Exploiting Kronecker Product Structure in Image Restoration

James G. Nagy
Emory University
Atlanta, GA

Outline

1. The Basic Problem
2. Regularization
3. SVD / Kronecker Product Approximations
4. Iterative Methods / Preconditioning
5. Summary

Basic Problem

Linear system of equations

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

where

- A, $\mathbf{b}$ are known
- A is large, structured, ill-conditioned
- Goal: Compute an approximation of $\mathbf{x}$

Applications: Ill-posed inverse problems.

- Geomagnetic Prospecting
- Tomography
- Image Restoration

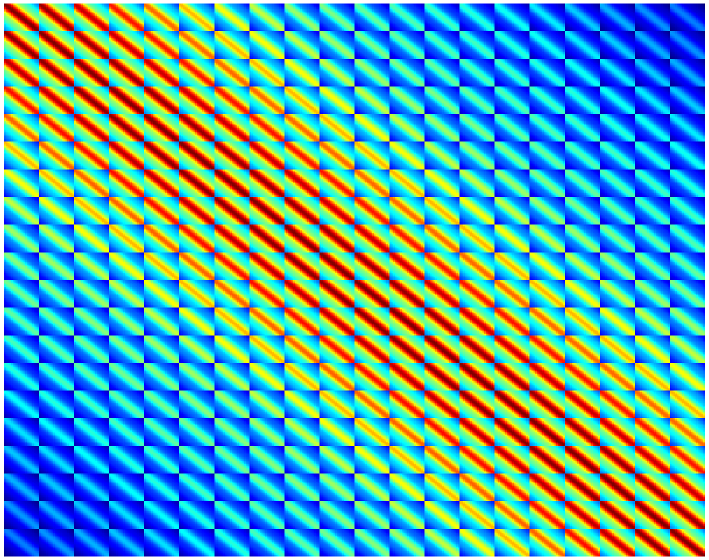
  $\mathbf{b}$ = observed image

  A = blurring matrix (structured)

  $\mathbf{e}$ = noise

  $\mathbf{x}$ = true image

A = matrix



b = blurred, noisy image

x = true image

Basic Problem – Properties

Computational difficulties revealed through SVD:

Let $\quad A = U\Sigma V^T \quad$ where

- $\Sigma =$diag$(\sigma_1, \sigma_2, \ldots, \sigma_N)\,, \qquad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N \geq 0$

- $U^T U = I\,, \quad V^T V = I$

For ill-posed inverse problems,

- $\sigma_1 \approx 1$, small singular values cluster at 0

- small singular values $\Rightarrow$ oscillating singular vectors

Basic Problem – Properties

Inverse solution for noisy, ill-posed problems:

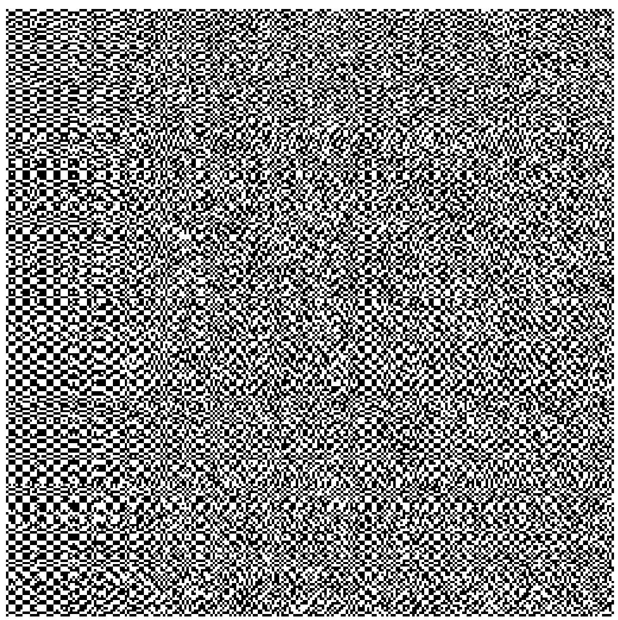
If $A = U\Sigma V^T$, then

$$\begin{aligned}
\hat{\mathbf{x}} &= A^{-1}(\mathbf{b} + \mathbf{e}) \\
&= V\Sigma^{-1}U^T(\mathbf{b} + \mathbf{e}) \\
&= \sum_{i=1}^{n} \frac{\mathbf{u}_i^T(\mathbf{b} + \mathbf{e})}{\sigma_i}\mathbf{v}_i \\
&= \sum_{i=1}^{n} \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\mathbf{v}_i + \sum_{i=1}^{n} \frac{\mathbf{u}_i^T\mathbf{e}}{\sigma_i}\mathbf{v}_i \\
&= \mathbf{x} + \text{error}
\end{aligned}$$

b = blurred, noisy image

x = true image

x = inverse solution



Regularization

Basic Idea: Filter out effects of small singular values.

$$\mathbf{x}_{\text{reg}} = \sum_{i=1}^{n} \phi_i \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i$$

where the "filter factors" satisfy

$$\phi_i \approx \begin{cases} 1 & \text{if } \sigma_i \text{ is large} \\ 0 & \text{if } \sigma_i \text{ is small} \end{cases}$$

Regularization

Some regularization methods:

1. Truncated SVD

$$\mathbf{x}_{\mathsf{tsvd}} = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i$$

2. Tikhonov

$$\mathbf{x}_{\mathsf{tik}} = \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i$$

3. Wiener

$$\mathbf{x}_{\mathsf{wien}} = \sum_{i=1}^{n} \frac{\delta_i \sigma_i^2}{\delta_i \sigma_i^2 + \gamma_i^2} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i$$

SVD Approximation

Basic idea:

- Decompose A as: (Van Loan and Pitsianis, '93)

$$A = C_1 \otimes D_1 + C_2 \otimes D_2 + \cdots + C_k \otimes D_k$$

where $C_1 \otimes D_1 = \text{argmin}||A - C \otimes D||_F$.

- Choose a "structured" (or sparse) U and V.

- Let $\Sigma = \text{argmin}||A - U\Sigma V^T||_F$.

That is,

$$\begin{aligned} \Sigma &= \text{diag}\left(U^T A V\right) \\ &= \text{diag}\left(U^T \left(\sum_{i=1}^{k} C_i \otimes D_i\right) V\right) \end{aligned}$$

SVD Approximation

Choices for U and V depend on problem (application).

- Since

  $$A = C_1 \otimes D_1 + C_2 \otimes D_2 + \cdots + C_k \otimes D_k$$

  and

  $$C_1 \otimes D_1 = \text{argmin} ||A - C \otimes D||_F$$

  we might use singular vectors of $C_1 \otimes D_1$.

- For image restoration, we also use

  Fourier Transforms (FFTs)

  Discrete Cosine Transforms (DCTs)

Efficient Implementation for Image Restoration

1. Matrix Structure

2. Efficiently computing SVD approximation

Matrix Structure in Image Restoration

First, how do we get the matrix, A?

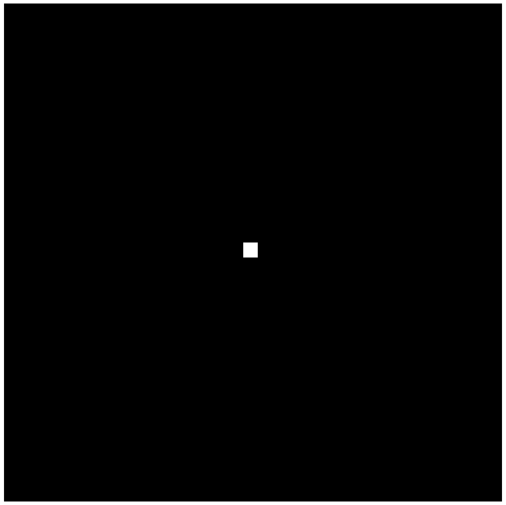
- Using linear algebra notation, the i-th column of A can be written as:

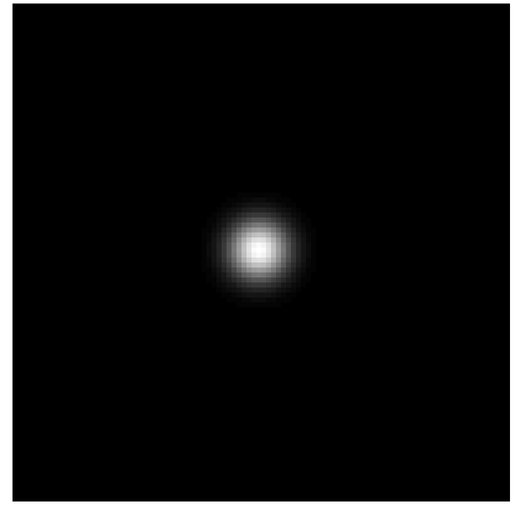
$$A\mathbf{e}_i = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_i & \cdots & \mathbf{a}_n \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{a}_i$$

- In an imaging system,

$$\begin{aligned} \mathbf{e}_i &= \text{point source} \\ A\mathbf{e}_i &= \text{point spread function (PSF)} \end{aligned}$$

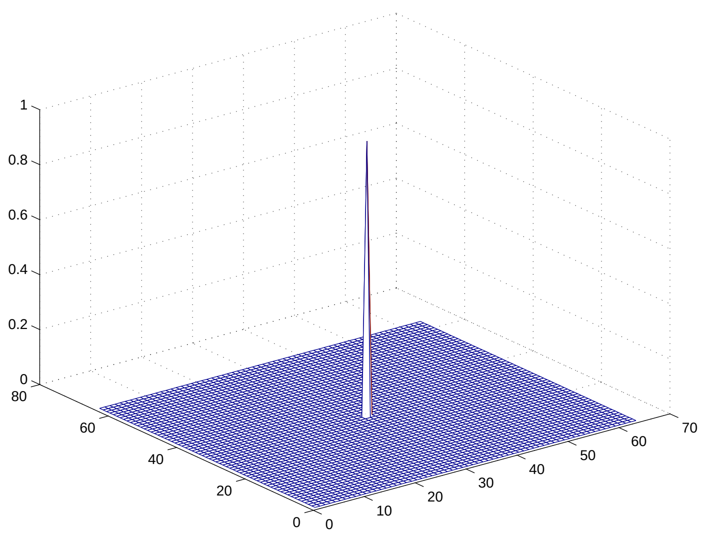
Matrix Structure in Image Restoration

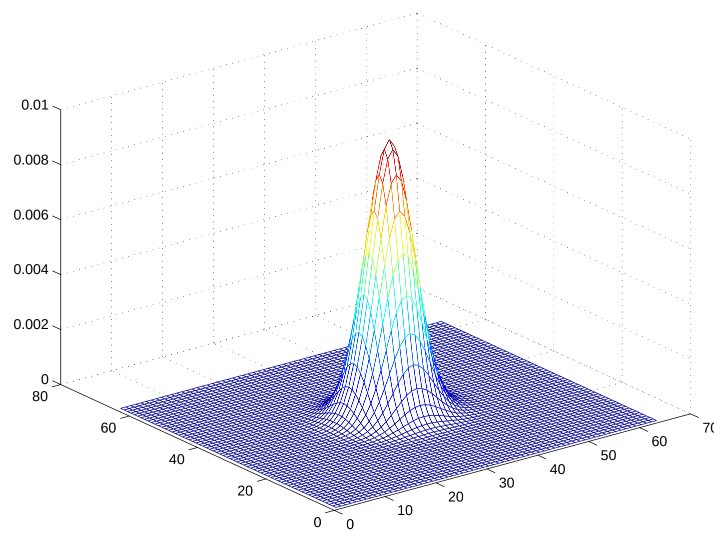




point source

PSF

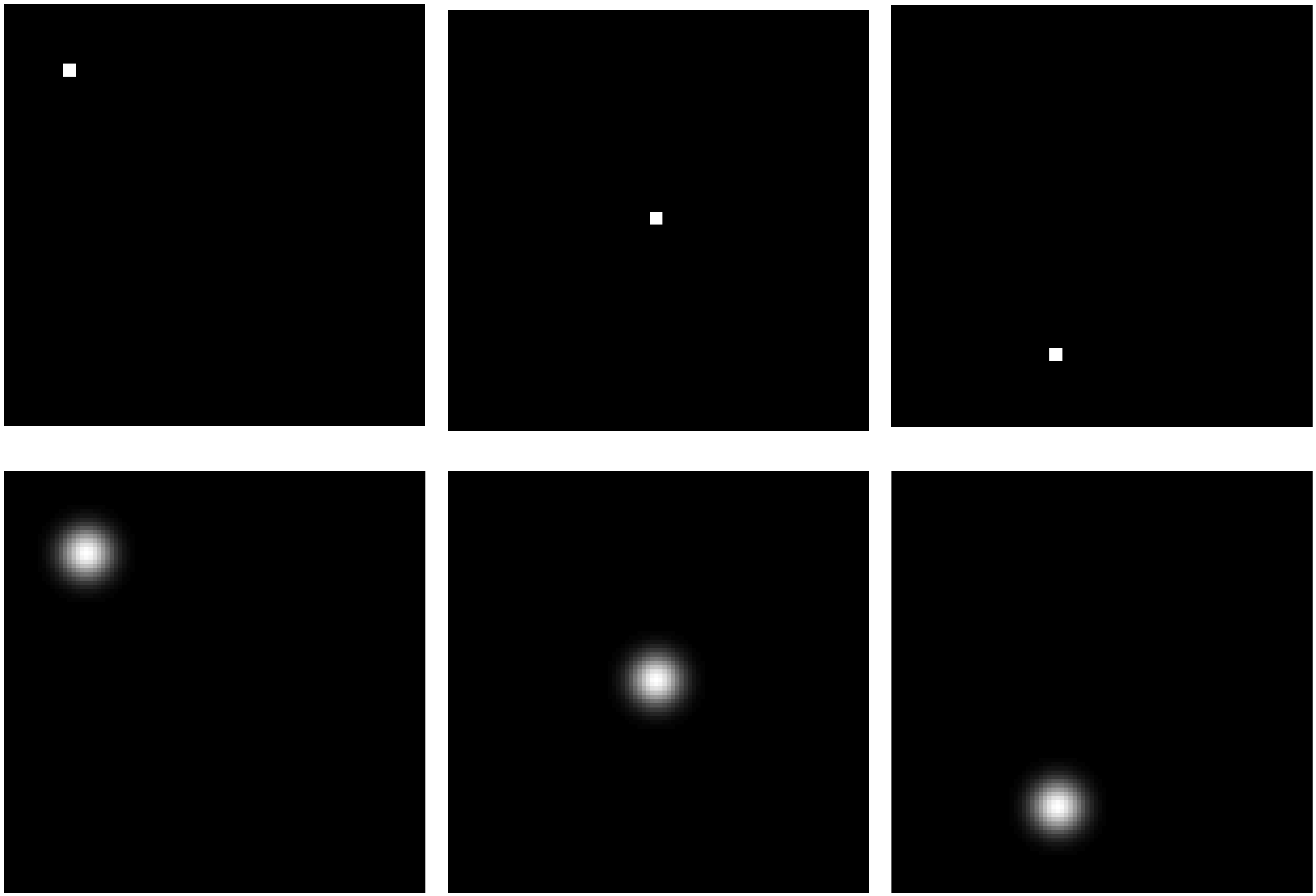




Matrix Structure in Image Restoration

Spatially invariant PSF implies:

$\mathbf{e}_i$ $\mathbf{e}_j$ $\mathbf{e}_k$



$A\mathbf{e}_i$ $A\mathbf{e}_j$ $A\mathbf{e}_k$

Matrix Structure in Image Restoration

That is, spatially invariant implies

- Each column of A is identical, modulo shift.
- One point PSF is enough to fully describe A.
- A has Toeplitz structure.

Matrix Structure in Image Restoration

$$
\mathbf{e}_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \text{blur} \rightarrow \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \rightarrow A\mathbf{e}_5 = \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{31} \\ p_{32} \\ p_{33} \end{bmatrix}
$$

$$
A = \left[\begin{array}{ccc|ccc|ccc}
p_{22} & p_{21} & & p_{12} & p_{11} & & & & \\
p_{23} & p_{22} & p_{21} & p_{13} & p_{12} & p_{11} & & & \\
 & p_{23} & p_{22} & & p_{13} & p_{12} & & & \\
\hline
p_{32} & p_{31} & & p_{22} & p_{21} & & p_{12} & p_{11} & \\
p_{33} & p_{32} & p_{31} & p_{23} & p_{22} & p_{21} & p_{13} & p_{12} & p_{11} \\
 & p_{33} & p_{32} & & p_{23} & p_{22} & & p_{13} & p_{12} \\
\hline
 & & & p_{32} & p_{31} & & p_{22} & p_{21} & \\
 & & & p_{33} & p_{32} & p_{31} & p_{23} & p_{22} & p_{21} \\
 & & & & p_{33} & p_{32} & & p_{23} & p_{22}
\end{array}\right]
$$

Matrix Structure in Image Restoration

Matrix Summary

Boundary Condition	Matrix Structure
zero	BTTB
periodic	BCCB (use FFT)
reflexive (strongly symmetric)	BTTB+BTHB+BHTB+BHHB (use DCT)

B = block
T = Toeplitz
C = circulant
H = Hankel

Matrix Structure in Image Restoration

For a separable PSF, we get:

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} = \mathbf{c}\mathbf{d}^T = \begin{bmatrix} c_1 d_1 & c_1 d_2 & c_1 d_3 \\ c_2 d_1 & c_2 d_2 & c_2 d_3 \\ c_3 d_1 & c_3 d_2 & c_3 d_3 \end{bmatrix} \to A\mathbf{e}_5 = \begin{bmatrix} c_1 \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} \\ c_2 \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} \\ c_3 \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} \end{bmatrix}$$

$$\left[\begin{array}{c|c|c} c_2 \begin{pmatrix} d_2 & d_1 & \\ d_3 & d_2 & d_1 \\ & d_3 & d_2 \end{pmatrix} & c_1 \begin{pmatrix} d_2 & d_1 & \\ d_3 & d_2 & d_1 \\ & d_3 & d_2 \end{pmatrix} & \\ \hline c_3 \begin{pmatrix} d_2 & d_1 & \\ d_3 & d_2 & d_1 \\ & d_3 & d_2 \end{pmatrix} & c_2 \begin{pmatrix} d_2 & d_1 & \\ d_3 & d_2 & d_1 \\ & d_3 & d_2 \end{pmatrix} & c_1 \begin{pmatrix} d_2 & d_1 & \\ d_3 & d_2 & d_1 \\ & d_3 & d_2 \end{pmatrix} \\ \hline & c_3 \begin{pmatrix} d_2 & d_1 & \\ d_3 & d_2 & d_1 \\ & d_3 & d_2 \end{pmatrix} & c_2 \begin{pmatrix} d_2 & d_1 & \\ d_3 & d_2 & d_1 \\ & d_3 & d_2 \end{pmatrix} \end{array}\right] = C \otimes D$$

Matrix Structure in Image Restoration

If the PSF is not separable, we can still compute:

$$P = \sum_{i=1}^{r} \mathbf{c}_i \mathbf{d}_i^T \qquad \text{(sum of rank-1 matrices)}$$

and therefore, get

$$A = \sum_{i=1}^{r} C_i \otimes D_i \qquad \text{(sum of Kron. products)}$$

In fact, we can get "optimal" decompositions.

(Kamm, N, '00; N., Ng, Perrone, 03)

SVD Approximation for Image Restoration

Use $A \approx U \Sigma V^T$, where

- If $\mathcal{F} \left(\sum C_i \otimes D_i \right) \mathcal{F}^*$ is best,

$$U = V = \mathcal{F}^* , \quad \Sigma = \text{diag} \left(\mathcal{F} \left(\sum C_i \otimes D_i \right) \mathcal{F}^* \right)$$

- If $\mathcal{C} \left(\sum C_i \otimes D_i \right) \mathcal{C}^T$ is best,

$$U = V = \mathcal{C}^T , \quad \Sigma = \text{diag} \left(\mathcal{C} \left(\sum C_i \otimes D_i \right) \mathcal{C}^T \right)$$

- If $(U_c \otimes U_d)^T \left(\sum C_i \otimes D_i \right) (V_c \otimes V_d)$ is best,

$$U = U_c \otimes U_d \, , \quad V = V_c \otimes V_d \, ,$$

$$\Sigma = \text{diag} \left((U_c \otimes U_d)^T \left(\sum C_i \otimes D_i \right) (V_c \otimes V_d) \right)$$

3-Dimensional Problems

- Need orthogonal tensor decompositions

  SIMAX: de Lathauwer, de Moor, Vandewalle, '00;

  Kolda, '01;

  Zhang, Golub, '01

- We use HOSVD (de Lathauwer, de Moor, Vandewalle, '00):

$$P = \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} \sum_{k=1}^{r_3} \sigma_{ijk} \mathbf{u}_i \circ \mathbf{v}_j \circ \mathbf{w}_k \, .$$

- These vectors define matrices C_i, D_j and F_k, with

$$A = \sum \sum_{\sigma_{ijk} \neq 0} \sum C_i \otimes D_j \otimes F_k \, ,$$

3-Dimensional Problems

- Need orthogonal tensor decompositions

  SIMAX: de Lathauwer, de Moor, Vandewalle, '00;

  Kolda, '01;

  Zhang, Golub, '01

- We use HOSVD (de Lathauwer, de Moor, Vandewalle, '00):

$$P = \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} \sum_{k=1}^{r_3} \sigma_{ijk} \mathbf{u}_i \circ \mathbf{v}_j \circ \mathbf{w}_k \,.$$

- These vectors define matrices C_i, D_j and F_k, with

$$A = \sum \sum_{\sigma_{ijk} \neq 0} \sum C_i \otimes D_j \otimes F_k \,,$$

Iterative Regularization

Basic idea:

- Use an iterative method (e.g., conjugate gradients)

- Terminate iteration before theoretical convergence:
  - Early iterations reconstruct solution.
  - Later iterations reconstruct noise.

Some important methods:

- CGLS, LSQR, GMRES

- MR2 (Hanke, '95)

- MRNSD (Kaufman, '93; N., Strakos, '00)

Iterative Regularization

Efficient for large problems, provided

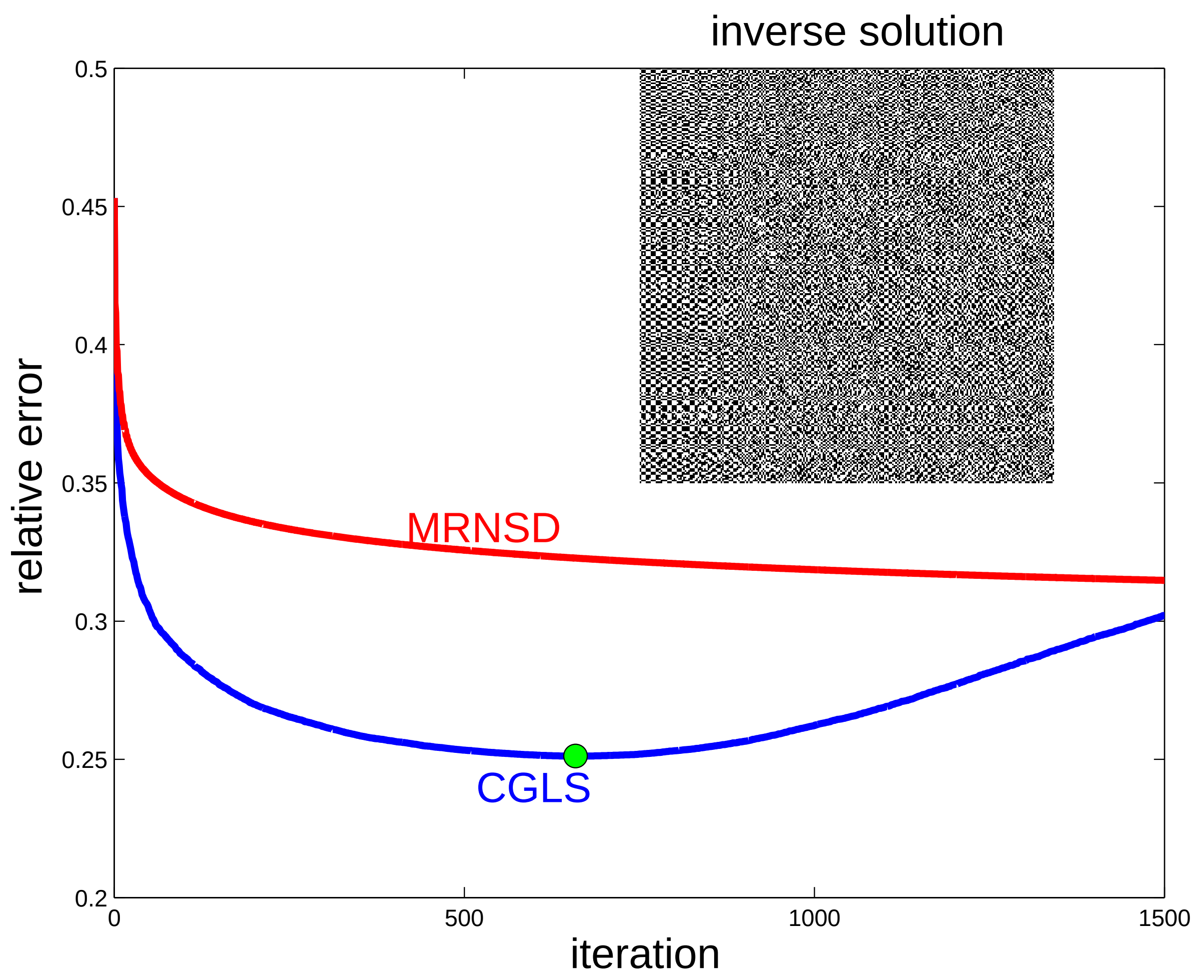
1. Multiplication with A is not expensive.

   Image restoration $\Leftrightarrow$ Use FFTs

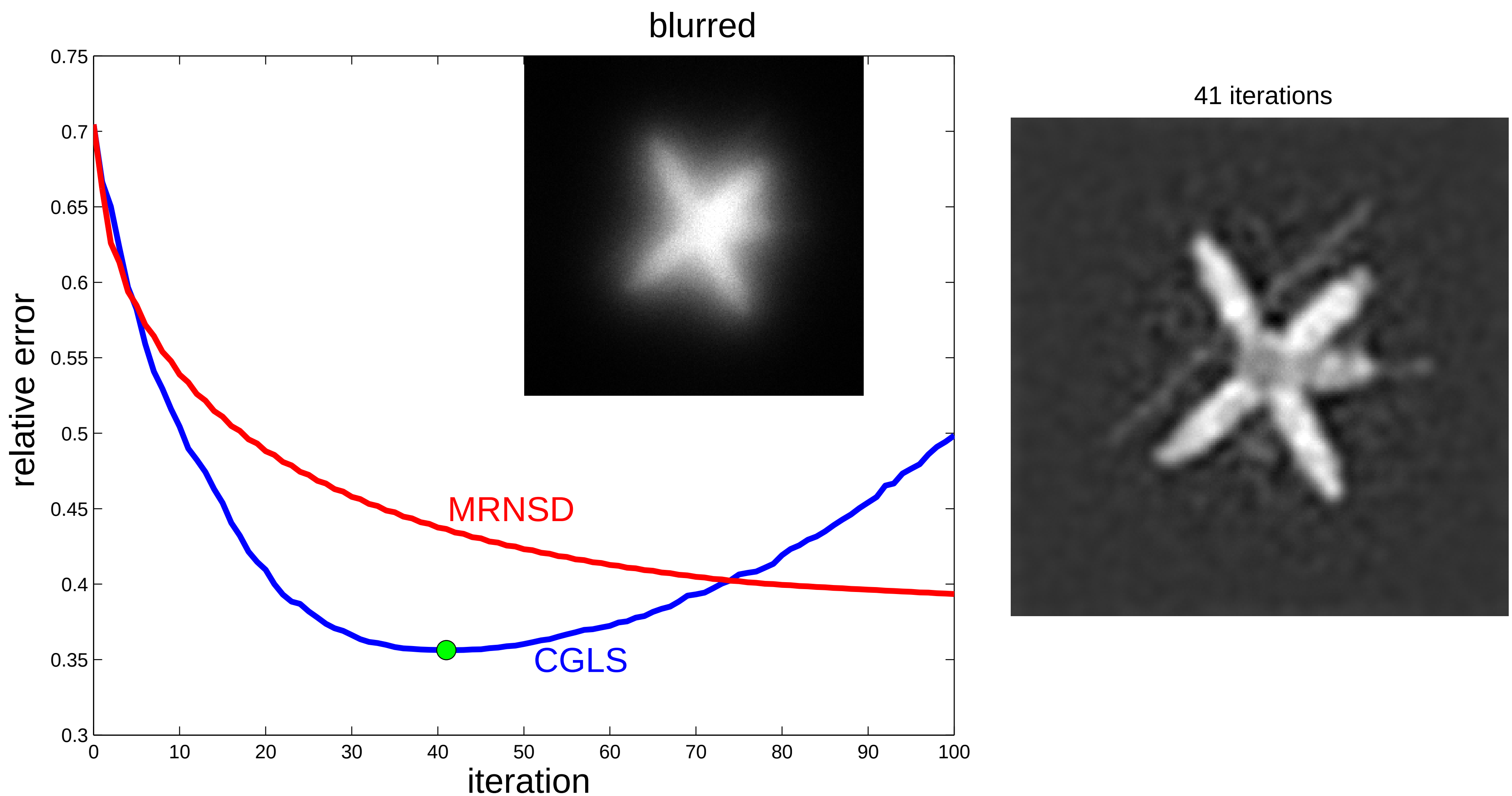
2. Convergence is rapid enough.

   - CGLS, LSQR, GMRES, MR2 often fast, especially for severely ill-posed, noisy problems.

   - MRNSD based on steepest descent, typically converges very slowly.



659 iterations


blurred
41 iterations
relative error
iteration
MRNSD
CGLS
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0
10
20
30
40
50
60
70
80
90
100

Preconditioning for Speed

Typical approach for $A\mathbf{x} = \mathbf{b}$

- Find matrix P so that $P^{-1}A \approx I$.

- "Ideal" choice: $P = A$

  In this case, converge in one iteration to $\mathbf{x} = A^{-1}\mathbf{b}$

For ill-conditioned, noisy problems

- Inverse solution is corrupted with noise

- "Ideal" regularized preconditioner: If $A = U\Sigma V^T$

  (Hanke, N., Plemmons, '93)

$$P = U\Sigma_k V^T = U\text{diag}(\sigma_1, \ldots, \sigma_k, 1, \ldots, 1)V^T$$

Preconditioning for Speed

Notice that the preconditioned system is:

$$
\begin{aligned}
P^{-1}A &= (U\Sigma_k V^T)^{-1}(U\Sigma V^T) \\
&= V\Sigma_k^{-1}\Sigma V^T \\
&= V\Delta V^T
\end{aligned}
$$

where $\Delta = \text{diag}(1, \ldots, 1, \sigma_{k+1}, \ldots, \sigma_n)$

That is,

- Large (good) singular values clustered at 1.
- Small (bad) singular values not clustered.

Preconditioning for Speed

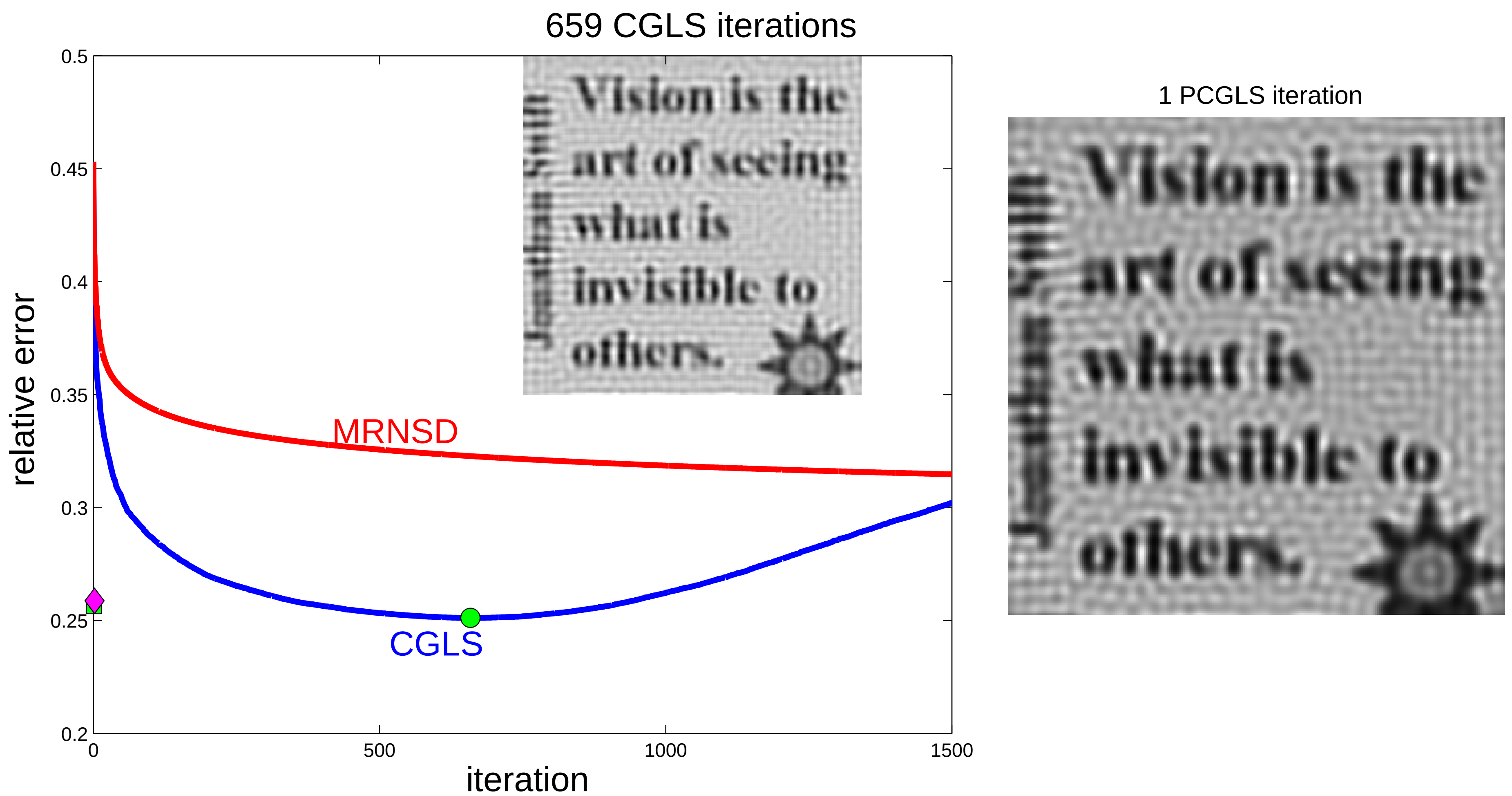
Remaining questions:

1. How to choose truncation index, k?

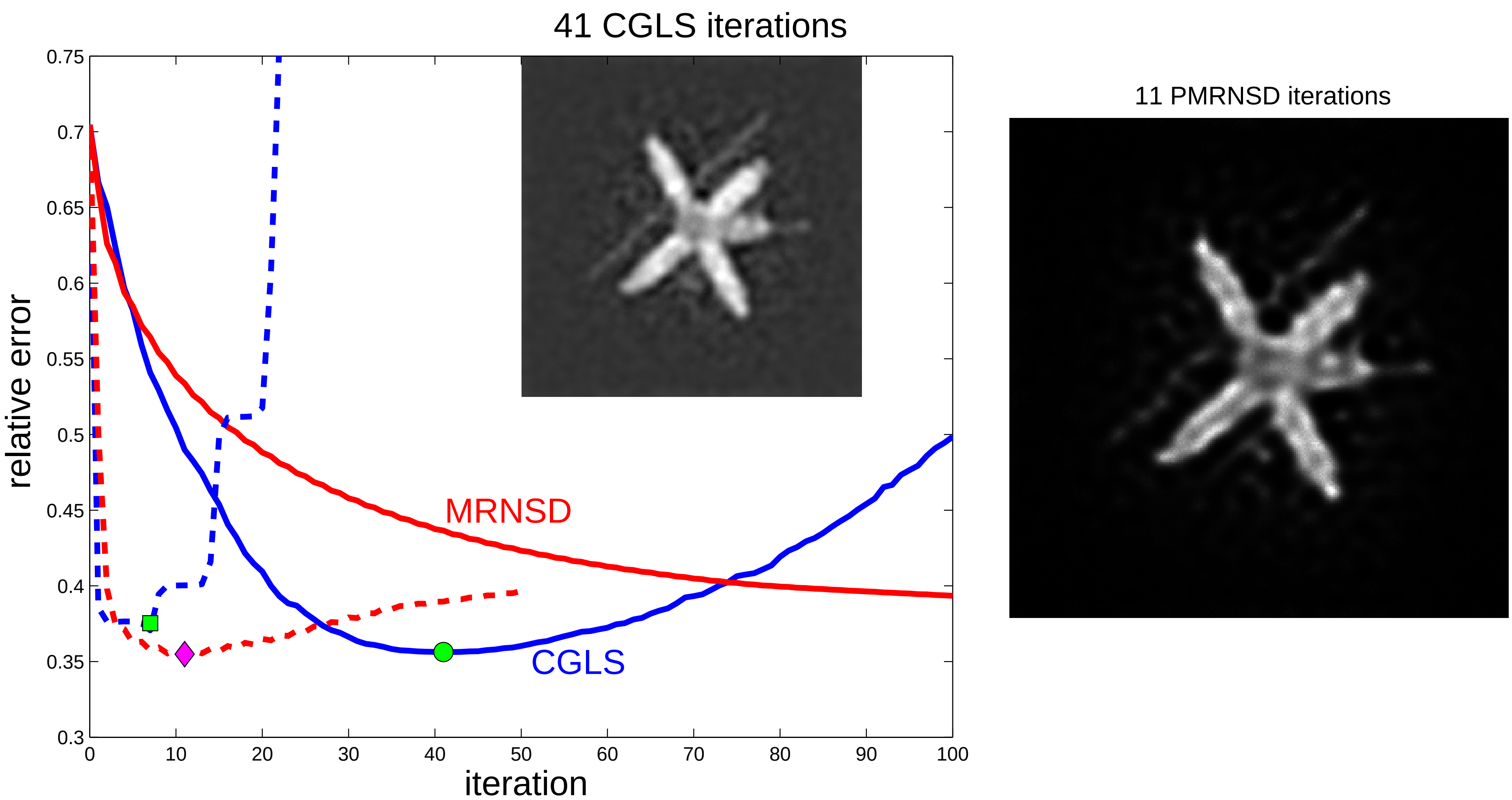
   Use regularization parameter choice methods,
   e.g., GCV, L-curve, Picard condition

2. We can't compute SVD, so now what?

   Use SVD approximation.



659 CGLS iterations
1 PCGLS iteration
relative error
iteration
MRNSD
CGLS
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0
500
1000
1500
Vision is the art of seeing what is invisible to others.


41 CGLS iterations
11 PMRNSD iterations
relative error
iteration
MRNSD
CGLS
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0
10
20
30
40
50
60
70
80
90
100

The End

- Preconditioning ill-posed problems is difficult, but possible.
- Can build approximate SVD from Kronecker product approximations.
- Can implement efficiently for image restoration.
- Matlab software: RestoreTools (N., Palmer, Perrone)

  Object oriented approach for image restoration.
  http://www.mathcs.emory.edu/~nagy/RestoreTools/

  Related software for ill-posed problems
  (Hansen, Jacobsen)
  http://www.imm.dtu.dk/~pch/Regutools/