Image Deblurring with Krylov Subspace Methods

Per Christian Hansen
Technical University of Denmark

For some details:
see extended abstract.

For more details and references:
see my books.

DTU Informatics
Department of Informatics and Mathematical Modeling
Image deblurring is an inverse problem; hence it is ill posed:
- small perturbation in data →
- large errors in reconstruction

This talk:
- Regularization
- Projection
- CGLS
- Other iterations
- Noise propagation
- Subspace precond.
Sources of Blurred Images
The Deblurring Problem

Fredholm integral equation of the first kind:

\[ \int_0^1 \int_0^1 K(x, y; x', y') f(x, y) \, dx \, dy = g(x', y') , \quad 0 \leq x', y' \leq 1. \]

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:
- Out of focus
- Motion
- Gaussian

Discretization yields a LARGE system of linear equations: \( A x = b \).
But the matrix \( A \) is very ill conditioned, and therefore

Do not solve \( A x = b \)!
Regularize!

We must apply regularization in order to deal with the ill conditioning of the problem and suppress the influence of the noise in the data.

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 \( \lambda \), 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 \( L = I \).

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 – works well if this subspace is spanned by desirable basis vectors (think of TSVD: \( \mathcal{W}_k = \text{span}\{v_1, v_2, \ldots, v_k\} \)).
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

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

Example: DCT basis

- Operations often do not require $W_k$ explicitly.

- Projected problem
Some Thought on the Basis Vectors

The DCT basis (and similar bases that define fast transforms):

- computationally convenient 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, and
- it does not involve information about the rhs $b$.

Is there a basis that is computationally attractive and also involves information about $b$ and thus the given problem?

→ Krylov subspaces!
Regularizing Iterations

Apply CG to the normal equations for the least squares problem

$$\min \|A x - b\|_2.$$  

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

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

where $\mathcal{K}_k$ is the $k$-dimensional Krylov subspace

$$\mathcal{K}_k = \text{span}\{A^T b, A^T A A^T b, (A^T A)^2 A^T b, \ldots\}.$$  

These methods are referred to as regularizing iterations.

Iterative methods are based on multiplications with $A$ and $A^T$ (blurring).
How come repeated blurrings can lead to reconstruction?

$\rightarrow$ CGLS constructs a polynomial approximation to $(A^T A + \lambda^2 I)^{-1} A^T$. 
The Behavior of CGLS

CGLS algorithm solves the problem without forming the Krylov basis explicitly. Finite precision: convergence slows down, but no deterioration of the solution. The solution and residual norms are monotone functions of $k$:

$$\|x^{(k)}\|_2 \geq \|x^{(k-1)}\|_2, \quad \|Ax^{(k)} - b\|_2 \leq \|Ax^{(k-1)} - b\|_2, \quad k = 1, 2, \ldots$$

Same example as before: CGLS iterates
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

$x^{(0)} \rightarrow x^{(4)} \rightarrow x^{(10)} \rightarrow x^{(50)} \rightarrow x^{(100)} \rightarrow x^{(200)} \rightarrow A^\dagger b$
Advantages of the Krylov Subspace

The SVD basis vectors $v_1, v_2, \ldots$ are well suited for representation of $A$.

But this basis “does not know” about the given problem – it can not utilize information about the right-hand side $b$.

The Krylov subspace $\mathcal{K}_k$ “knows” about the right-hand side and therefore adapts itself to the given problem, through the starting vector

$$A^T b = A^T A x^{\text{exact}} + A^T e = \sum_{i=1}^n \sigma_i^2 (v_i^T x^{\text{exact}}) v_i + \sum_{i=1}^n \sigma_i (u_i^T e) v_i.$$ 

Hence the Krylov basis vectors are rich in those directions that are needed.

---

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

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 $\sigma_i^2$ whose corresponding SVD components $u_i^T b$ are large.
CGLS Focuses on Significant Components

Example: **phillips** (from Regularization Tools).
Exact solution has many zero SVD coefficients.
- TSVD solution $x_k$ includes all coef. from 1 thru $k$.
- CGLS solution $x^{(k)}$ includes only those coef. we need.

CGLS suppresses noise better than TSVD in this case.
Another Story: CGLS for Tikhonov

Of course, one could also use CGLS to solve the Tikhonov problem in the form

$$\min_x \left\| \begin{pmatrix} A \\ \lambda L \end{pmatrix} - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2^2.$$  

But this approach typically requires that the system is solved many times, for many different values of $\lambda$.

Also, preconditioning is often necessary – but it can be difficult to design a good preconditioner for the Tikhonov problem.

We shall not pursue this aspect further in this talk.
Other Krylov Subspace Methods

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

MINRES and GMRES come to mind – these methods are based on the Krylov subspace:

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

Unfortunately it is a bad idea to include the noisy vector $b$ in the subspace. A better choice is the “shifted” Krylov subspace:

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

The corresponding methods are called MR-II and RRGMRES (they are now included in Regularization Tools).

Example on next slide.
We find:

♥ The absence of $b$ in the Krylov subspace is essential for MR-II.

♦ MR-II computes a filtered SVD solution.

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

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

♣ 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.
Progress of the Iterations

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

CGLS solution can be written in terms of a matrix polynomial:

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

where \( \mathcal{P}_k(\theta) = (1 - \mathcal{R}_k(\theta))/\theta \) and \( \mathcal{R}_k \) is the Ritz polynomial associated with the Krylov subspace \( \mathcal{K}_k \).

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

\[ x^{(k)} = \mathcal{P}_k(A^T A) A^T b^{\text{exact}} + \mathcal{P}_k(A^T A) A^T e \equiv x^{(k)}_{b^{\text{exact}}} + x^{(k)}_e. \]

Similarly for the other iterative methods.

Note that signal component \( x^{(k)}_{b^{\text{exact}}} \) depends on the noise \( e \) via \( \mathcal{P}_k \).
Signal and Noise Components

\[ x^{(k)} = \mathcal{P}_k(A^T A) A^T b = \mathcal{P}_k(A^T A) A^T b^{\text{exact}} + \mathcal{P}_k(A^T A) A^T e. \]

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

Two different matrices \( A \)

Tends to mask the appearance of the noise!!
Same Behavior in All Methods

The noise components are always correlated with the image!
General-Form Tikhonov Regularization

CGLS is linked to the SVD of $A$ and thru the Krylov subspace, the Ritz polynomial, and the convergence of the Ritz values.

Thus CGLS is also related to Tikhonov regularization in standard form

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

But occasionally we prefer the *general* formulation

$$\min_x \left\{ \|Ax - b\|_2^2 + \lambda^2 \|Lx\|_2^2 \right\}, \quad L \neq I.$$

How do we modify CGLS such that it can incorporate the matrix $L$?

Use the *standard-form transformation*

$$\min_{\tilde{x}} \|\overline{A} \tilde{x} - b\|_2^2 + \lambda^2 \|\tilde{x}\|_2^2 \quad \text{with} \quad \overline{A} = AL^\# \quad \text{and} \quad x = L^\# \tilde{x} + x_N,$$

where $L^\# = \text{oblique pseudoinverse of } L$ and $x_N \in \mathcal{N}(L)$.  

Subspace Preconditioning

If we apply CGLS to the standard-form problem

$$\min_{\bar{x}} \| \bar{A} \bar{x} - b \|_2^2 + \lambda^2 \| \bar{x} \|_2^2,$$

then the iterates, when transformed back via $L^\#, \text{ lie in the affine space}$

$$\text{span}\{MA^T b, (MA^T A) MA^T b, (MA^T A)^2 MA^T b, \ldots\} + x_N,$$

where $M = L^\#(L^\#)^T$.

Hence $L$ is a preconditioner for CGLS that provides a better suited subspace.

The Krylov subspace methods are implemented such that $\bar{A}$ is never formed.

How is the oblique pseudoinverse $L^\#$ defined? And why this particular matrix?

Outside scope of this talk.

Next slide please ...
Write $x = x_\mathcal{M} + x_\mathcal{N}$ with $x_\mathcal{N} \in \mathcal{N}(L)$ and $x_\mathcal{M}$ being $A^T A$–orthogonal to $x_\mathcal{N}$. This corresponds to an oblique splitting of the subspace $\mathbb{R}^n$.

Then the vector $A x = A x_\mathcal{M} + A x_\mathcal{N}$ splits into two orthogonal components.

The Tikhonov problem reduces to two independent problems for $x_\mathcal{M}$ and $x_\mathcal{N}$:

$$\min \|A x_\mathcal{M} - b\|_2^2 + \lambda^2 \|x_\mathcal{M}\|_2^2 \quad \text{and} \quad \min \|A x_\mathcal{N} - b\|_2^2.$$

Since $x_\mathcal{M} = L^\# L x$ we get $A x_\mathcal{M} = (A L^\#) (L x) \to$ the standard-form problem.
Conclusion

An important area with plenty of
• theoretical aspects,
• computational challenges, and
• important applications.

More stuff not covered here:
• Boundary conditions
• Stopping criteria
• Hybrid methods: projection + regularization
• Nonnegativity constraints
• Other iterative methods: ART, SIRT, Richardson-Lucy, ...
• Blind deconvolution
• Applications in astronomy, biometrics, computer-vision, ...