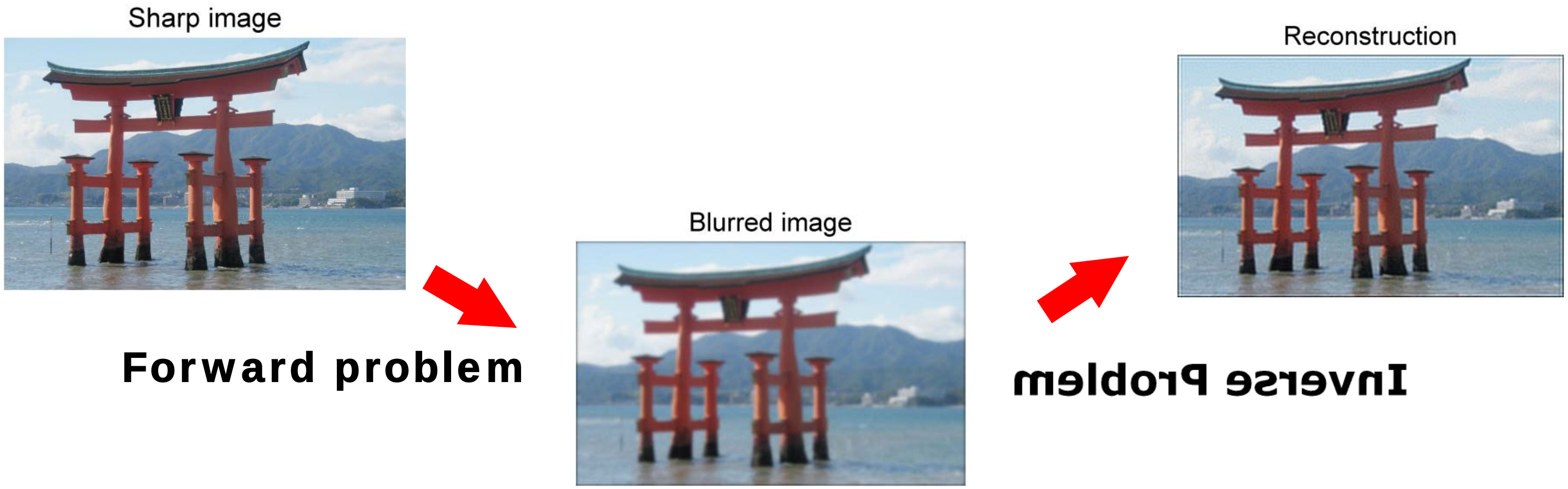
Tutorial:
Algebraic Iterative Reconstruction

Per Christian Hansen

Technical University of Denmark

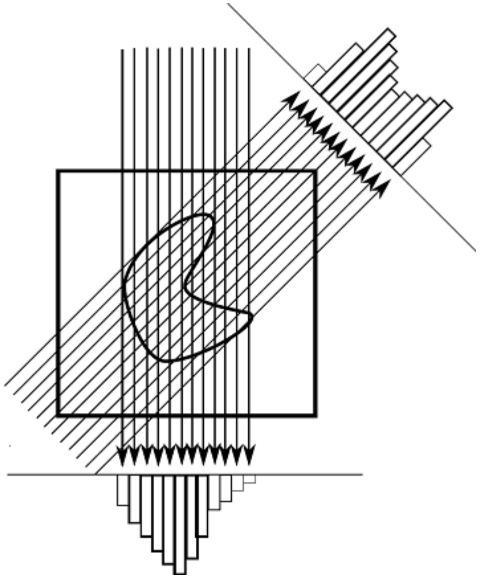
Overview of Talk



- Classical methods: filtered back projection etc.
- An alternative: the algebraic formulation
- Iterative methods (e.g., ART, SIRT, CGLS)
- Semi-convergence
- AIR Tools – a new MATLAB® package
- Examples

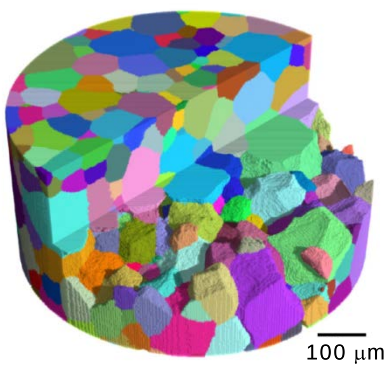
Tomographic Imaging

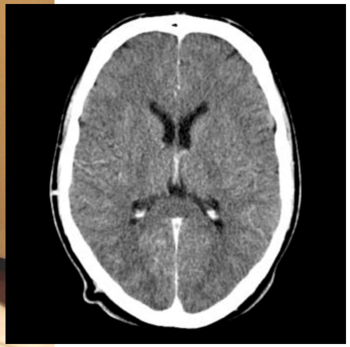
Image reconstruction from projections



Mapping of materials

Medical scanning





The Origin of Tomography

Johan Radon, *Über die Bestimmung von Funktionen durch ihre Integralwerte Längs gewisser Mannings-faltigkeiten*, Berichte Sächsische Akadamie der Wissenschaften, Leipzig, Math.-Phys. Kl., 69, pp. 262-277, 1917.

<u>Main result:</u>

An object can be perfectly reconstructed from a full set of projections.

NOBELFÖRSAMLINGEN KAROLINSKA INSTITUTET

THE NOBEL ASSEMBLY AT THE KAROLINSKA INSTITUTE

11 October 1979

The Nobel Assembly of Karolinska Institutet has decided today to award the Nobel Prize in Physiology or Medicine for 1979 jointly to

Allan M Cormack and Godfrey Newbold Hounsfield

for the "development of computer assisted tomography".

Some Reconstruction Algorithms

Transform-Based Methods

The forward problem is formulated as a certain transform
$\rightarrow$ formulate a stable way to compute the inverse transform.

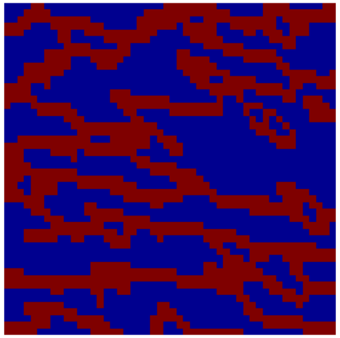
Examples: inverse Radon transform, filtered back projection.

Algebraic Iterative Methods

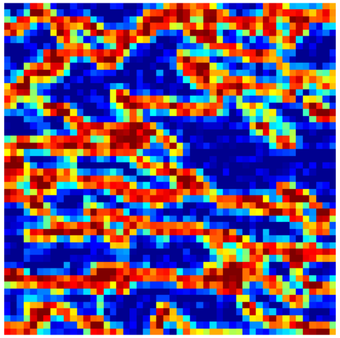
Write the forward problem as an algebraic model $A\,x = b$
$\rightarrow$ reconstruction amounts to solving $A\,x = b$ iteratively.

Examples: ART, Landweber, Cimmino, conjugate gradients.

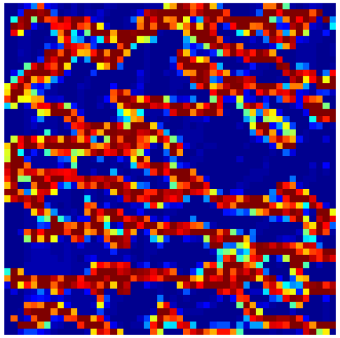
'Phantom'



Filtered back projection



ART reconstruction



Filtered Back Projection

The steps of the inverse Radon transform:

Choose a filter: $\mathcal{F}(\omega) = |\omega| \cdot \mathcal{F}_{\mathrm{low-pass}}(\omega)$.

Apply filter for each angle ϕ in the sinogram: $G_\phi(\rho) = \mathrm{ifft}\big(\mathcal{F} \cdot \mathrm{fft}(g_\phi)\big)$.

Back projection to image: $f(x,y) = \int_0^{2\pi} G_\phi(x\cos\phi + y\sin\phi)\, d\phi$.

Interpolation to go from polar to rectanglar coordinates (pixels).

Advantages

- Fast because it relies on FFT computations!
- Low memory requirements.
- Lots of experience with this method from many years of use.

Drawbacks

- Needs many projections for accurate reconstructions.
- Difficult to apply to non-uniform distributions of rays.
- Filtering is “hard wired” into the algorithm (low-pass filter).
- Difficult to incorporate prior information about the solution.

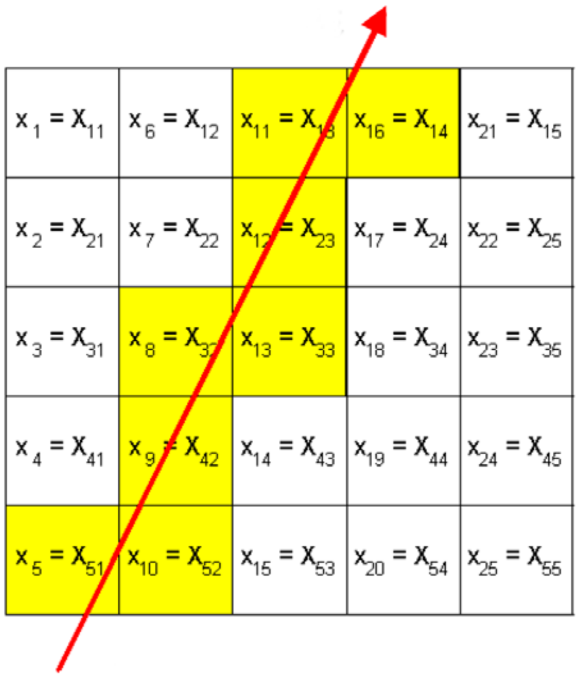
Setting Up the Algebraic Model

Damping of *i*-th X-ray through domain:

$$b_i = \int_{\mathrm{ray}_i} \chi(\mathbf{s})\, d\ell, \qquad \chi(\mathbf{s}) = \text{attenuation coef.}$$

Assume $\chi(\mathbf{s})$ is a constant x_j in pixel j, leading to:

$$b_i = \sum_j a_{ij}\, x_j, \qquad a_{ij} = \text{length of ray } i \text{ in pixel } j.$$



This leads to a large, sparse system:

$$A\,x = b \qquad b = b^* + e$$

"Geometry"

Image

Projections

Noise

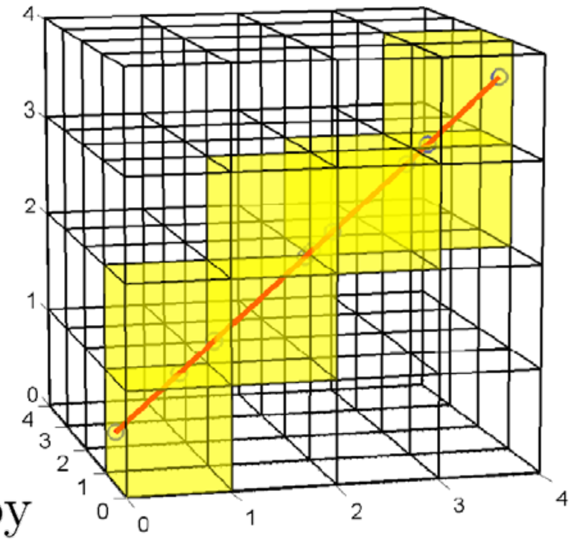
$$b^* = A\,x^*$$

More About the Coefficient Matrix, 3D Case

$$b_i = \sum_j a_{ij}\, x_j, \qquad a_{ij} = \text{length of ray } i \text{ in voxel } j.$$

To compute the matrix element a_{ij} we simply need to know the intersection of ray i with voxel j. Let ray i be given by the line

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + t \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}, \qquad t \in \mathbb{R}.$$



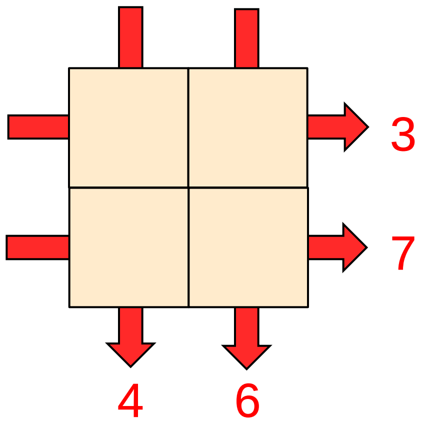
The intersection with the plane $x = p$ is given by

$$\begin{pmatrix} y_j \\ z_j \end{pmatrix} = \begin{pmatrix} y_0 \\ z_0 \end{pmatrix} + \frac{p - x_0}{\alpha} \begin{pmatrix} \beta \\ \gamma \end{pmatrix}, \qquad p = 0, 1, 2, \ldots$$

with similar equations for the planes $y = y_j$ and $z = z_j$.

From these intersetions it is easy to compute the ray length in voxel j.

Example: the “Sudoku” Problem



$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 3 \\ 7 \\ 4 \\ 6 \end{pmatrix}$$

Infinitely many solutions ($k \in \Re$):

$$\begin{array}{|c|c|}\hline & \\ \hline & \\ \hline \end{array} = \begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array} + k \times \begin{array}{|c|c|}\hline -1 & 1 \\ \hline 1 & -1 \\ \hline \end{array}$$

0	3
4	3

1	2
3	4

2	1
2	5

Prior: solution is integer and non-negative

3	0
1	6

Some Row Action Methods

ART – Algebraic Reconstruction Techniques

- Kaczmarz's method + variants.
- *Sequential* row-action methods that update the solution using one row of *A* at a time.
- Good semiconvergence observed, but lack of underlying theory of this important phenomenon.

SIRT – Simultaneous Iterative Reconstruction Techniques

- Landweber, Cimmino, CAV, DROP, SART, ...
- These methods use all the rows of *A simultaneously* in one iteration (i.e., they are based on matrix multiplications).
- Slower semiconcergence, but otherwise good understanding of convergence theory.

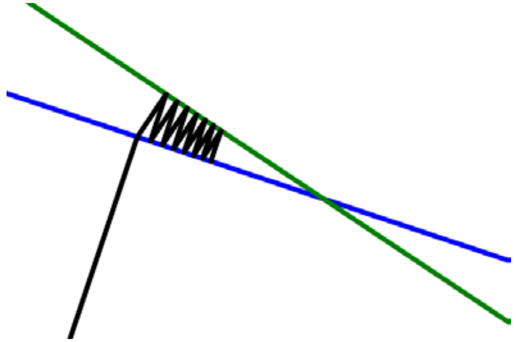
Krylov subspace methods

- CGLS, LSQR, GMRES, ...
- These methods are also based on matrix multiplications

ART Methods

The typical step in these methods involves the ith row a_i of A in the following update of the iteration vector:

$$x \leftarrow x + \lambda \frac{b_i - \langle a_i, x \rangle}{\|a_i\|_2^2} a_i,$$



where λ is a relaxation parameter.

Different sweeps:

Kaczmarz: $i = 1, 2, \ldots, m, \qquad m =$ no. of rows.

Symmetric Kaczmarz: $i = 1, 2, \ldots, m-1, m, m-1, \ldots, 3, 2.$

Randomized Kaczmarz: select row i randomly with probability proportional to the row norm $\|a_i\|_2$.

Nonnegativity and Box Constraints

Easy to inporporate a projection $\mathcal{P}$ corresponding to nonnegativity constraints ($x \geq 0$) or box constraints ($a \leq x \leq b$):

$$x \leftarrow \mathcal{P}\left(x + \lambda \frac{b_i - \langle a_i, x\rangle}{\|a_i\|_2^2} a_i\right).$$

Matlab: `x(x<0) = 0;`

The Steepest Descent Method

Consider the least squares problem:

$$\min_x \|A\,x - b\|_2^2 \qquad \Leftrightarrow \qquad A^T A\,x = A^T b\ .$$

The gradient for $f(x) = \|A\,x - b\|_2^2$ is $\nabla f(x) = A^T(A\,x - b)$.

The steepest-descent method involves the steps:

$$x^{k+1} = x^k + \lambda_k\, A^T(b - A\,x^k), \qquad k = 0, 1, 2, \ldots$$

With $\mathcal{P}$ this becomes the gradient projection algorithm:

$$x^{k+1} = \mathcal{P}\big(x^k + \lambda_k\, A^T(b - A\,x^k)\big)\,, \qquad k = 0, 1, 2, \ldots$$

The SIRT methods are based on this approach. ➡ Next page

SIRT Methods

Diagonally Relaxed Orthogonal Projection

Simultaneous Algebraic Reconstruction Technique

The general form:

$$x^{k+1} = x^k + \lambda_k T A^T M (b - A\, x^k), \qquad k = 0, 1, 2, \ldots$$

Some methods use the row norms $\|a_i\|_2$.

Landweber: $T = I$ and $M = I$.

Cimmino: $T = I$ and $M = D = \frac{1}{m}\mathrm{diag}\left(\frac{1}{\|a_i\|_2^2}\right)$.

CAV (component averaging method): $T = I$ and $M = D_S = \mathrm{diag}\left(\frac{1}{\|a_i\|_S^2}\right)$ with $S = \mathrm{diag}(\mathbf{nnz}(\text{column } j))$.

DROP: $T = S^{-1}$ and $M = mD$.

SART: $T = \mathrm{diag}(\text{row sums})^{-1}$ and $M = \mathrm{diag}(\text{column sums})^{-1}$.

Krylov Subspace Methods

In spite of their fast convergence for some problems, these methods are less known in the tomography community.

The most important method is CGLS, obtained by applying the classical Conjugate Gradient method to the least squares problem:

$$
\begin{aligned}
& x^{(0)} = 0 \quad \text{(starting vector)} \\
& r^{(0)} = b - A\, x^{(0)} \\
& d^{(0)} = A^T r^{(0)} \\
& \text{for } k = 1, 2, \ldots \\
& \qquad \bar{\alpha}_k = \| A^T r^{(k-1)} \|_2^2 / \| A\, d^{(k-1)} \|_2^2 \\
& \qquad x^{(k)} = x^{(k-1)} + \bar{\alpha}_k\, d^{(k-1)} \\
& \qquad r^{(k)} = r^{(k-1)} - \bar{\alpha}_k\, A\, d^{(k-1)} \\
& \qquad \bar{\beta}_k = \| A^T r^{(k)} \|_2^2 / \| A^T r^{(k-1)} \|_2^2 \\
& \qquad d^{(k)} = A^T r^{(k)} + \bar{\beta}_k\, d^{(k-1)} \\
& \text{end}
\end{aligned}
$$

The work:

One mult. with *A*

One mult. with A^T

Convergence of the Iterative Methods

Assume that the solution is *smooth*, as controlled by a parameter $\alpha > 0$,

$$u_i^T b = \sigma_i^{1+\alpha}, \qquad i = 1, \ldots, n,$$

and that the right-hand side has no errors/noise.

Then the iterates x^k converge to an exact solution $x^* \in \mathcal{R}(A^T)$ as follows.

ART and SIRT methods:

$$\|x^k - x^*\|_2 = \mathcal{O}(k^{-\alpha/2}), \qquad k = 0, 1, 2, \ldots$$

CGLS:

$$\|x^k - x^*\|_2 = \mathcal{O}(k^{-\alpha}), \qquad k = 0, 1, 2, \ldots$$

The interesting case is when errors/noise is present in the right-hand side!

Semi-Convergence of the Iterative Methods

Noise model: $b = A\,x^* + e$, where x^* = exact solution, e = additive noise.

Throughout all the iterations, the residual norm $\|A\,x^k - b\|_2$ decreases as the iterates x^k converge to the least squares solution x_{LS}.

But x_{LS} is dominated by errors from the noisy right-hand side b!

However, during the first iterations, the iterates x^k capture "important" information in b, associated with the exact data $b^* = A\,x^*$.

- In this phase, the iterates x^k approach the exact solution x^*.

At later stages, the iterates starts to capture undesired noise components.

- Now the iterates x^k diverge from the exact solution and they approach the undesiredleast squares solution x_{LS}.

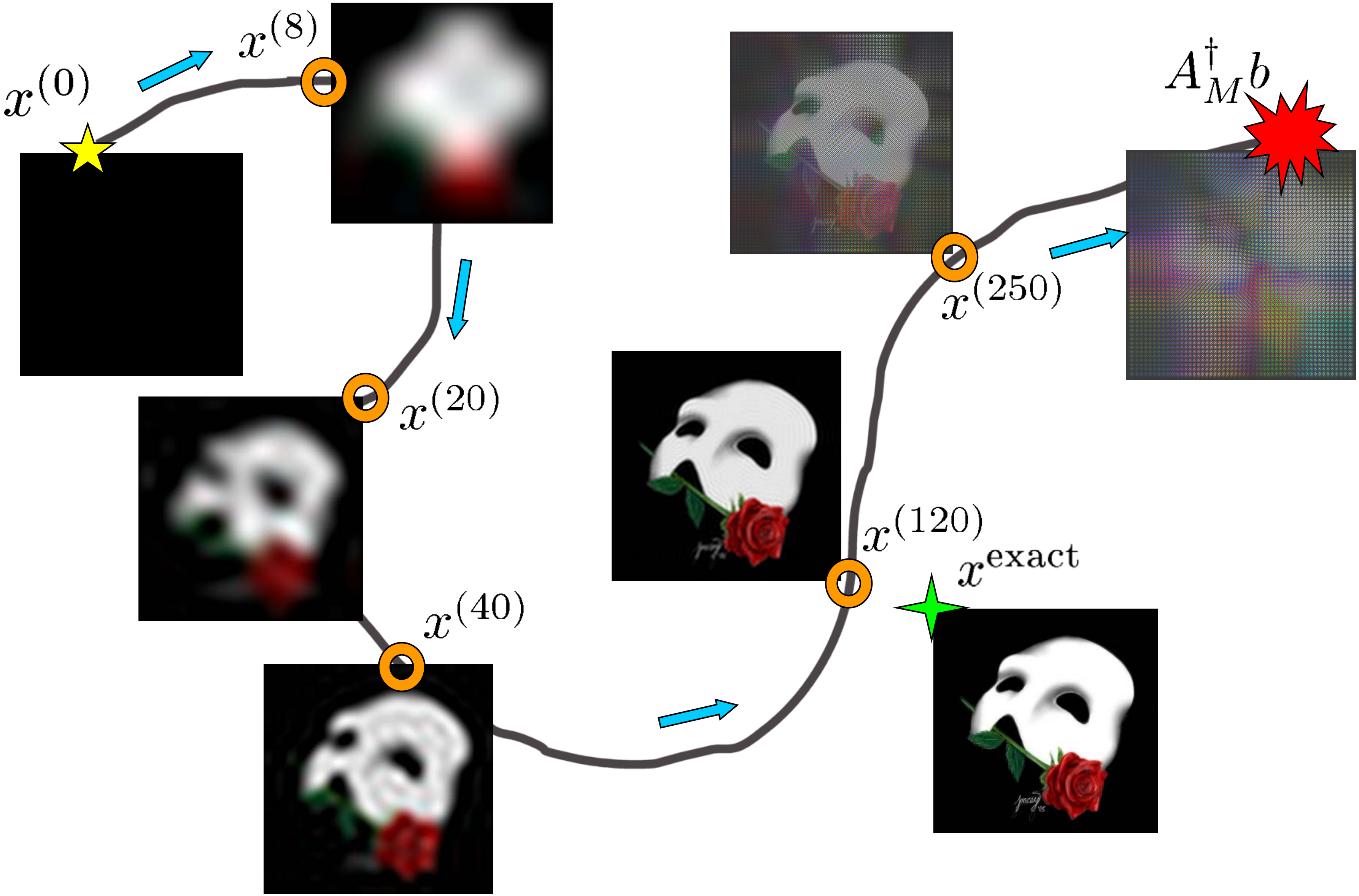
This behavior is called *semi-convergence*, a term coined by Natterer (1986).

"... even if [the iterative method] provides a satisfactory solution after a certain number of iterations, it deteriorates if the iteration goes on."

Many Studies of Semi-Convergence

- G. Nolet, *Solving or resolving inadequate and noisy tomographic systems* (1985)
- A. S. Nemirovskii, *The regularizing properties of the adjoint gradient method in ill-posed problems* (1986)
- F. Natterer, *The Mathematics of Computerized Tomography* (1986)
- Brakhage, *On ill-posed problems and the method of conjugate gradients* (1987).
- C. R. Vogel, *Solving ill-conditioned linear systems using the conjugate gradient method* (1887)
- A. van der Sluis & H. van der Vorst, *SIRT- and CG-type methods for the iterative solution of sparse linear least-squares problems* (1990)
- M. Bertero & P. Boccacci, *Inverse Problems in Imaging* (1998)
- M. Kilmer & G. W. Stewart, *Iterative regularization and MINRES* (1999)
- H. W. Engl, M. Hanke & A. Neubauer, *Regularization of Inverse Problems* (2000)

Illustration of Semi-Convergence

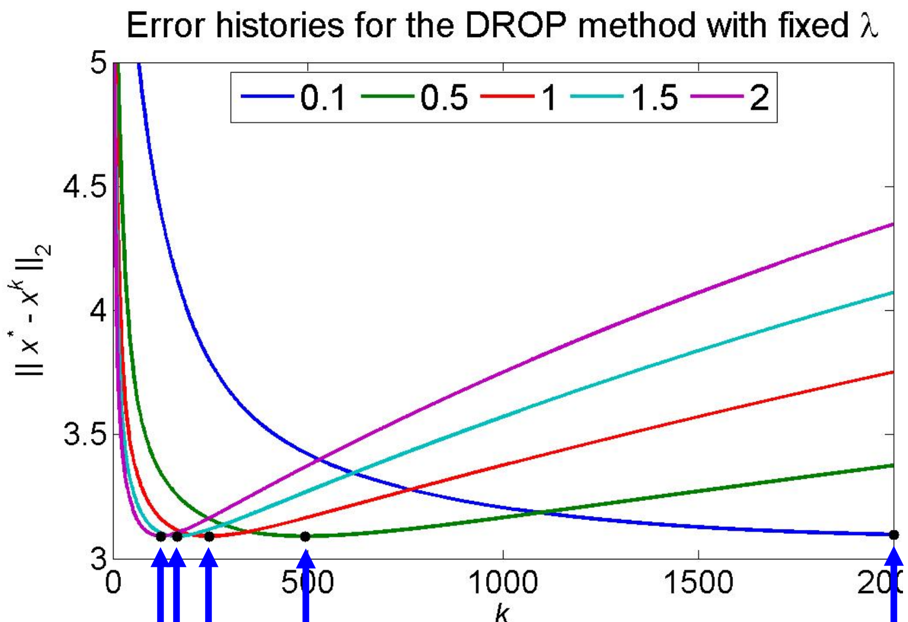


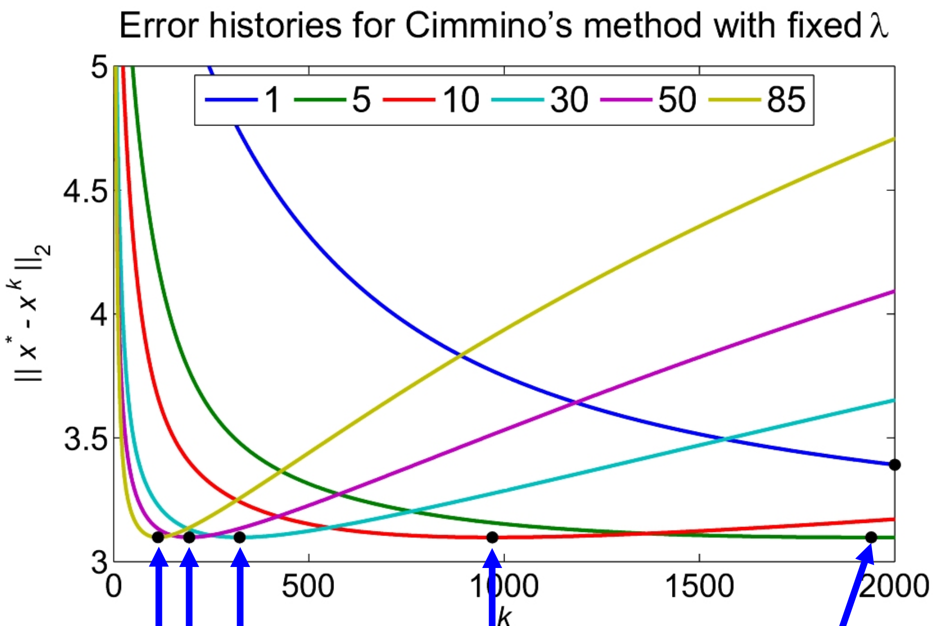
Another Look at Semi-Convergence

Notation: $b = A\,x^* + e$, x^* = exact solution, e = noise.

Initial iterations: the error $\|x^* - x^k\|_2$ decreases.

Later: the error increases as $x^k \to \operatorname{argmin}_x \|A\,x - b\|_M$.





The <u>minimum error</u> is *independent* of both λ and the method.

Analysis of Semi-Convergence

Consider the SIRT methods with $T = I$ and the SVD:

$$M^{1/2} A = U\,\Sigma\,V^T = \textstyle\sum_{i=1}^n u_i\,\sigma_i\,v_i^T .$$

Then x^k is a filtered SVD solution:

$$x^k = \textstyle\sum_{i=1}^n \varphi_i^{[k]}\,\frac{u_i^T (M^{\frac{1}{2}}\,b)}{\sigma_i}\,v_i, \qquad \varphi_i^{[k]} = 1 - \left(1 - \lambda\,\sigma_i^2\right)^k .$$

Recall that we solve *noisy* systems $A\,x = b$ with $b = A\,x^* + e$.

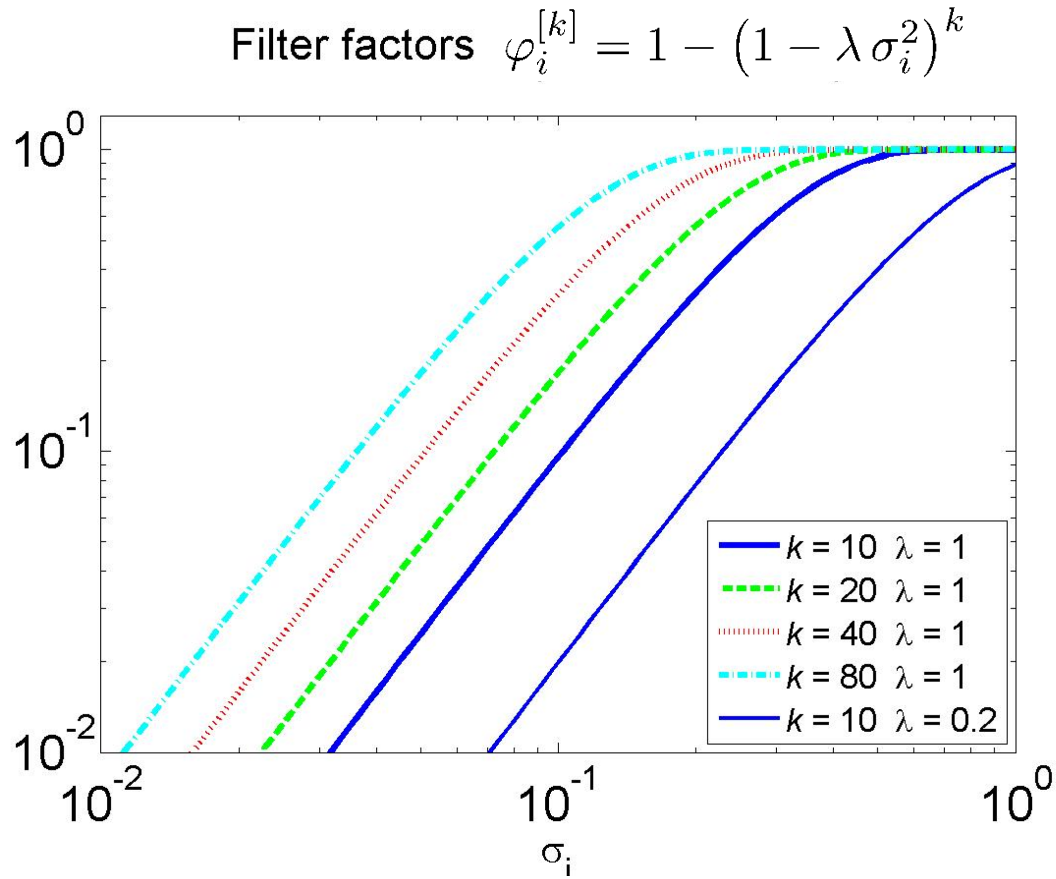
The ith component of the error, in the SVD basis, is

$$v_i^T (x^* - x^k) = (1 - \varphi_i^{[k]})\,v_i^T x^* - \varphi_i^{[k]}\,\frac{u_i^T (M^{\frac{1}{2}}\,e)}{\sigma_i} .$$

RE: regularization error

NE: noise error

The Behavior of the Filter Factors



The filter factors *dampen* the "inverted noise" $\langle u_i^T e \rangle / \sigma_i$.

$\lambda\,\sigma_i^2 \ll 1 \Rightarrow \varphi_i^{[k]} \approx k\,\lambda\,\sigma_i^2 \Rightarrow k$ and λ play the same role.

Analysis of Semi-Convergence – ART

Not much theory has been developed for the semi-convergence of ART.

A first attept:

T. Elfving, P. C. Hansen, and T. Nikazad, *Semi-convergence properties of Kaczmarzs method*, Inverse Problems, 30 (2014):

$$\|\text{noise-error}_k\|_2 \approx \frac{\sqrt{\omega}\,\delta}{\sigma_r}\sqrt{k} + \mathcal{O}(\sigma_r^2).$$

AIR Tools – A MATLAB Package of Algebraic Iterative Reconstruction Methods

- Some important algebraic iterative reconstruction methods
- presented in a common framework
- using identical functions calls,
- and with easy access to:
  - strategies for choosing the relaxation parameter,
  - strategies for stopping the iterations.

The package allows the user to easily test and compare different methods and strategies on test problems.

Also: “model implementations” for dedi-cated software (Fortran, C, Python, ...).

Contents of the Package

ART – Algebraic Reconstruction Techniques

- Kaczmarz's method + symmetric and randomized variants.
- Row-action methods that treat one row of A at a time.

SIRT – Simultaneous Iterative Reconstruction Techniques

- Landweber, Cimmino, CAV, DROP, SART.
- These methods are based on matrix multiplications.

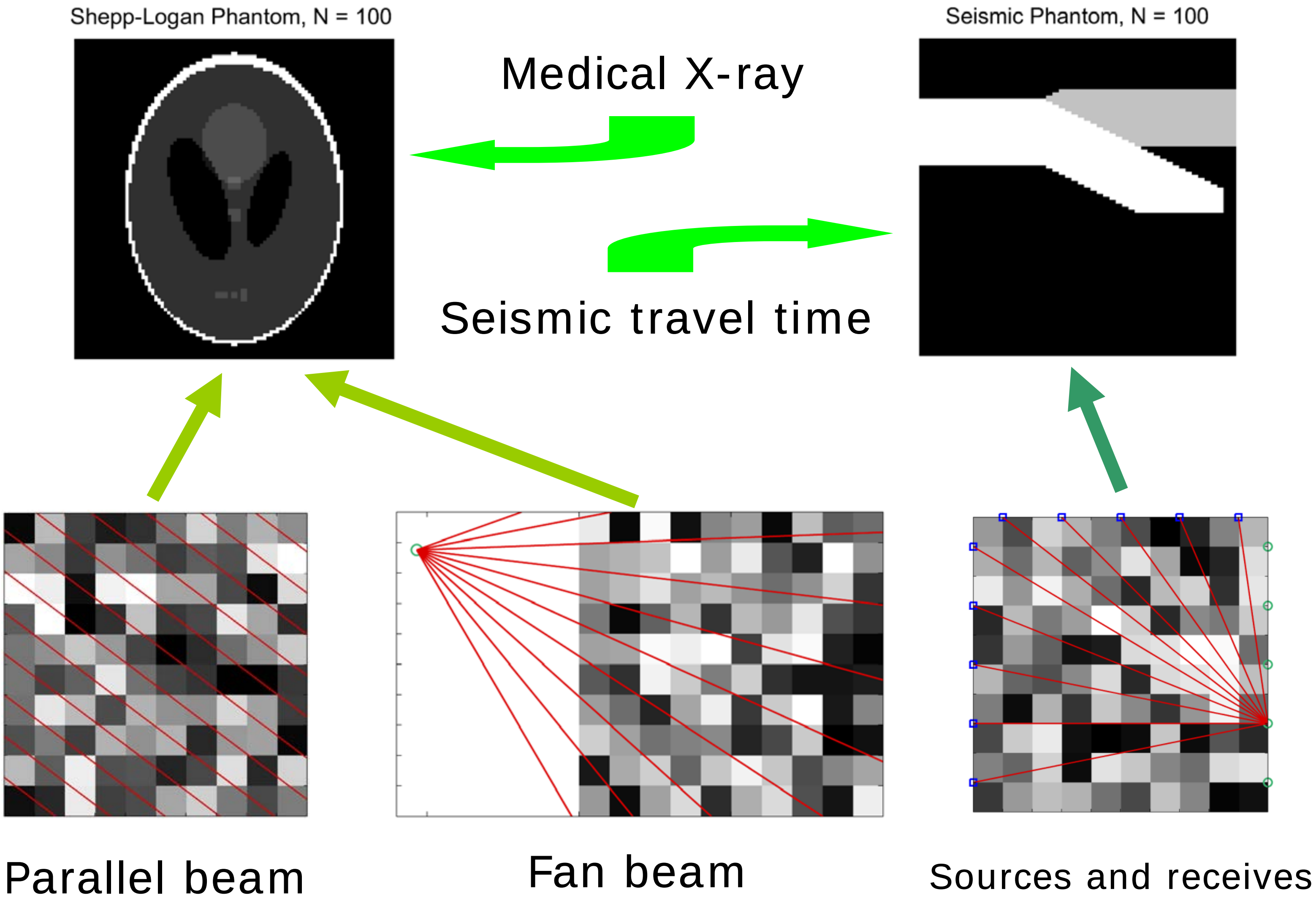
Making the methods useful

- Choice of relaxation parameter λ.
- Stopping rules for semi-convergence.
- Non-negativity constraints.

Tomography test problems

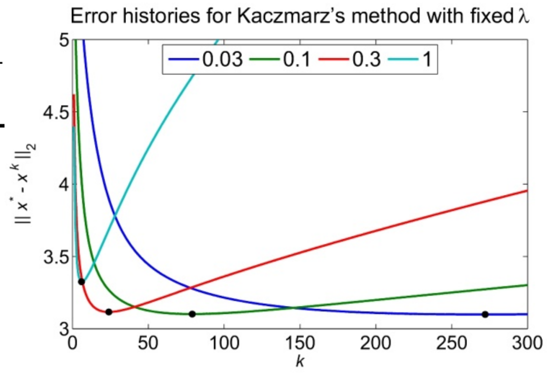
- Medical X-ray (parallel beam, fan beam), seismic travel-time, binary and smooth images (parallel beam)

Tomography Test Problems



Choosing the Relaxation Parameter

Training. Using a noisy test problem, find the *fixed* $\lambda_k = \lambda$ that gives the fastest semi-convergence. We implemented a modified golden-section search for all methods.



Line search (Dos Santos):

$$\lambda_k^{\text{line}} = \langle M r^k, r^k \rangle \,/\, \|A^T M r^k\|_2^2, \qquad r^k = b - A\,x^k.$$

Control noise propagation (Elfving, Nikazad, H):

$$\lambda_k^{\text{cnp}} = \begin{cases} \dfrac{\sqrt{2}}{\rho} & k = 0, 1 \\ \nu\,\dfrac{2}{\rho}\,\dfrac{1-\zeta_k}{(1-\zeta_k^k)^2} & k \geq 2, \end{cases}$$

ζ_k = root of a certain polynomial, ν = fudge parameter.

Stopping Rules

Let $\delta = \|e\|_2$, $\quad \tau =$ fudge parameter found by training, and $r_M^k = M^{1/2}(b - A\,x^k)$. Find the smallest k such that:

Discrepancy principle:

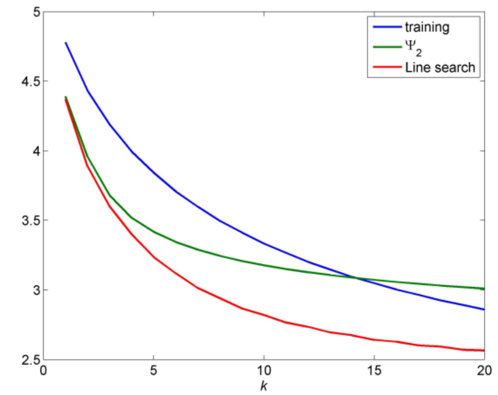
$$\begin{cases} \|r_M^k\|_2 \le \tau\,\delta\,\|M^{1/2}\|_2 & \text{SIRT methods with } T = I \\ \|r^k\|_2 \le \tau\,\delta & \text{all other methods.} \end{cases}$$

Monotone error rule (SIRT methods only):

$$\frac{\langle r_M^k, r_M^k + r_M^{k+1}\rangle}{\|r_M^k\|_2} \le \tau\,\delta\,\|M^{1/2}\|_2 .$$

NCP = normalized cumulative periodogram (Bert Rust): stop when the residual can be considered as noise.

Using AIR Tools – An Example



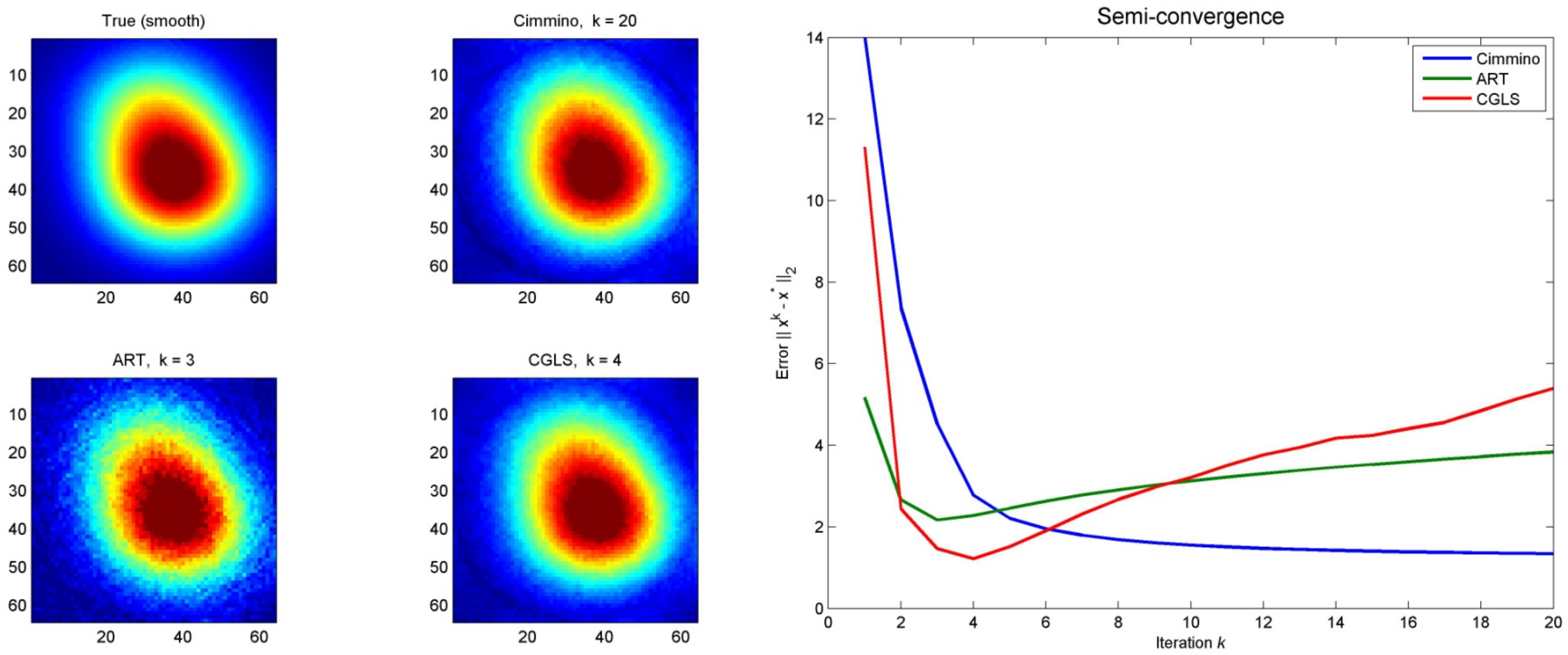

```
N = 24;          % Problem size is N-by-N.
eta = 0.05;      % Relative noise level.
kmax = 20;       % Number of of iterations.

[A,bex,xex] = fanbeamtomo(N,10:10:180,32); % Test problem
nx = norm(xex); e = randn(size(bex));       %  with noise.
e = eta*norm(bex)*e/norm(e); b = bex + e;

lambda = trainLambdaSIRT(A,b,xex,@cimmino); % Train lambda.
options.lambda = lambda;                     % Iterate with
  X1 = cimmino(A,b,1:kmax,[],options);       %  fixed lambda.
options.lambda = 'psi2';                     % Iterate with
  X2 = cimmino(A,b,1:kmax,[],options);       %  'ncp' strategy.
options.lambda = 'line';                     % Iterate with
  X3 = cimmino(A,b,1:kmax,[],options);       %  line search.
```

Using AIR Tools – Another Example

```
N = 64;                      % Problem size.
eta = 0.02;                  % Relative noise level.
k = 20;                      % Number of iterations.
[A,bex,x] = odftomo(N);      % Test problem, smooth image.

% Noisy data.
e = randn(size(bex)); e = eta*norm(bex)*e/norm(e); b = bex + e;

% ART (Kaczmarz) with non-negativity constraints.
options.nonneg = true;
Xart = kaczmarz(A,b,1:k,[],options);

% Cimmino with non-neg. constraints and Psi-2 relax. param. choice.
options.lambda = 'psi2';
Xcimmino = cimmino(A,b,1:k,[],options);

% CGLS followed by non-neg. projection.
Xcgls = cgls(A,b,1:k); Xcgls(Xcgls<0) = 0;
```

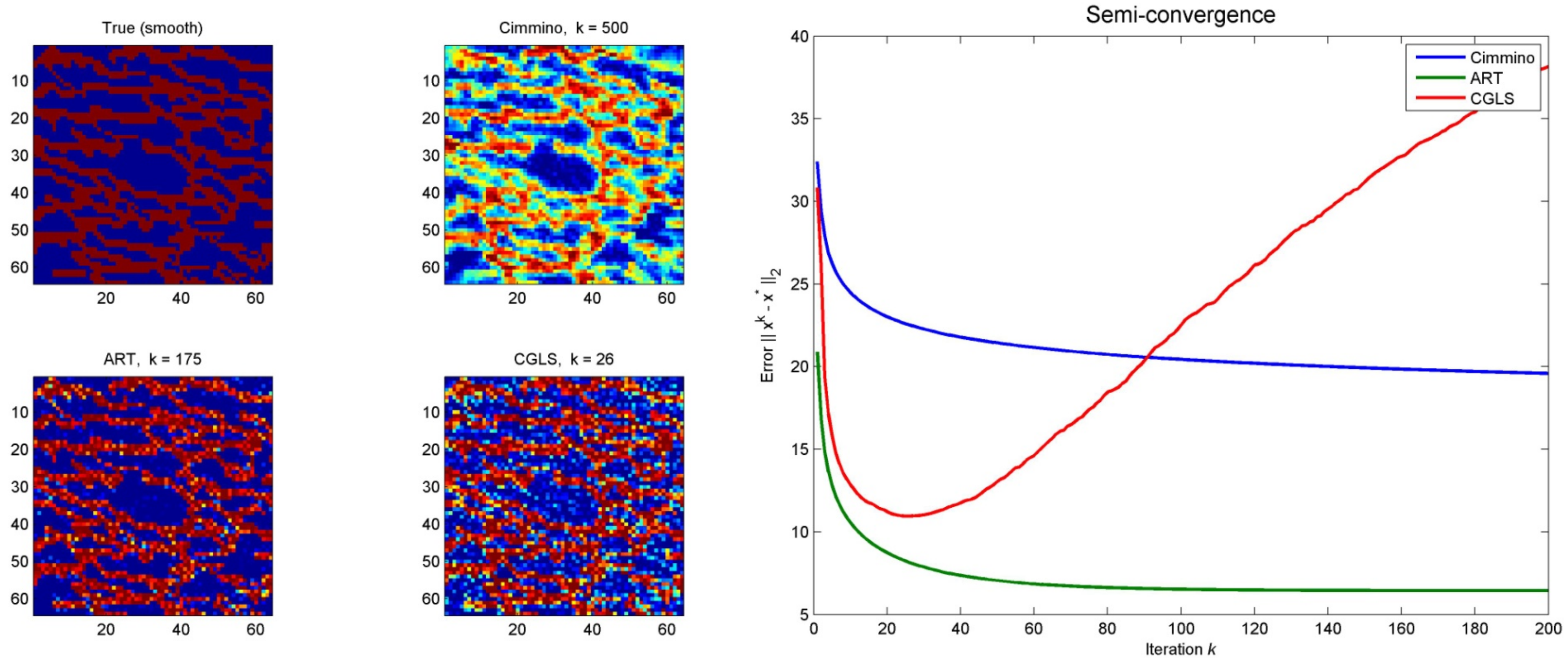
Next page

Results for Smooth Image Example



CGLS gives the best result in just k = 4 iterations.

Results for Binary Image Example

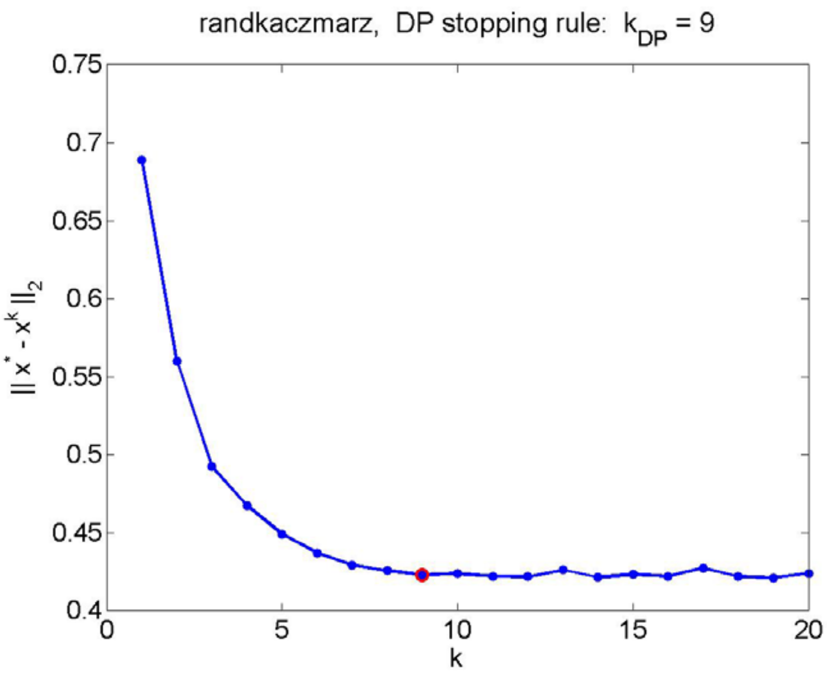


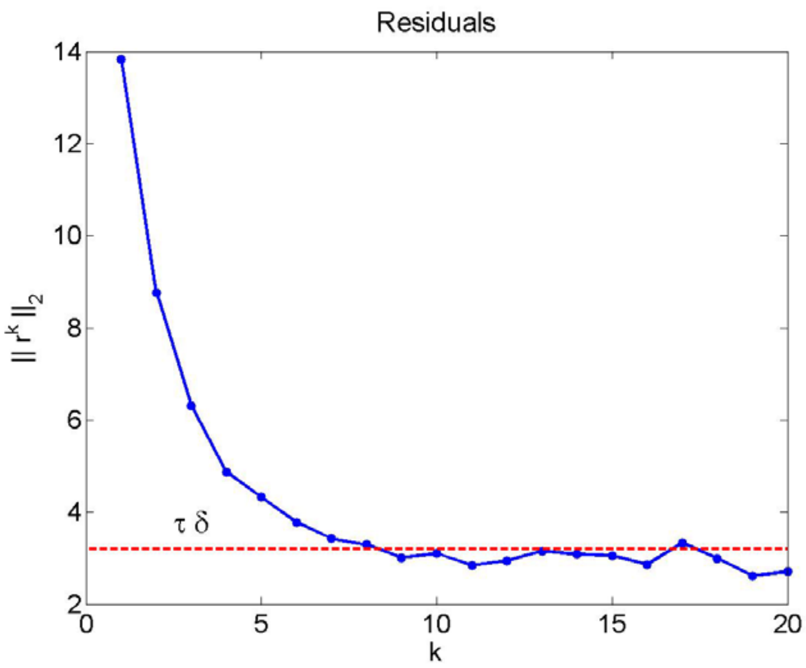
ART (Kaczmarz) is the most succesful method here.

Using AIR Tools – Yet Another Example

```
N = 24;         % Problem size is N-by-N.
eta = 0.05;     % Relative noise level.
kmax = 20;      % Number of of iterations.

[A,bex,xex] = fanbeamtomo(N,10:10:180,32); % Test problem
nx = norm(xex); e = randn(size(bex));      %  with noise.
e = eta*norm(bex)*e/norm(e); b = bex + e;

% Find tau parameter for Discrepancy Principle by training.
delta = norm(e);
options.lambda = 1.5;
tau = trainDPME(A,bex,xex,@randkaczmarz,'DP',delta,2,options);

% Use randomized Kaczmarz with DP stopping criterion.
options.stoprule.type = 'DP';
options.stoprule.taudelta = tau*delta;
[x,info] = randkaczmarz(A,b,kmax,[],options);
k = info(2);    % Number of iterations used.
```

Results for Randomized Kaczmarz Example





For some methods the residual norms do not decay monotonically. We stop when the residual norm is below $\tau\delta$ for the first time.

Conclusions

- Algebraic methods are flexible and fast.
- They allow incorporation of prior information, e.g.,
  - non-negativity or box constraints,
  - smoothness constraints,
  - piecewise smoothness = total variation (not covered).
- Several methods available for choosing the optimal fixed λ.
- The MATLAB package AIR Tools is available.