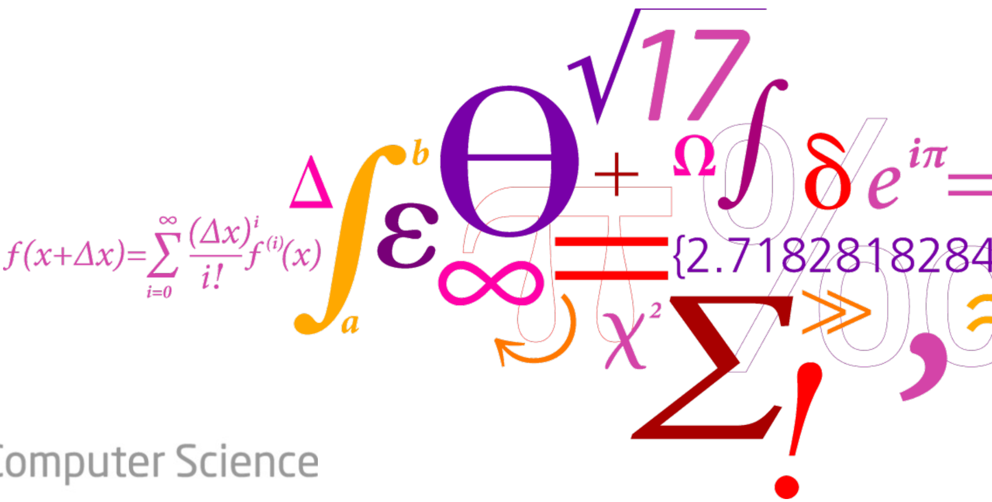
Repeated Blurring Operations Produce Sharper Images:
Image Deblurring with Krylov Subspace Methods

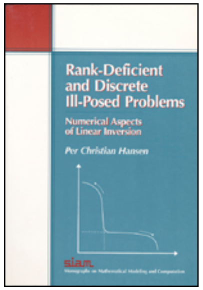
Per Christian Hansen
Technical University of Denmark

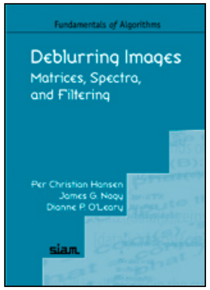


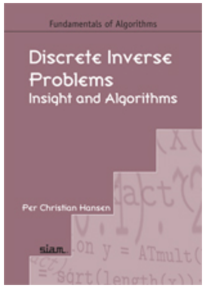
DTU Compute
Department of Applied Mathematics and Computer Science

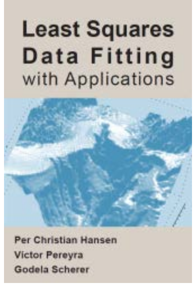
About Me …

- Professor of Scientific Computing at DTU
- Interests: inverse problems, tomography, regularization algorithms, matrix computations, image deblurring, signal processing, Matlab software, …
- Head of the project High-Definition Tomography, funded by an ERC Advanced Research Grant.
- Author of four books.









- Regularization Tools was developed on my “portable” Mac SE/30.

Image Deblurring

Sharp image

Forward problem

Blurred image

Reconstruction

Inverse Problem

Our model: $A\,x = b$

Blurring process

Sharp image

Blurred image

Nutrition Facts:

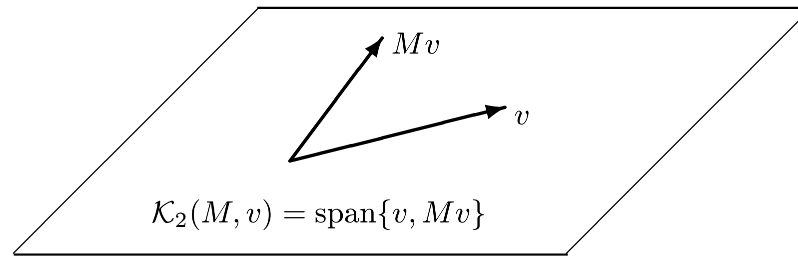
- Introduction to iterative methods (10%)
- Image deblurring/ reconstruction (20%)
- Inverse problems (10%)
- Regularization by Projection (9%)
- CGLS (12%)
- Semi-convergence (12%)
- Other methods (10%)
- Noise propagation (12%)
- Other matters (5%)

Krylov Subspaces

Given a square matrix M and a vector v, the associated Krylov subspace is defined by

$$\mathcal{K}_k(M, v) \equiv \text{span}\{v, Mv, M^2 v, \ldots, M^{k-1} v\}, \qquad k = 1, 2, \ldots$$

with $\dim(\mathcal{K}_k(M, v)) \square\ k$.



Krylov subspaces have many important applications in scientific computing:

- solving large systems of linear equations,
- computing eigenvalues,
- solving algebraic Riccati equations, and
- determining controllability in a control system.

They are also important tools for regularization of large-scale discretizations of inverse problems, which is the topic of this talk.

The CGLS Algorithm

The CGLS algorithm produces a sequence of iterates $x^{(k)}$ which solve

$$\min \|A\,x - b\|_2 \qquad \text{subject to} \qquad x \in \mathcal{K}_k(A^T b, A^T A)\ .$$

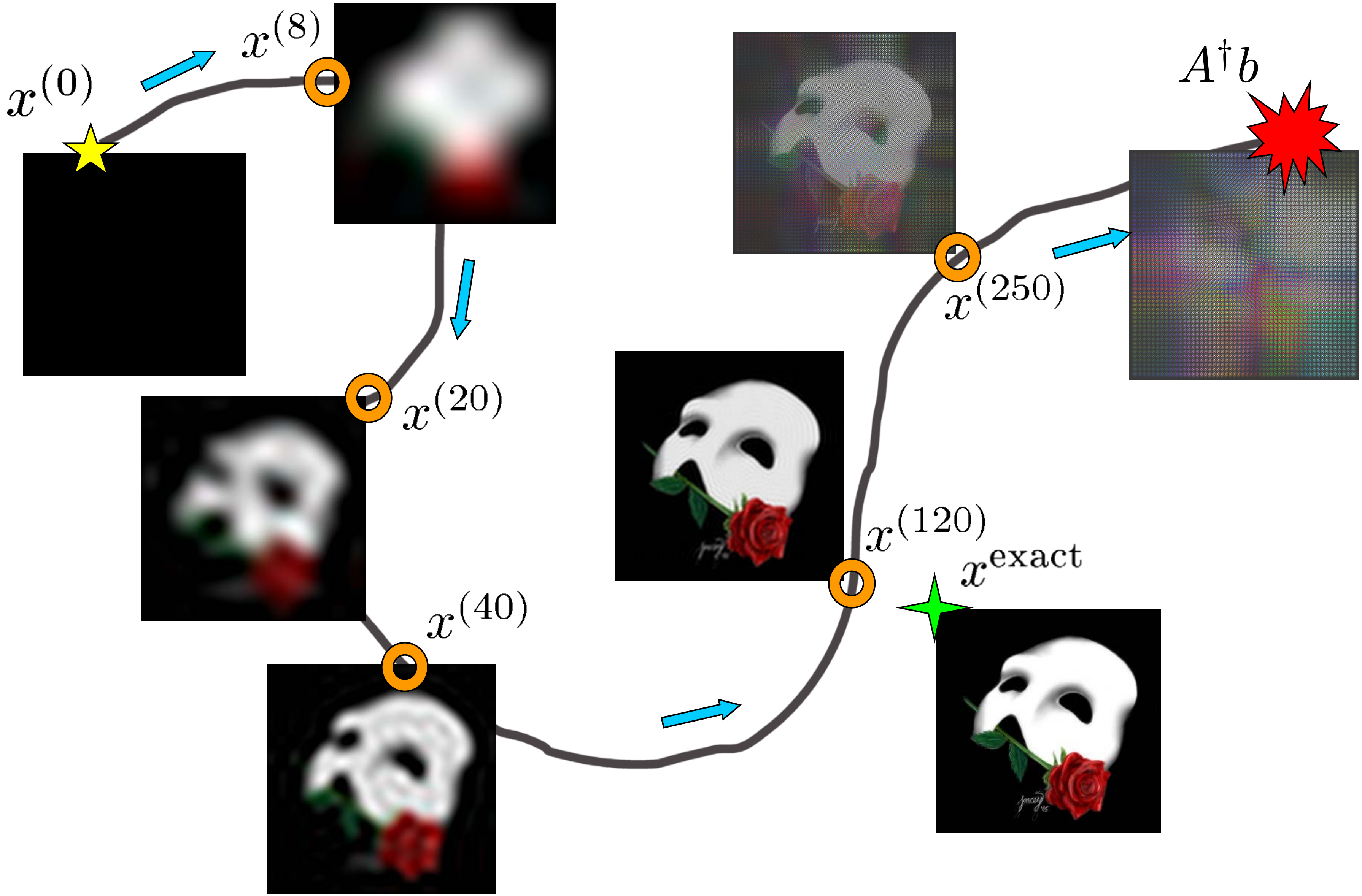
$$\left.\begin{aligned} x^{(0)} &= \text{starting vector (e.g., zero)} \\ r^{(0)} &= b - A\,x^{(0)} \\ d^{(0)} &= A^T r^{(0)} \end{aligned}\right\} \text{Initialization}$$

$$\begin{aligned}
&\text{for } k = 1, 2, \ldots \\
&\qquad \bar{\alpha}_k = \|A^T r^{(k-1)}\|_2^2 / \|A\,d^{(k-1)}\|_2^2 \\
&\qquad x^{(k)} = x^{(k-1)} + \bar{\alpha}_k\, d^{(k-1)} \\
&\qquad r^{(k)} = r^{(k-1)} - \bar{\alpha}_k\, \underbrace{A\,d^{(k-1)}}_{\text{Mult. with } A} \\
&\qquad \bar{\beta}_k = \|\underbrace{A^T r^{(k)}}_{\text{Mult. with } A^{\mathrm{T}}}\|_2^2 / \|A^T r^{(k-1)}\|_2^2 \\
&\qquad d^{(k)} = A^T r^{(k)} + \bar{\beta}_k\, d^{(k-1)} \\
&\text{end}
\end{aligned}$$

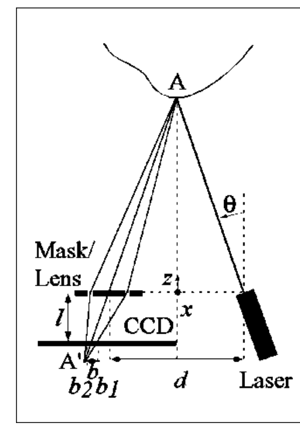
Our model: $A\,x = b$

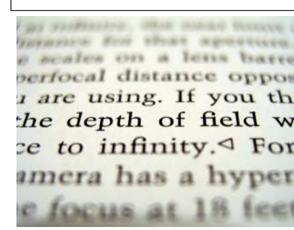
Multiplication with A and A^T is a blurring operation!

Illustration of Semi-Convergence



Sources of Blurred Images





Some Types of Blur and Distortion

From the camera:

- the lens is out of focus,
- imperfections in the lens, and
- noise in the CCD and the analog/digital converter.

From the environments:

- motion of the object (or camera),
- fluctuations in the light's path (turbulence), and
- false light, cosmic radiation (in astronomical images).

Given a mathematical/statistical ***model*** of the blur/distortion, we can ***deblur*** the image and compute a sharper reconstruction (as apposed to ”cosmetic improvements” by PhotoShop etc).

Top 10 Algorithms

J. J. Dongarra, F. Sullivan et al., *The Top 10 Algorithms*, IEEE Computing in Science and Engineering, 2 (2000), pp. 22-79.

1946: The Monte Carlo method (Metropolis Algorithm).

1947: The Simplex Method for Linear Programming.

1950: Krylov Subspace Methods (CG, CGLS, Arnoldi, etc.).

1951: Decomposition Approach to matrix computations.

1957: The Fortran Optimizing Compiler.

1961: The QR Algorithm for computing eigenvalues and –vectors.

1962: The Quicksort Algorithm.

1965: The Fast Fourier Transform algorithm.

1977: The Integer Relation Detection Algorithm.

1987: The Fast Multipole Algorithm for *N*-body simulations.

Key algorithms in image deblurring.

The Deblurring Problem

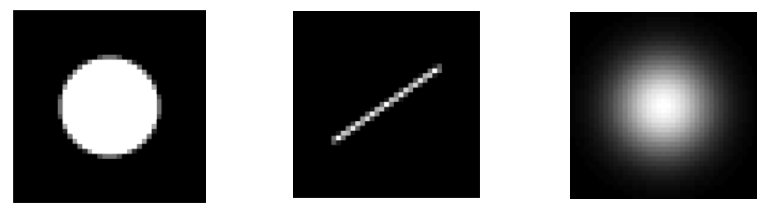
Fredholm integral equation of the first kind:

$$\int_0^1 \int_0^1 K(x, y; x', y')\, f(x, y)\, dx\, dy = g(x', y') \,, \qquad 0 \square x', y' \square 1.$$

Think of f as an unknown sharp image, and g as the blurred version.

Think of K as a model for the point spread function.

Examples of
point spread functions



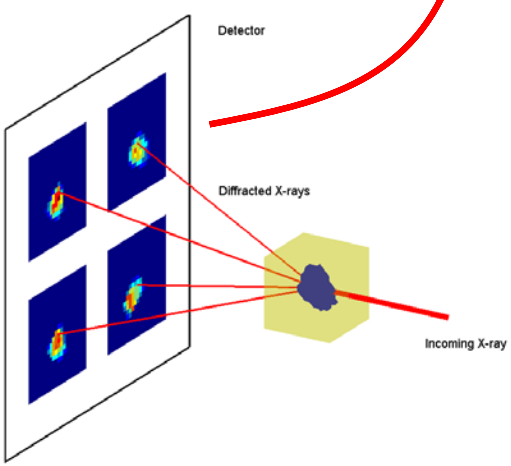
Discretization yields a LARGE system of linear equations: $A\,x = b$.

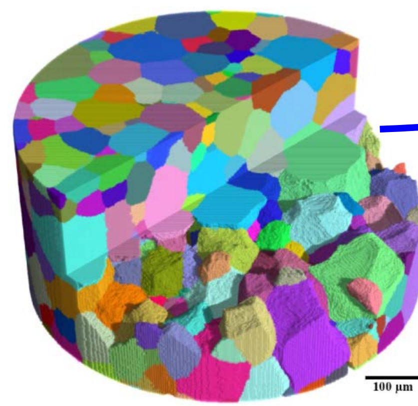
Two important aspects related to this system:

- Use the right boundary conditions.
- The matrix A is very ill conditioned $\rightarrow$ Do not solve $A\,x = b$!

Inverse Problems

Goal: use measured data to *compute* “hidden” information.





Inverse Problems: Regularization is Needed!

The inverse problems of image deblurring and tomographic reconstruction are *ill-posed problems*, i.e, they violate one or more of the Hadamard conditions for a well-posed problem:

- the solution exists,
- the solution is unique,
- the solution is stable with respect to perturbations of data.

$$\text{Algebraic model:} \qquad A\,x = b, \qquad b = A\,x^{\text{exact}} + e.$$

In the algebraic model, the matrix *A* is very ill conditioned, and we do **not** want to compute the "naive solution":

$$x_{\text{LSQ}} = A^{\dagger} b = x^{\text{exact}} + A^{-1} e, \qquad \|A^{-1} e\| \gg \|x^{\text{exact}}\|$$

We must use *regularization* to compute a stable solution.

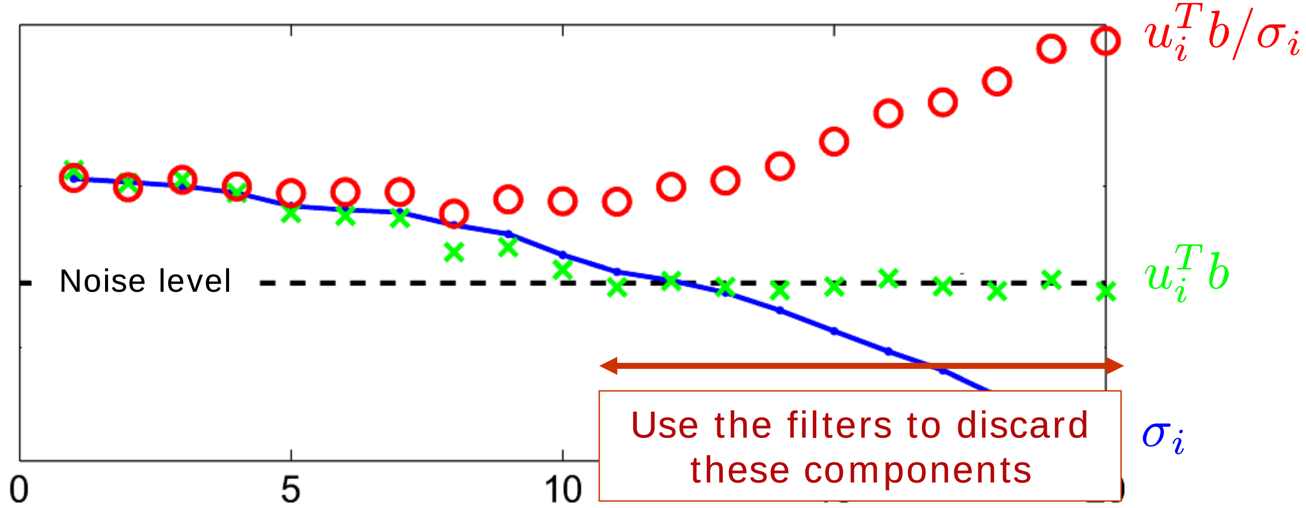
Regularization by Spectral Filtering

We often compute regularized solutions via the following procedure:

1. Choose a (possibly orthonormal) spectral basis: $w_1, w_2, \ldots, w_n$.
2. Write $x_{\mathrm{LSQ}} = A^\dagger b = \sum_{i=1}^n \alpha_i \, w_i$.
3. Introduce spectral filtering: $x_{\mathrm{reg}} = \sum_{i=1}^n \phi_i \, \alpha_i \, w_i$, $\quad \phi_i =$ filter factors.

Example: SVD basis. Given the SVD $A = \sum_{i=1}^n u_i \, \sigma_i \, v_i^T$ we have:

- Spectral basis: $v_1, v_2, \ldots, v_n$. Coefficients for x_{LSQ} are: $\alpha_i = u_i^T b / \sigma_i$.



Some Regularization Methods

Tikhonov regularization:

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

The choice of smoothing norm, together with the choice of λ, forces x to be effectively dominated by components in a low-dimensional subspace, determined by the GSVD of (A, L) – or the SVD of A if $L = I$.

Filter factors $\phi_i = \sigma_i^2/(\sigma_i^2 + \lambda^2)$ when $L = I$.

Regularization by projection:

$$\min_x \|A\,x - b\|_2 \quad \text{subject to} \quad x \in \mathcal{W}_k$$

where $\mathcal{W}_k$ is a k-dimensional subspace, the **signal subspace.**

This works well if "most of" x^{exact} lies in a low-dimensional subspace (sparsity); hence $\mathcal{W}_k$ must be spanned by desirable basis vectors

Think of Truncated SVD: $\mathcal{W}_k = \text{span}\{v_1, v_2, \ldots, v_k\}$, $v_i =$ right singular vectors.

The Projection Method

A more practical formulation of regularization by projection.

We are given the matrix $W_k = (w_1, \ldots, w_k) \in \mathbb{R}^{n\times k}$ such that $\mathcal{W}_k = \mathcal{R}(W_k)$.

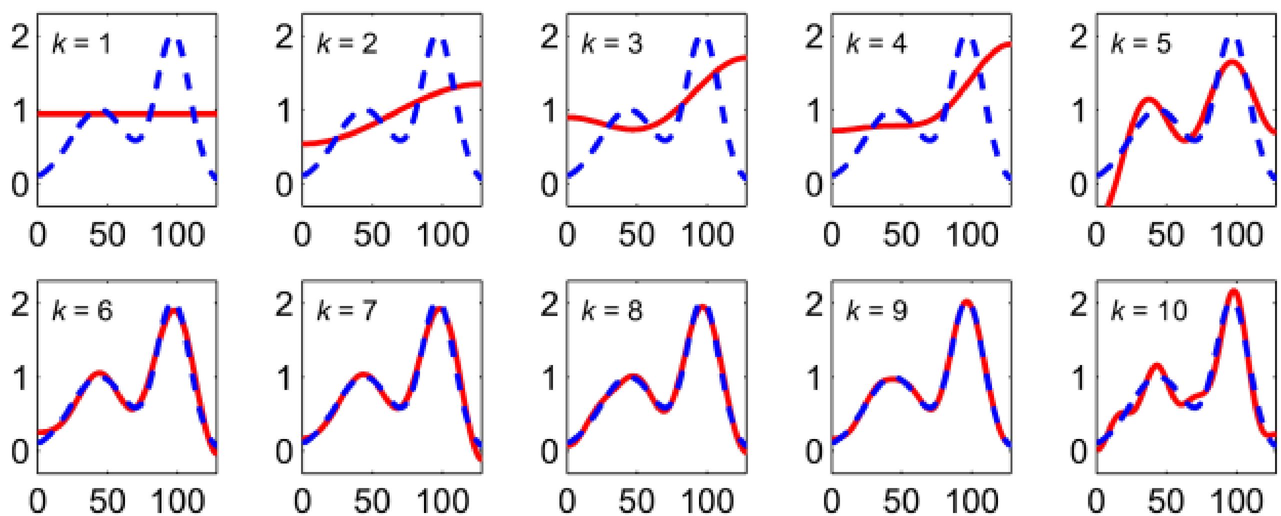
We can write the requirement as $x = W_k\, y$, leading to the formulation

$$x^{(k)} = W_k\, y^{(k)}, \qquad y^{(k)} = \operatorname{argmin}_y \|(A\, W_k)\, y - b\|_2 .$$

Projected problem

Operations often do not require W_k explicitly.

Example:
DCT basis



Some Thought on the Basis Vectors

The DCT basis – and similar bases that define fast transforms:

- computationally convenient (fast) to work with, but
- may not be well suited for the particular problem.

The SVD basis – or GSVD basis if $L \neq I$ – gives an "optimal" basis for representation of the matrix A, but ...

- it is computationally expensive (slow), and
- it does not involve information about the right-hand side b.

Is there a basis that is computationally attractive and also involves information about both A and b, and thus the complete given problem?

$\rightarrow$ Krylov subspaces!

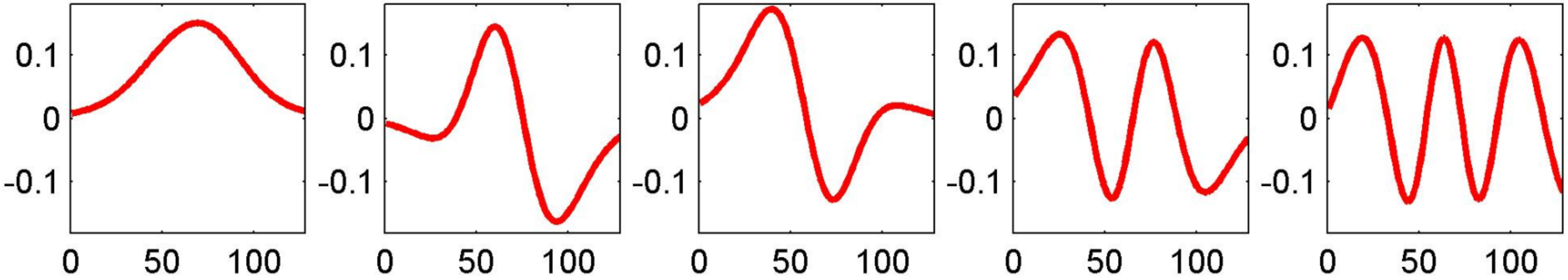
The Krylov Subspace

The Krylov subspace of interest here, defined as

$$\mathcal{K}_k(A^T b, A^T A) \equiv \mathrm{span}\{A^T b, (A^T A)\, A^T b, (A^T A)^2 A^T b, \ldots, (A^T A)^{k-1} A^T b\},$$

always *adapts* itself to the problem at hand!

Orthonormal basis vectors for a certain $\mathcal{K}_5(A^T b, A^T A)$:



A few early references:

- W. Squire, 1976
- G. Nolet, 1985
- A. S. Nemirovskii, 1986
- A. van der Sluis and H. A. van der Vorst, 1990
- M. Hanke, 1995

The Use of CGLS

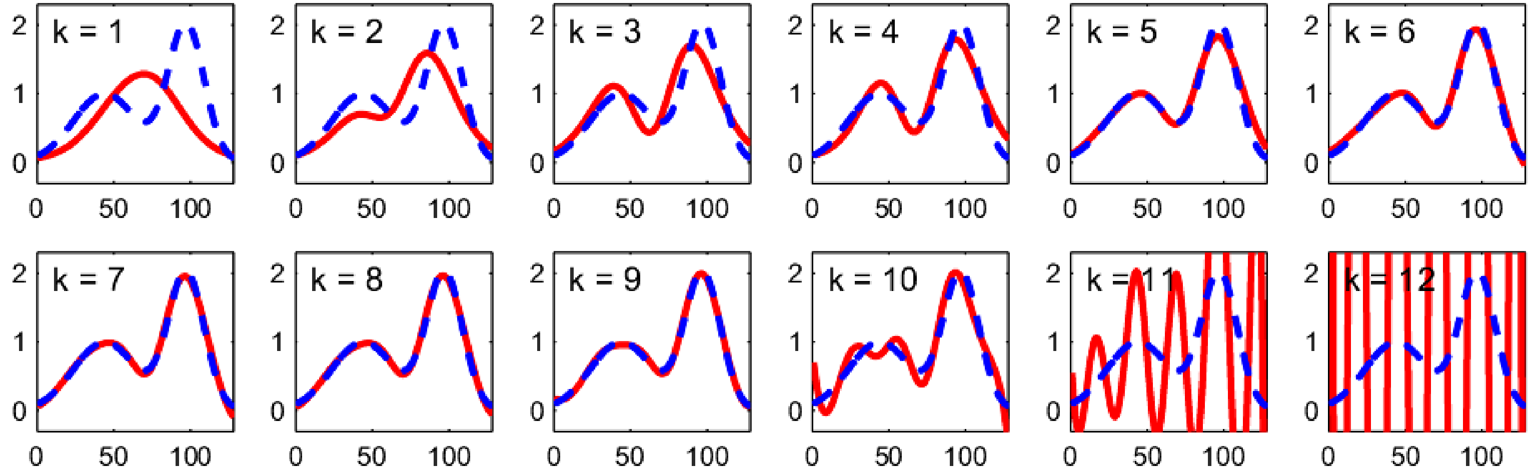
Can we compute $x^{(k)}$ without forming and storing the Krylov basis?

Apply CG to the normal equations for the least squares problem

$$\min \|A\,x - b\|_2 \qquad \Leftrightarrow \qquad A^T A\,x = A^T b\ .$$

This stable stable and efficient implementation of this algorithm is called CGLS, and it produces a sequence of iterates $x^{(k)}$ which solve

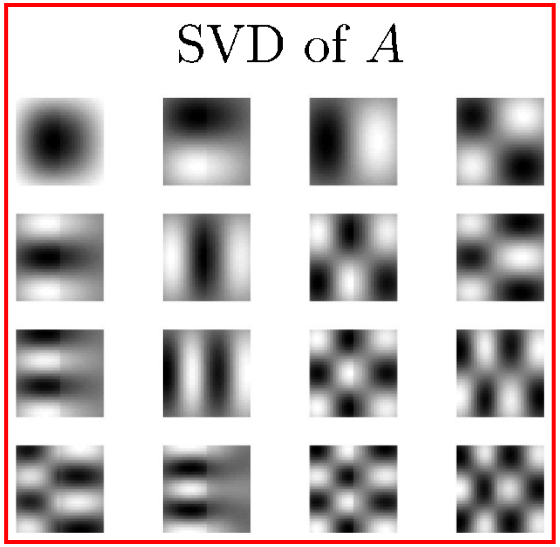
$$\min \|A\,x - b\|_2 \qquad \text{subject to} \qquad x \in \mathcal{K}_k(A^T b, A^T A)\ .$$

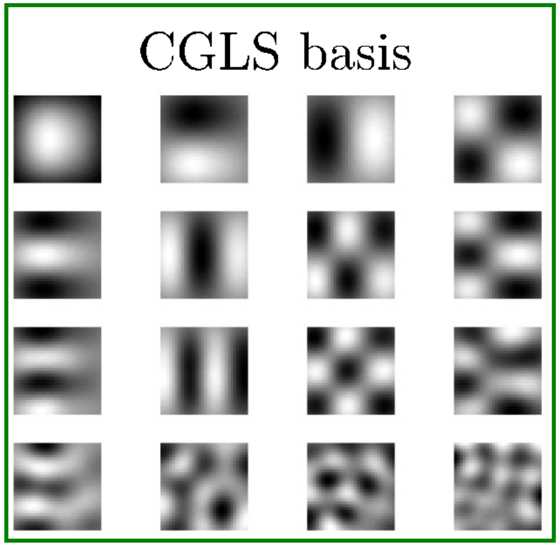


CGLS Basis Vectors for Image Deblurring

Truncated SVD subspace $= \text{span}\{v_1, v_2, v_3, \ldots\}$.

CGLS subspace $= \text{span}\{A^T b, (A^T A) A^T b, (A^T A)^2 A^T b, \ldots\}$.





Two different but *similar* signal subspaces.

Regularizing Iterations

CGLS algorithm solves the problem without forming the Krylov basis explicitly.

This use of CGLS to compute regularized solutions in the Krylov subspace $\mathcal{K}_k$ is referred to as *regularizing iterations*.

Finite precision: convergence slows down, but no deterioration of the solution.

The solution and residual norms are monotone functions of k:

$$\|x^{(k)}\|_2 \geq \|x^{(k-1)}\|_2, \qquad \|A\,x^{(k)} - b\|_2 \;\square\; \|A\,x^{(k-1)} - b\|_2, \qquad k = 1, 2, \ldots$$

Iterative methods are based on multiplications with A and A^T (blurring).

How come repeated blurings can lead to reconstruction?

$\rightarrow$ CGLS constructs a truncated polynomial approximation to $A^\dagger$.

The CGLS Polynomials

CGLS implicitly constructs a polynomial $\mathcal{P}_k$ such that

$$x^{(k)} = \mathcal{P}_k(A^T A) A^T b \; .$$

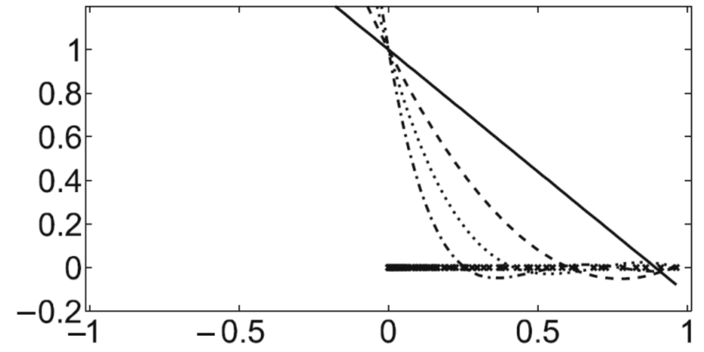
But how is $\mathcal{P}_k$ constructed? Consider the residual

$$\begin{aligned} r^{(k)} &= b - A\,x^{(k)} \;=\; \left(I - A\,\mathcal{P}_k(A^T A)\,A^T\right) b \\ \|r^{(k)}\|_2^2 &= \left\| \left(I - \Sigma\,\mathcal{P}_k(\Sigma^2)\Sigma\right) U^T b \right\|_2^2 \\ &= \sum_{i=1}^n \left(1 - \sigma_i^2 \mathcal{P}_k(\sigma_i^2)\right)^2 (u_i^T b)^2 \;=\; \sum_{i=1}^n \mathcal{Q}_k(\sigma_i^2)(u_i^T b)^2 \end{aligned}$$

To minimize residual norm $\|r^{(k)}\|_2$:

$\rightarrow$ make $\mathcal{Q}_k(\sigma_i^2)$ small where $(u_i^T b)^2$ is large

$\rightarrow$ force $\mathcal{Q}_k(\sigma_i^2)$ to have roots near σ_i^2 that corresp. to large $(u_i^T b)^2$.



Semi-Convergence

During the first iterations, the Krylov subspace $\mathcal{K}_k$ captures the "important" information in the noisy right-hand side b.

- In this phase, the CGLS iterate $x^{(k)}$ approaches the exact solution.

At later stages, the Krylov subspace $\mathcal{K}_k$ starts to capture undesired noise components in b.

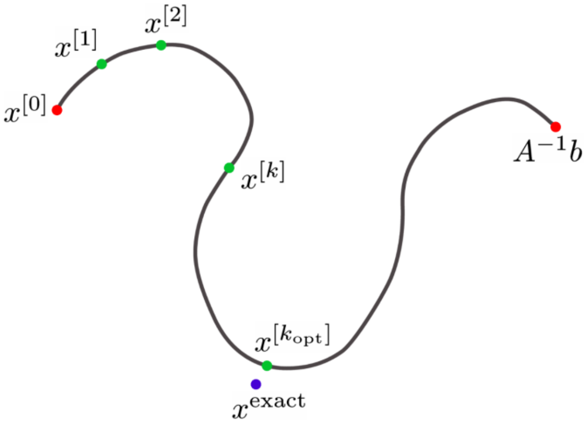
- Now the CGLS iterate $x^{(k)}$ diverges from the exact solution and approach the undesired solution $A^{\dagger}b$ to the least squares problem.

The iteration number k ($=$ the dimension of the Krylov subspace $\mathcal{K}_k$) plays the role of the regularization parameter.

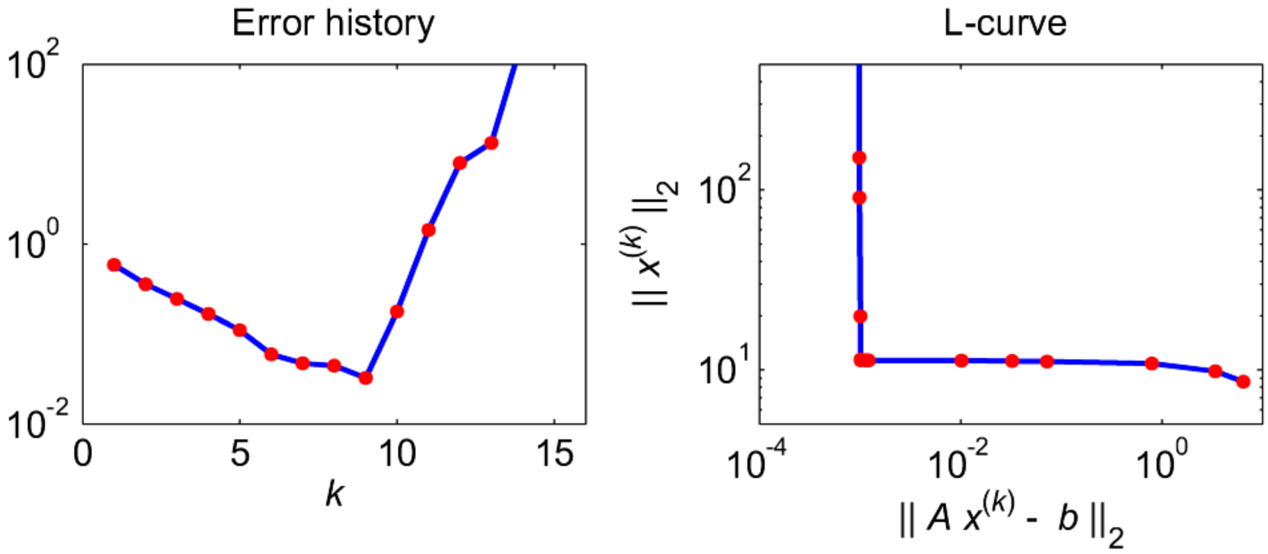
This behavior is called *semi-convergence*.

Illustration of Semi-Convergence

Recall this illustration:



The "ideal" behavior of the error $\| x^{(k)} - x^{\mathrm{exact}} \|_2$ and the associated L-curve:



SVD Perspectives of the Krylov Subspace

Regularization by filtered SVD expansion:

$$x_{\mathrm{reg}} = \textstyle\sum_{i=1}^{n} \phi_i \, \frac{u_i^T b}{\sigma_i} \, v_i \ .$$

Truncated SVD: $\phi_i = 1$ or 0. Tikhonov regularization:

$$\phi_i = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \ .$$

CGLS also corresponds to SVD filtering:

$$x^{(k)} = \textstyle\sum_{i=1}^{n} \phi_i^{(k)} \, \frac{u_i^T b}{\sigma_i} \, v_i , \qquad \phi_i^{(k)} = 1 - \textstyle\prod_{j=1}^{k} \frac{\theta_j^{(k)} - \sigma_i^2}{\theta_j^{(k)}}$$

Here $\theta_j^{(k)}$ are the Ritz values, i.e., the eigenvalues of the projection of $A^T A$ on the Krylov subspace $\mathcal{K}_k$. They converge to those σ_i^2 whose corresponding SVD components $u_i^T b$ are large.

The CGLS Filter Factors

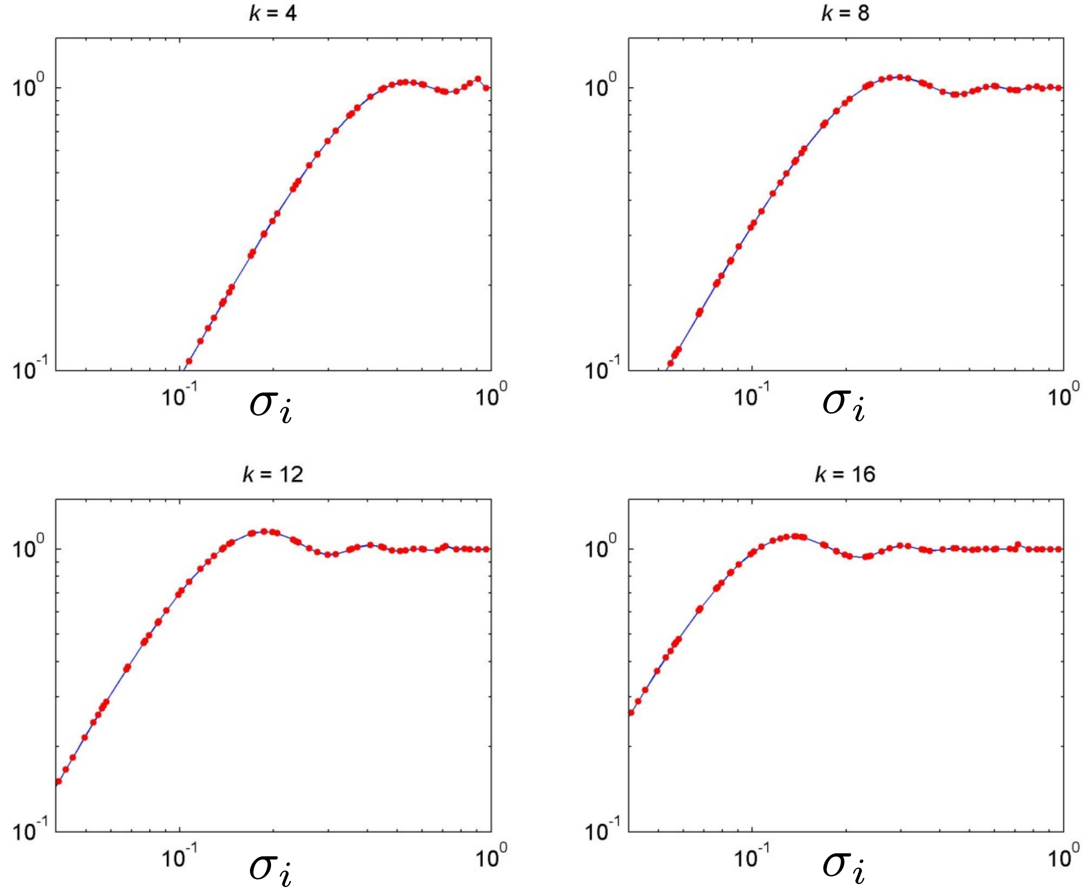
A closer look at the filter factors $\phi_i^{(k)}$ in the filtered SVD expansion

$$x^{(k)} = \sum_{i=1}^{n} \phi_i^{(k)} \, \frac{u_i^T b}{\sigma_i} \, v_i$$

$$= V \, \Phi_k \, \Sigma^\dagger U^T b$$

$$\Phi_k = \mathcal{P}_k(\Sigma^2) \, \Sigma^2$$

Filter factors $\phi_i^{(k)}$



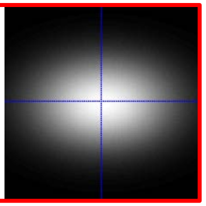
Recall that $\mathcal{P}_k$ is the unique polynomial such that

$$x^{(k)} = \mathcal{P}_k(A^T A) \, A^T b.$$

Other Krylov Subspace Methods

Sometimes it is impractical to use methods that need A^T, e.g, if $A = A^T$ or if we have a black-box function that computes $A\,x$.

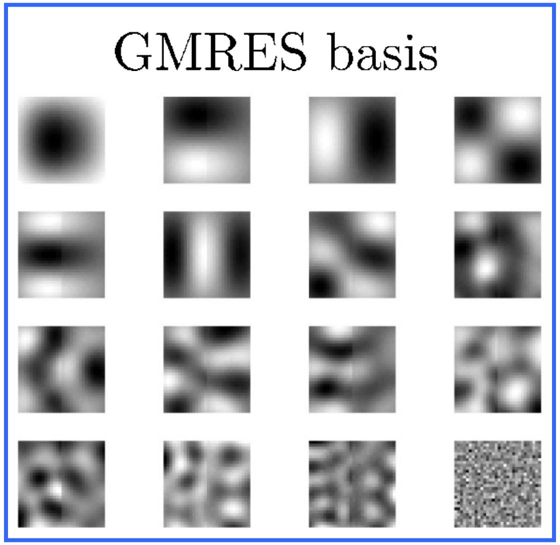
A is symmetric, e.g., if the PSF is "doubly symmetric."



MINRES and GMRES come to mind if the matrix A is square – these methods are based on the Krylov subspace:

$$\mathcal{K}_k = \text{span}\{b, Ab, A^2b, \ldots, A^{k-1}b\}.$$

Unfortunately it is a bad idea to include the noisy vector b in the subspace.



A better choice is the "shiftet" Krylov subspace:

$$\vec{\mathcal{K}}_k = \text{span}\{Ab, A^2b, \ldots, A^kb\}.$$

The corresponding methods are called

- MR-II (Hanke, 1995) and
- RRGMRES (Calvetti, Lewis & Reichel, 2000).

Both are included in REGULARIZATION TOOLS.

Comparing Krylov Methods: MINRES, MR-II

♥ The presence of *b* in the MINRES Krylov subspace gives very noisy solutions.

♠ The absence of *b* in the MR-II Krylov subspace is essential for the noise reduction.

♣ MR-II computes a filtered SVD solution:

$$x^{(k)} = V\,\Phi_k\,\Sigma^\dagger V^T b$$

$$\Phi_k = \mathcal{P}_k(\Omega\,\Sigma)\,\Omega\,\Sigma$$

$$\Lambda = \Omega\,\Sigma, \quad \Omega = \mathrm{diag}(\pm 1)$$

♦ Negative eigenvalues of *A* do not inhibit the regularizing effect of MR-II, but they can slow down the convergence.

Comparing: GMRES, RRGMRES

♥ The presence of b in the GMRES Krylov subspace gives very noisy solutions.

♠ The absence of b in the RRGMRES Krylov subspace is essential for the noise reduction.

♣ RRGMRES *mixes* the SVD components in each iteration and $x^{(k)}$ is not a filtered SVD solution:

$$x^{(k)} = V\,\Phi_k\,\Sigma^{\dagger} U^T b$$

$$\Phi_k = \mathcal{P}_k(C\,\Sigma)\,C\,\Sigma$$

$$C = V^T U$$

♦ RRGMRES works well if the mixing is weak (e.g., if $A \approx A^T$), or if the Krylov basis vectors are well suited for the problem.

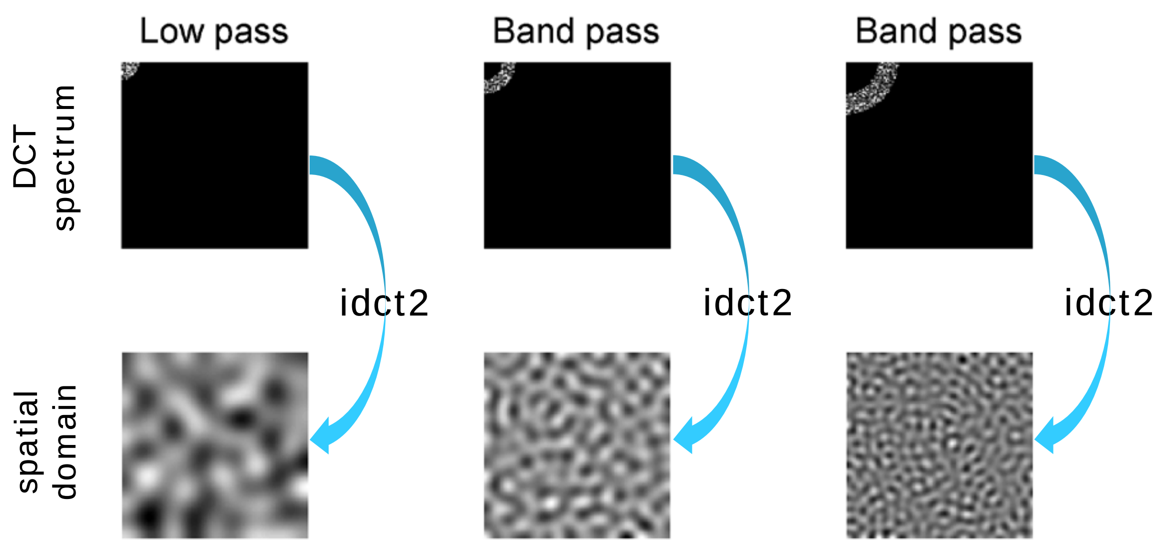
Back to CGLS: The “Freckles”

CGLS:
k = 4, 10
and 25
iterations

Initially, the image gets sharper – then ”freckles” start to appear.

Low frequencies carry the main information.

”Freckles” are band-pass filtered noise.



Noise Propagation

Recall once again that we can write the CGLS solution as:

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

where $\mathcal{P}_k$ is the polynomium associated with the Krylov subspace $\mathcal{K}_k(A^T b, A^T A)$.

Thus $\mathcal{P}_k$ is fixed by A and b, and if $b = b^{\mathrm{exact}} + e$ then

$$x^{(k)} = \mathcal{P}_k(A^T A)\, A^T b^{\mathrm{exact}} + \mathcal{P}_k(A^T A)\, A^T e \equiv \underbrace{x^{(k)}_{b^{\mathrm{exact}}}}_{\text{Signal component}} + \underbrace{x^{(k)}_{e}}_{\text{Noise component}}.$$

Similarly for the other iterative methods.

Note that signal component $x^{(k)}_{b^{\mathrm{exact}}}$ depends on the noise e via $\mathcal{P}_k$.

Signal and Noise Components

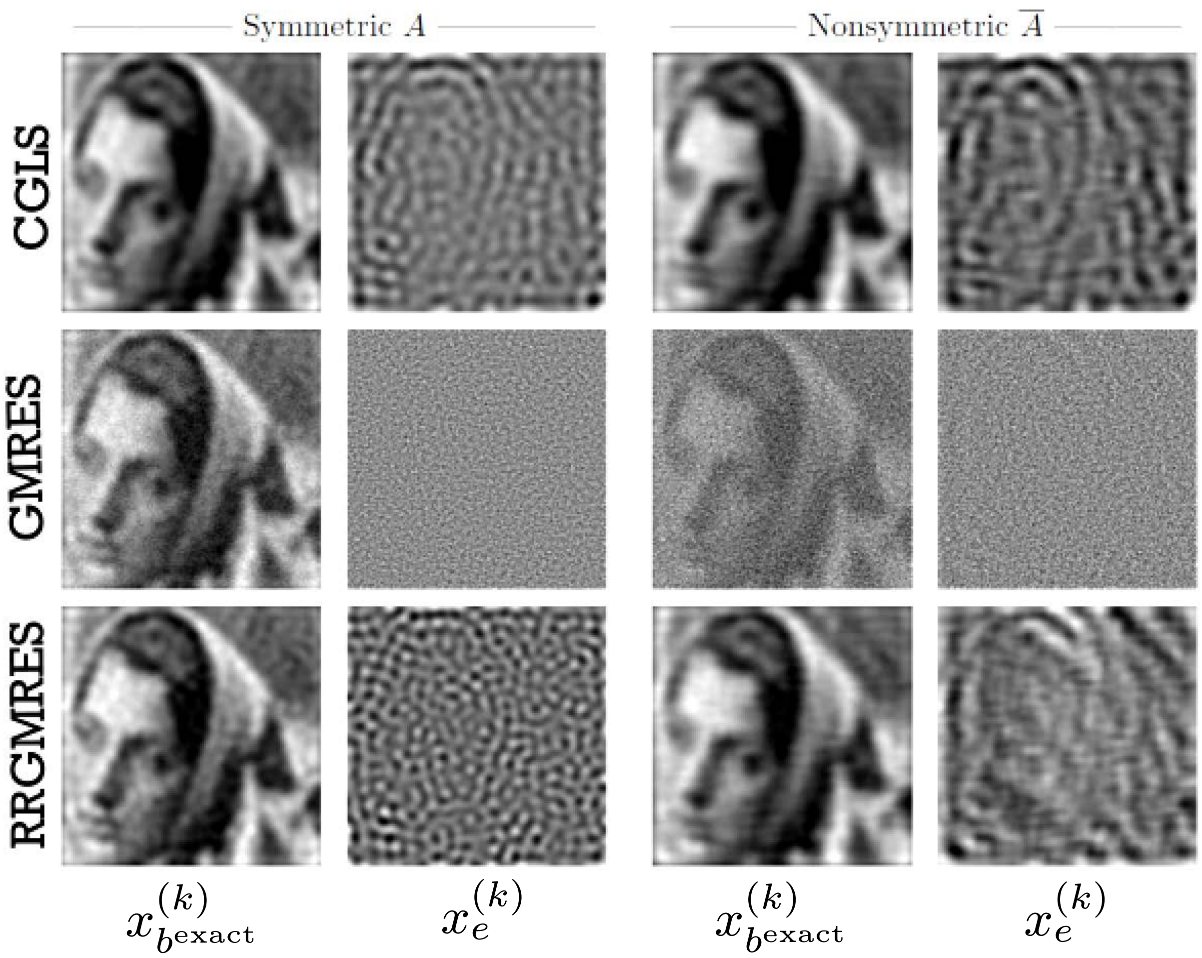
$$x^{(k)} = \mathcal{P}_k(A^T A)\, A^T b = \underbrace{\mathcal{P}_k(A^T A)\, A^T b^{\text{exact}}}_{\text{signal}} + \underbrace{\mathcal{P}_k(A^T A)\, A^T e}_{\text{noise}} \ .$$

Note that the noise components (the freckles) are *correlated* with structures in the image!

Tends to mask the appearance of the noise!!

Two different blurring matrices A

Same Behavior in All Methods



The noise components are always correlated with the image!

Yet Another Krylov Subspace Method

If certain components (or features) are missing from the Krylov subspace, then it makes good sense to *augment* the subspace with these components.

Specifically, for GMRES and RRGMRES, use the subspaces:

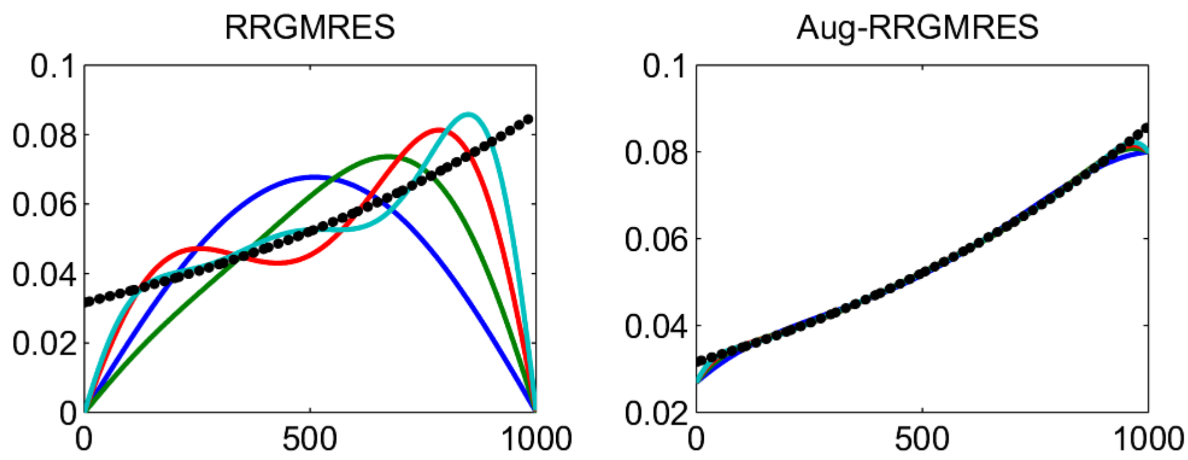
$$\mathcal{S}_k = \mathrm{span}\{w_1, \ldots, w_p\} + \mathrm{span}\{b, Ab, A^2 b, \ldots, A^{k-1} b\}.$$

$$\vec{\mathcal{S}}_k = \mathrm{span}\{w_1, \ldots, w_p\} + \mathrm{span}\{Ab, A^2 b, A^3 b, \ldots, A^k b\}.$$

Example: deriv2.

All vectors in the Krylov subspace $\to$ 0 at the ends.

$w_1 = (1,1,\ldots,1)^T$
$w_2 = (1,2,\ldots,n)^T$



Implementation Aspects, RRGMRES

Baglama & Reichel (2007) proposed algorithm AugRRGMRES that uses the simple formulation

$$A\,W_p = V_p\,H_0 \quad \to \quad A\,\big[\,W_p\,,\,V_k\,\big] = \big[\,V_p\,,\,V_{k+1}\,\big]\,H_k\ .$$

But their algorithm actually solves the problem

$$\min_x \|A\,x-b\|_2^2 \quad \text{s.t.} \quad x \in \mathcal{W}_p + \mathcal{K}_j\big((I-V_pV_p^T)A, (I-V_pV_p^T)Ab\big)\ .$$

Dong, Garde & H recently proposed an alternative algorithm R^3GMRES (Regularized RRGMRES) that uses the desired subspace

$$\mathcal{W}_p + \mathcal{K}_j(A, Ab)\ .$$

Their algorithm is a bit more complicated, but has the same complexity as RRGMRES and AugRRGMRES.

Stopping Rules = Reg. Param. Choice

The classical stopping rule for iterative methods is:

- Stop when the residual norm $\| b - A\, x^{(k)} \|_2$ is ”small.”

It does not work for ill-posed problems: a small residual norm does not imply that $x^{(k)}$ is close to the exact solution!

Must stop when all available information has been extracted from the right-hand side b, just before the noise start to dominate $x^{(k)}$. Some stopping rules:

- discrepancy principle,
- generalized cross validation (GCV),
- L-curve criterion (?),
- normalized cumulative periodogram (NCP),
- and probably several others …

What we covered here

- We understand the inherent regularizing properties of Krylov subspace methods.
- We understand how the noise enters the solutions.
- We can incorporate smoothing conditions/priors via a process similar to preconditioning (not discussed in this talk).
- We can augment the methods to incorporate certain types of priors.

Future work – challenges

- Development of *robust stopping rules*.
- Preconditioning for *acceleration* of the iterations.
- *Flexible* ways of incorporating other *constraints/priors*.
- Nonnegativity constraints?!