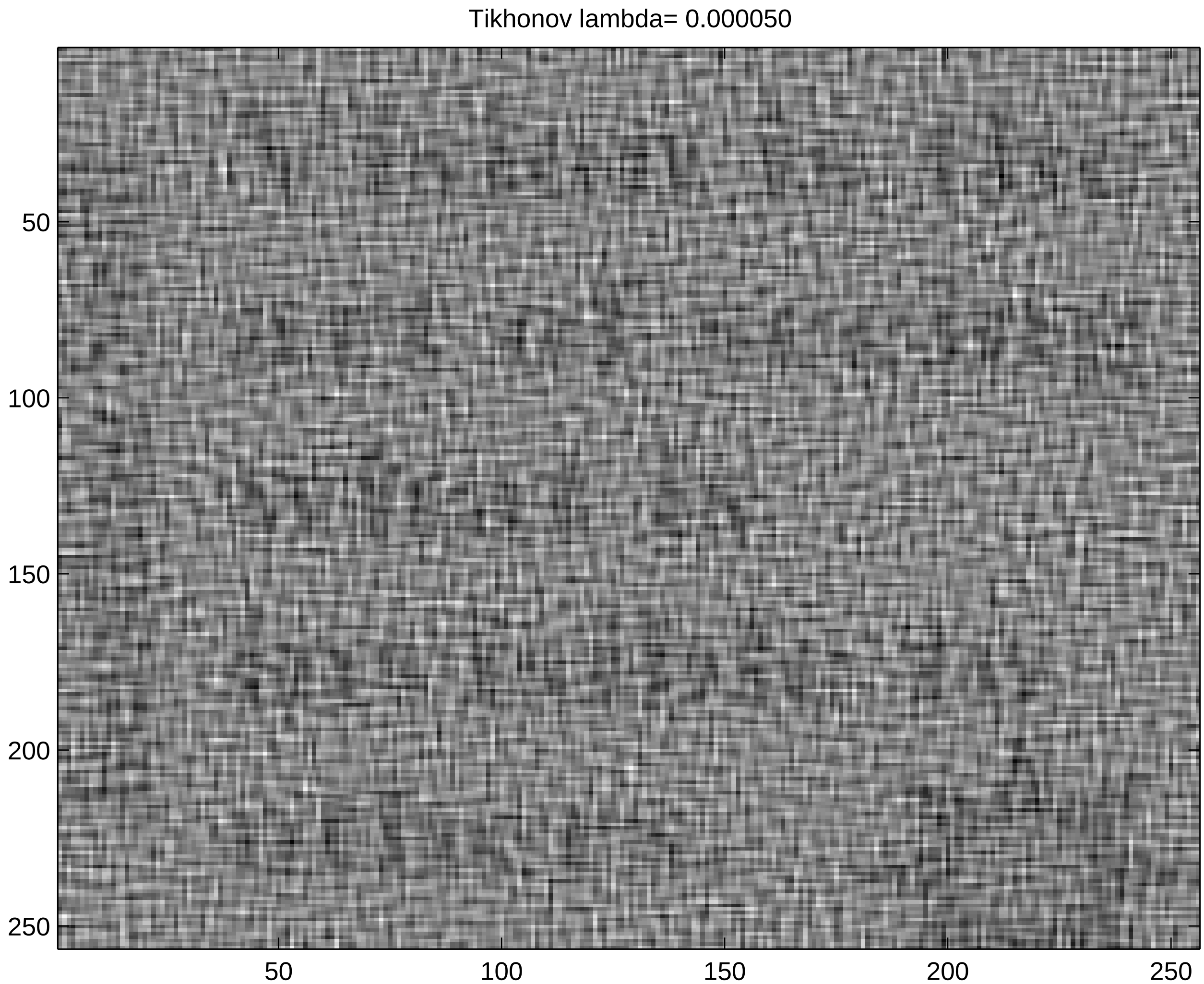

Tikhonov lambda= 0.002500
Vision is the
art of seeing
what is
invisible to
others.
50
100
150
200
250
50
100
150
200
250

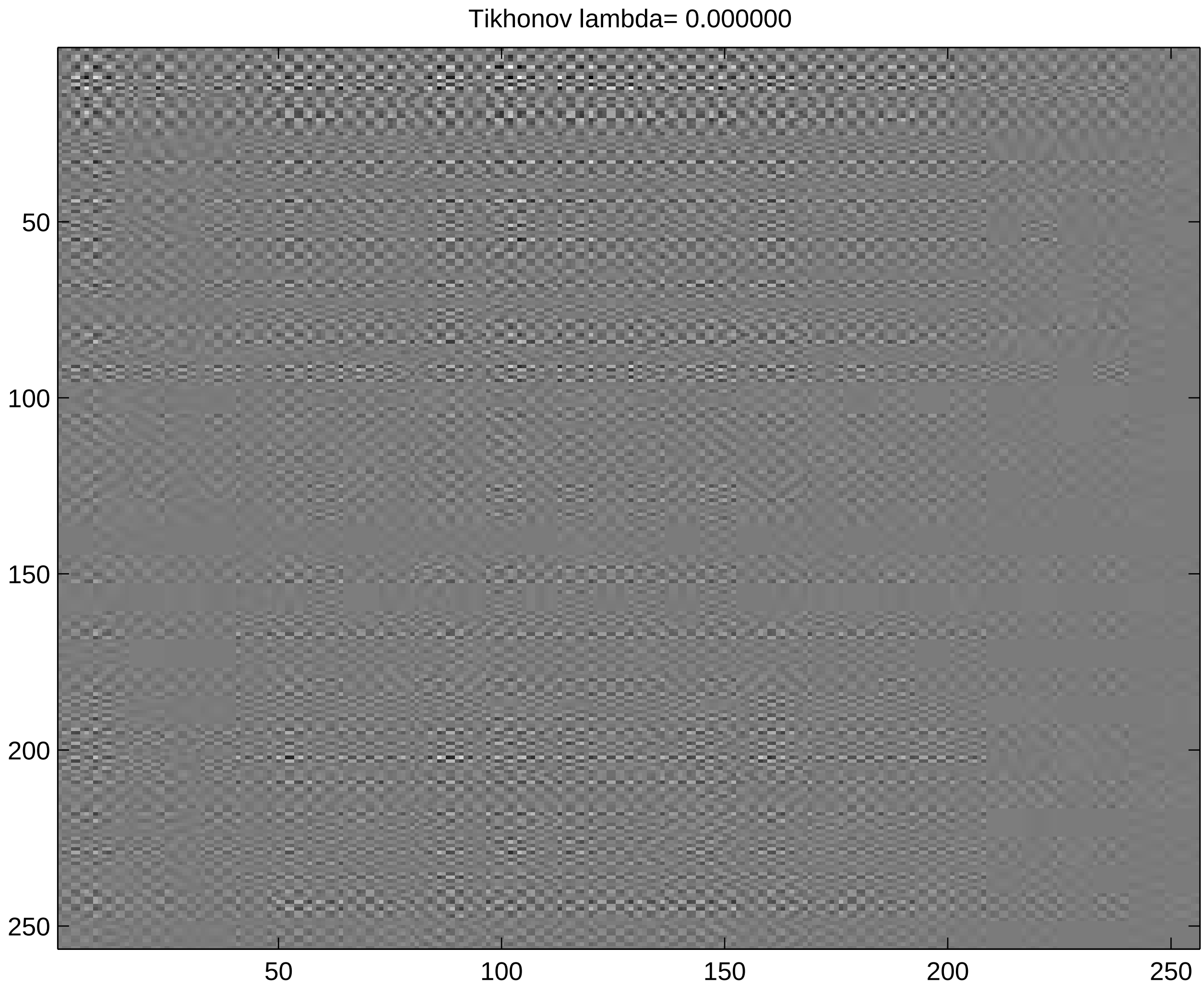

Tikhonov lambda= 0.001500
Vision is the
art of seeing
what is
invisible to
others.
50
100
150
200
250
50
100
150
200
250


Tikhonov lambda= 0.001000
Vision is the art of seeing what is invisible to others.
50
100
150
200
250
50
100
150
200
250


Tikhonov lambda= 0.000950
Vision is the
art of seeing
what is
invisible to
others.
Jonathan Swift


Tikhonov lambda= 0.000167
Vision is the
art of seeing
what is
invisible to
others.


Tikhonov lambda= 0.000050
50
100
150
200
250
50
100
150
200
250



What we have learned

- The ill-conditioning of the matrix and the noise in the data make image deblurring very difficult.
- We can deblur well using Tikhonov regularization.
- Choosing the regularization parameter is easy if the "eye" norm can be used.
- In general, a good regularization parameter is hard to find.

Efficient algorithms for solving the Tikhonov problem

We turn to the problem of solving the least squares problem.

Unquiz: Show that if $\mathbf{A}$ has a singular value decomposition $\mathbf{A} = \mathbf{U\Sigma V}^T$, then the Tikhonov problem can be transformed into the equivalent least squares problem

$$\min_{\hat{\mathbf{x}}} \left\| \begin{bmatrix} \hat{\mathbf{b}} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{\Sigma} \\ \alpha \mathbf{I} \end{bmatrix} \hat{\mathbf{x}} \right\|_2^2 \tag{1}$$

where $\hat{\mathbf{x}} = \mathbf{V}^T\mathbf{x}$ and $\hat{\mathbf{b}} = \mathbf{U}^T\mathbf{b}$.

Unquiz: Derive a linear system of equations whose solution is the solution to (**??**). Hint: set the derivative of the minimization function to zero and solve for $\hat{\mathbf{x}}$.

This gives us an algorithm to determine the Tikhonov solution to a discrete ill-posed problem. Next we consider a second method.

Method 2: Truncated SVD

Another way to regularize the problem is to truncate the singular value decomposition. The next problem demonstrates how the solution to the least squares problem can be expressed in terms of the SVD.

Unquiz: Show that the solution to the problem

$$\min_{\mathbf{x}} ||\mathbf{b} - \mathbf{Ax}||_2^2$$

can be written as

$$\mathbf{x}_{\ell s} = \mathbf{V\Sigma}^{\dagger}\mathbf{U}^T\mathbf{b} \equiv \sum_{i=1}^{n} \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\mathbf{v}_i \,,$$

where $\mathbf{u}_i$ is the ith column of $\mathbf{U}$ and $\mathbf{v}_i$ is the ith column of $\mathbf{V}$.

We see that trouble occurs in $\mathbf{x}_{\ell s}$ if we have a small value of σ_i dividing a term $\mathbf{u}_i^T\mathbf{b}$ that is dominated by error. In that case, $\mathbf{x}_{\ell s}$ is dominated by error, too.

To overcome this, Richard Hanson and also James Varah suggested truncating the expansion above:

$$\mathbf{x}_t = \sum_{i=1}^{p} \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\mathbf{v}_i$$

for some value of $p < n$.

We could determine p the same way as we determined α above, using the "eye" norm.

Extending these methods to very large problems

- The SVD gives us all the information we need to solve discrete ill-posed problems.
- It works fine for 1-dimensional problems; e.g., spectroscopy.
- For 2- and 3-dimensional problems (images, video), it is too expensive. For example, to deblur a 1-megapixel image we need an SVD of a matrix of size 10^6.
- We can use iterative methods (discussed later), but first let's consider an important special case of a very large problem that allows use of the SVD.

A Special case: Kronecker product structure in A

The set-up: We have

- a blurred, noisy image $\mathbf{G}$
- some knowledge of the blurring operator

We want to reconstruct the true original image $\mathbf{F}$.

The vectors in the linear system $\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta}$ represent the image arrays stacked by columns to form vectors.

In MATLAB notation,

$$\mathbf{x} = \texttt{reshape}(F, n, 1), \qquad \mathbf{b} = \texttt{reshape}(G, n, 1).$$

The goal in this problem is, given $\mathbf{A}$ and $\mathbf{G}$, reconstruct an approximation of the unknown image $\mathbf{F}$.

In some cases $\mathbf{A}$ can be written as a Kronecker product, $\mathbf{A} = \mathbf{C} \otimes \mathbf{G}$, and the SVD can be used.

A few facts on Kronecker products

The Kronecker product $\mathbf{C} \otimes \mathbf{G}$, where $\mathbf{C}$ is an $m \times m$ matrix, is defined to be

$$\mathbf{C} \otimes \mathbf{G} = \begin{bmatrix} c_{11}\mathbf{G} & c_{12}\mathbf{G} & \dots & c_{1m}\mathbf{G} \\ c_{21}\mathbf{G} & c_{22}\mathbf{G} & \dots & c_{2m}\mathbf{G} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1}\mathbf{G} & c_{m2}\mathbf{G} & \dots & c_{mm}\mathbf{G} \end{bmatrix}.$$

Kronecker products have a very convenient property: If $\mathbf{C} = \mathbf{U}_C \boldsymbol{\Sigma}_C \mathbf{V}_C^T$, $\mathbf{G} = \mathbf{U}_G \boldsymbol{\Sigma}_G \mathbf{V}_G^T$, then

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

where $\mathbf{U} = \mathbf{U}_C \otimes \mathbf{U}_G$, $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_C \otimes \boldsymbol{\Sigma}_G$, and $\mathbf{V} = \mathbf{V}_C \otimes \mathbf{V}_G$.

Therefore, it is possible to compute the SVD of a rather large matrix if it is the Kronecker product of two smaller ones.

See the sample MATLAB program, `projdemo.m`, illustrating this property.

Structure from the convolution integral

The image we get from taking a picture of a point source is a discrete form of a point spread function (PSF).

In some cases we have two convenient properties:

- The PSF is independent of location.
- The horizontal and vertical components of the blur can be separated.

If this is the case, then the $p \times q$ PSF array $\mathbf{P}$ can be decomposed as

$$\mathbf{P} = \mathbf{c}\,\mathbf{r}^T = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_p \end{bmatrix} \begin{bmatrix} r_1 & r_2 & \cdots & r_q \end{bmatrix}$$

where $\mathbf{r}$ represents the horizontal component of the blur (i.e., blur across the rows of the image array), and $\mathbf{c}$ represents the vertical component (i.e., blur across the columns of the image).

The special structure for this blur implies that $\mathbf{P}$ is a rank-one matrix with elements given by

$$p_{ij} = c_i\, r_j.$$

Example: It can be shown that if $p = q = 3$, the coefficient matrix takes the form

$$\mathbf{A} \quad = \quad \left[\begin{array}{ccc|ccc|ccc}
c_2r_2 & c_1r_2 & & c_2r_1 & c_1r_1 & & & & \\
c_3r_2 & c_2r_2 & c_1r_2 & c_3r_1 & c_2r_1 & c_1r_1 & & & \\
 & c_3r_2 & c_2r_2 & & c_3r_1 & c_2r_1 & & & \\
\hline
c_2r_3 & c_1r_3 & & c_2r_2 & c_1r_2 & & c_2r_1 & c_1r_1 & \\
c_3r_3 & c_2r_3 & c_1r_3 & c_3r_2 & c_2r_2 & c_1r_2 & c_3r_1 & c_2r_1 & c_1r_1 \\
 & c_3r_3 & c_2r_3 & & c_3r_2 & c_2r_2 & & c_3r_1 & c_2r_1 \\
\hline
 & & & c_2r_3 & c_1r_3 & & c_2r_2 & c_1r_2 & \\
 & & & c_3r_3 & c_2r_3 & c_1r_3 & c_3r_2 & c_2r_2 & c_1r_2 \\
 & & & & c_3r_3 & c_2r_3 & & c_3r_2 & c_2r_2
\end{array}\right]$$

$$
= \left[\begin{array}{c|c|c}
r_2 \begin{bmatrix} c_2 & c_1 & \\ c_3 & c_2 & c_1 \\ & c_3 & c_2 \end{bmatrix} & r_1 \begin{bmatrix} c_2 & c_1 & \\ c_3 & c_2 & c_1 \\ & c_3 & c_2 \end{bmatrix} & 0 \\ \hline
r_3 \begin{bmatrix} c_2 & c_1 & \\ c_3 & c_2 & c_1 \\ & c_3 & c_2 \end{bmatrix} & r_2 \begin{bmatrix} c_2 & c_1 & \\ c_3 & c_2 & c_1 \\ & c_3 & c_2 \end{bmatrix} & r_1 \begin{bmatrix} c_2 & c_1 & \\ c_3 & c_2 & c_1 \\ & c_3 & c_2 \end{bmatrix} \\ \hline
0 & r_3 \begin{bmatrix} c_2 & c_1 & \\ c_3 & c_2 & c_1 \\ & c_3 & c_2 \end{bmatrix} & r_2 \begin{bmatrix} c_2 & c_1 & \\ c_3 & c_2 & c_1 \\ & c_3 & c_2 \end{bmatrix}
\end{array}\right].
$$

In general the coefficient matrix $\mathbf{A}$ for separable blur has block structure of the form

$$
\mathbf{A} = \mathbf{A}_r \otimes \mathbf{A}_c = \begin{bmatrix} a_{11}^{(r)}\mathbf{A}_c & a_{12}^{(r)}\mathbf{A}_c & \cdots & a_{1n}^{(r)}\mathbf{A}_c \\ a_{21}^{(r)}\mathbf{A}_c & a_{22}^{(r)}\mathbf{A}_c & \cdots & a_{2n}^{(r)}\mathbf{A}_c \\ \vdots & \vdots & & \vdots \\ a_{n1}^{(r)}\mathbf{A}_c & a_{n2}^{(r)}\mathbf{A}_c & \cdots & a_{nn}^{(r)}\mathbf{A}_c \end{bmatrix}
$$

where $\mathbf{A}_c$ is an $m \times m$ matrix, and $\mathbf{A}_r$ is an $n \times n$ matrix with entries denoted by $a_{ij}^{(r)}$.

$$
\mathbf{A} = \mathbf{A}_r \otimes \mathbf{A}_c \tag{2}
$$

where, if the images $\mathbf{X}$ and $\mathbf{B}$ have $p \times q$ pixels, then $\mathbf{A}_r$ is $q \times q$ and $\mathbf{A}_c$ is $p \times p$.

Constructing the Kronecker Product from the PSF

To explictly construct $\mathbf{A}_r$ and $\mathbf{A}_c$ from the PSF array, $\mathbf{P}$, we need to be able to find the vectors $\mathbf{r}$ and $\mathbf{c}$.

This can be done by computing the largest singular value, and corresponding singular vectors, of $\mathbf{P}$.

In Matlab, this can be done efficiently with the built-in `svds` function:

```
[u, s, v] = svds(P, 1);
c = sqrt(s)*u;
r = sqrt(s)*v;
```

In a careful implementation we would compute the first two singular values and check that the ratio of s_2/s_1 is small enough to neglect all but the first and hence that the Kronecker product representation $\mathbf{A}_r \otimes \mathbf{A}_c$ is an accurate representation of $\mathbf{A}$.

If this is the case, then the $p \times q$ PSF array $\mathbf{P}$ can be decomposed as

$$\mathbf{P} = \mathbf{c}\,\mathbf{r}^T = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_p \end{bmatrix} \begin{bmatrix} r_1 & r_2 & \cdots & r_q \end{bmatrix}$$

where $\mathbf{r}$ represents the horizontal component of the blur (i.e., blur across the rows of the image array), and $\mathbf{c}$ represents the vertical component (i.e., blur across the columns of the image).

With the Kronecker product as a tool, we can use Tikhonov or Truncated SVD as a regularization tool for image processing.

In order to solve our image deblurring problem, we need to operate rather carefully with the small matrices; otherwise, storage quickly becomes an issue. Again, see the sample program for guidance.

Final Words

- Discrete ill-posed problems require regularization in order to produce physically meaningful solutions.
- Two examples of regularization methods are Tikhonov and truncated SVD.
- Structure in the convolution must be exploited if the problem is big.
- We have several important issues still to consider, including choice of the regularization parameter and solution of the problem if there is significant error in $\mathbf{A}$.