ART Exhibit

Per Christian Hansen

joint work with, among others

Tommy Elfving

Touraj Nikazad

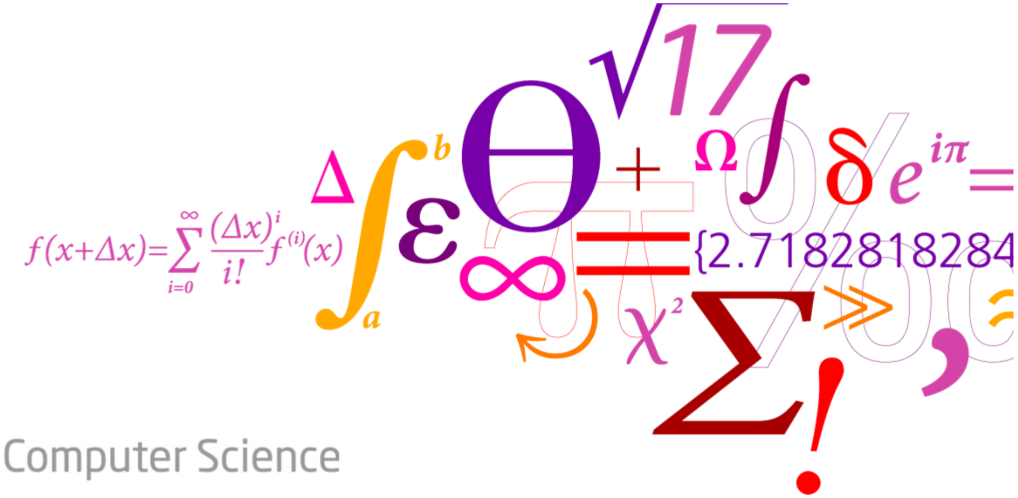
Hans Henrik B. Sørensen

HDtomo

HIGH - DEFINITION TOMOGRAPHY

DTU Compute

Department of Applied Mathematics and Computer Science

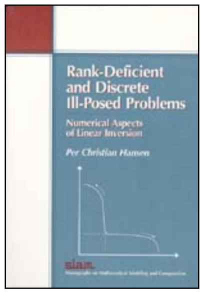


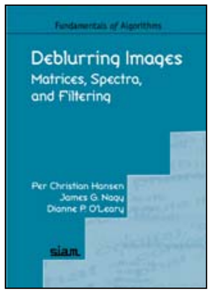
About Me …

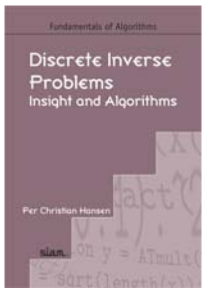
Forward problem **Inverse Problem**

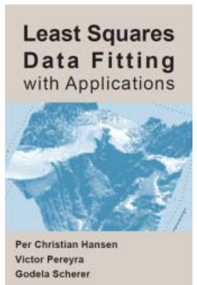
Interests: **numerical methods for inverse problems and tomography**, fast and reliable numerical regularization algorithms, matrix computations, image deblurring algorithms, signal processing, Matlab software, …

HDtomo
HIGH - DEFINITION TOMOGRAPHY









Why are We Interested in ART?

There are many ways to compute reconstructions in tomography: explicit inversion formulas, Bayesian methods, algebraic iterative methods, variational formulations, . . .

I will focus on a particular algebraic iterative method, ART:

- surprisingly simple to formulate,
- has a simple geometric interpretation,
- works well for a number of applications,
- has fast initial convergence,
- easily allows simple constraints (e.g., nonnegativity).

What is ART?

A simple iterative procedure for solving $A\,x = b$ where each iteration updates x via sweeps over the rows a_i^T of the matrix A.

Kaczmarz (1937): orthogonally project x on the hyperplane defined by a_i^T and the corresponding element b_i of the right-hand side:

$$x \leftarrow \mathcal{P}_i\,x = x + \frac{b_i - a_i^T x}{\|a_i\|_2^2}\,a_i \ , \qquad i = 1, 2, \ldots, m \ .$$

Gordon, Bender, Herman (1970): coined the term "ART" and introduced a nonnegativity projection:

$$x \leftarrow \max\left\{0\,,\, x + \frac{b_i - a_i^T x}{\|a_i\|_2^2}\,a_i\right\} \ , \qquad i = 1, 2, \ldots, m \ .$$

"ART" is now used synonymously with Kaczmarz's formulation with a relaxation parameter ω_k and a projection $\mathcal{P}_C$ on a convex set:

$$x \leftarrow \mathcal{P}_C\left(x + \omega_k\,\frac{b_i - a_i^T x}{\|a_i\|_2^2}\,a_i\right) \ , \qquad i = 1, 2, \ldots, m \ .$$

Software for ART

I am afraid that this list is far from complete.

- SNARK09: C++ package from NYU, 2D reconstructions.
- ASTRA: C++ & CUDA with Matlab wrapper, from Antwerp + CWI.
- Image reconstruction toolbox: Matlab package from Prof. Jeff Fessler, Univ. of Michigan
- AIR TOOLS: Matlab package from DTU.
- What did I miss?

Some Interesting ART Topics

ART is a rich source for research problems!

This list is quite biased towards my own work with the AIR TOOLS.

- Semi-convergence theory.
- Implementation of block ART.
- Choice of relaxation paremter.
- Stopping Rules.
- Extensions and variations of ART.

This presentation

IOP Publishing — Inverse Problems
Inverse Problems **30** (2014) 055007 (16pp) — doi:10.1088/0266-5611/30/5/055007

Semi-convergence properties of Kaczmarz's method

Tommy Elfving[1], Per Christian Hansen[2] and Touraj Nikazad[3]

[1] Department of Mathematics, Linköping University, SE-581 83 Linköping, Sweden
[2] Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs Lyngby, Denmark
[3] Department of Mathematics, Iran University of Science and Technology, Narmak, Tehran, 16846-13114, Iran

E-mail: toelf@mai.liu.se, pcha@dtu.dk and tnikazad@yahoo.com

Received 29 August 2013, revised 6 February 2014
Accepted for publication 26 February 2014
Published 1 May 2014

Abstract
Kaczmarz's method—sometimes referred to as the algebraic reconstruction technique—is an iterative method that is widely used in tomographic imaging due to its favorable semi-convergence properties. Specifically, when applied to a problem with noisy data, during the early iterations it converges very quickly toward a good approximation of the exact solution, and thus produces a regularized solution. While this property is generally accepted and utilized, there is surprisingly little theoretical justification for it. The purpose of this paper is to present insight into the semi-convergence of Kaczmarz's method as well as its projected counterpart (and their block versions). To do this we study how the data errors propagate into the iteration vectors and we derive upper bounds for this noise propagation. Our bounds are compared with numerical results obtained from tomographic imaging.

Keywords: Kaczmarz's method, ART, sequential iterative reconstruction technique, semi-convergence, non-negativity constraints, tomographic imaging

Inverse Problems

This is to certify that the article

Semi-convergence properties of Kaczmarz's method
by Tommy Elfving, Per Christian Hansen and Touraj Nikazad

has been selected by the Editors of *Inverse Problems* for inclusion in the 'Highlights of 2014' collection. Papers are chosen on the basis of referee endorsement, novelty and scientific impact.

S R Arridge
Editor-in-Chief
Inverse Problems
www.iopscience.org/ip

IOP Publishing

SIAM J. SCI. COMPUT.
Vol. 36, No. 5, pp. C524–C546

MULTICORE PERFORMANCE OF BLOCK ALGEBRAIC ITERATIVE RECONSTRUCTION METHODS*

HANS HENRIK B. SØRENSEN† AND PER CHRISTIAN HANSEN†

Abstract. Algebraic iterative methods are routinely used for solving the ill-posed sparse linear systems arising in tomographic image reconstruction. Here we consider the algebraic reconstruction technique (ART) and the simultaneous iterative reconstruction techniques (SIRT), both of which rely on semiconvergence. Block versions of these methods, based on a partitioning of the linear system, are able to combine the fast semiconvergence of ART with the better multicore properties of SIRT. These block methods separate into two classes: those that, in each iteration, access the blocks in a sequential manner, and those that compute a result for each block in parallel and then combine these results before the next iteration. The goal of this work is to demonstrate which block methods are best suited for implementation on modern multicore computers. To compare the performance of the different block methods, we use a fixed relaxation parameter in each method, namely, the one that leads to the fastest semiconvergence. Computational results show that for multicore computers, the sequential approach is preferable.

Key words. algebraic iterative reconstruction, ART, SIRT, block methods, relaxation parameter, semiconvergence, tomographic imaging

AMS subject classifications. 65F10, 65R32

DOI. 10.1137/130920642

1. Introduction. Discretizations of tomographic imaging problems often lead to large sparse systems of linear equations with noisy data:

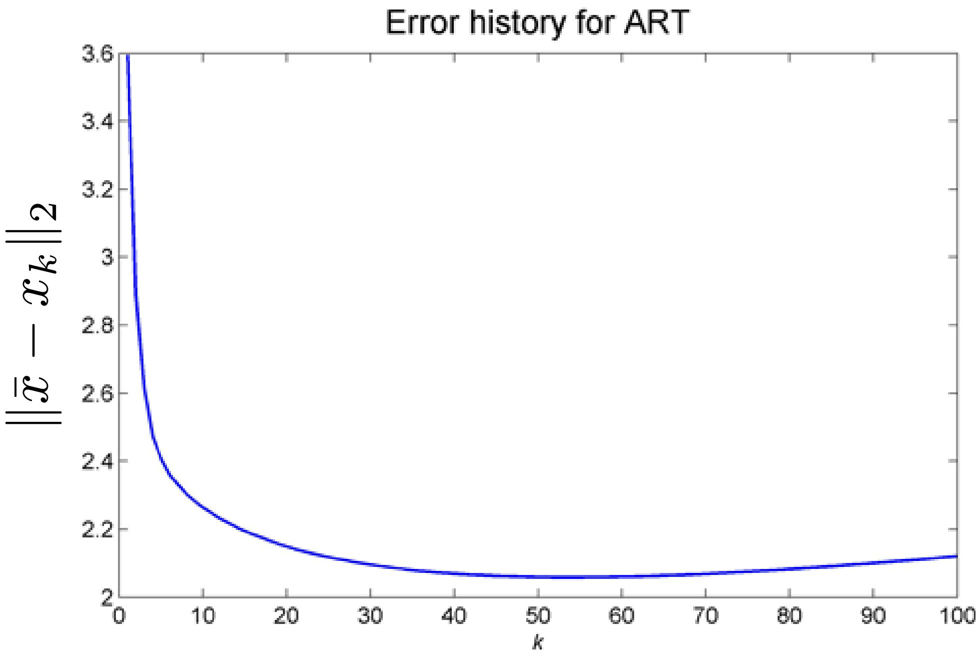
$$A\,x \simeq b, \qquad b = \bar{b} + e, \qquad A \in \mathbb{R}^{m\times n}. \tag{1.1}$$

Semi-Convergence

Notation: $b = A\,\bar{x} + e$, $\bar{x}$ = exact solution, e = noise.

Initial iterations: the error $\|\bar{x} - x_k\|_2$ decreases.

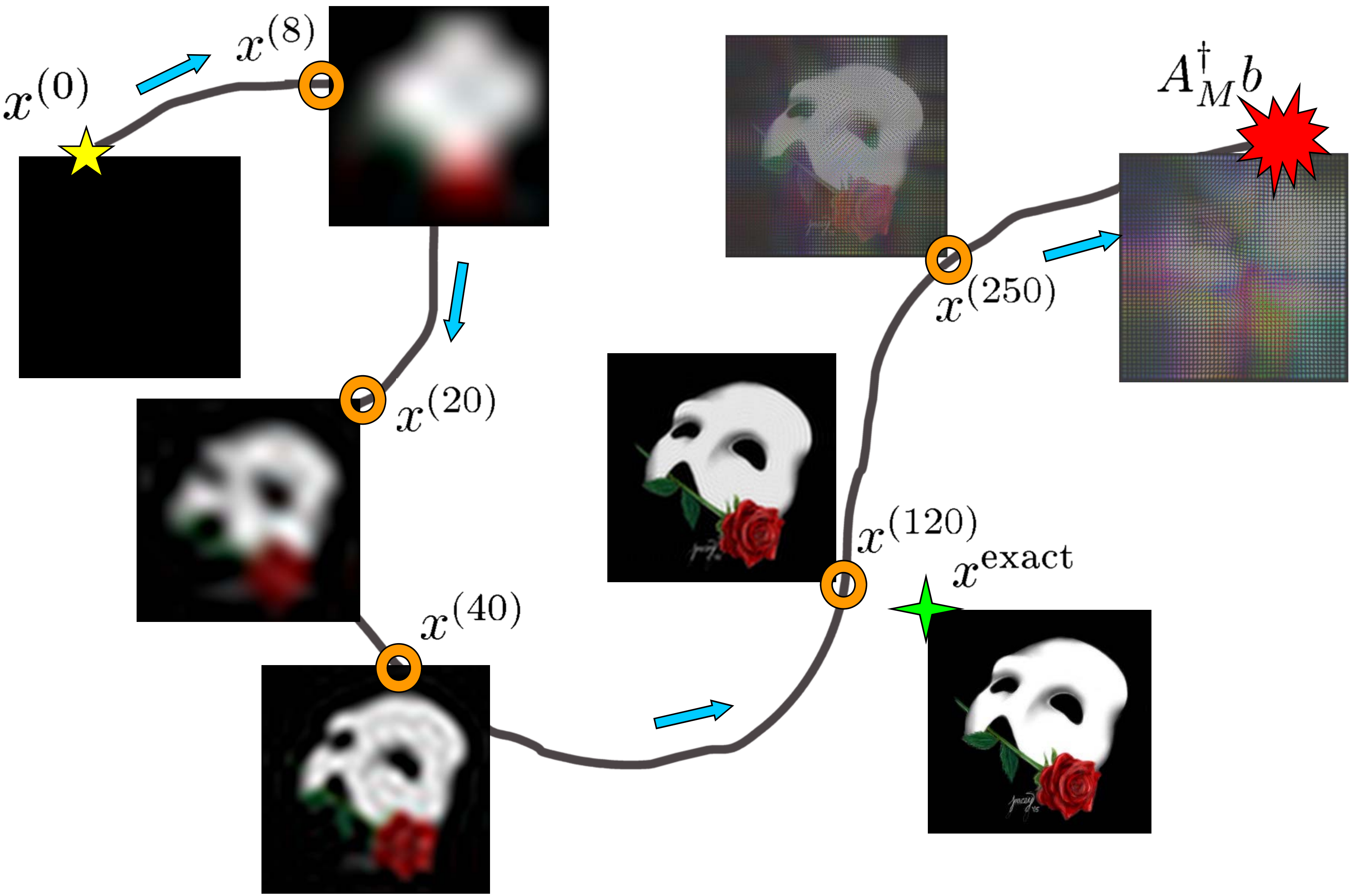
Later: the error increases as $x_k \to$ (weighted) least squares solution.



A few references:

- F. Natterer, *The Mathematics of Computerized Tomography* (1986)
- 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



Analysis of Semi-Convergence for ART

Elfving, H, Nikazad, *Semi-convergence properties of Kaczmarzs method*, Inverse Problems, 30 (2014), DOI: 10.1088/0266-5611/30/5/055007.

Let $\bar{x}$ be the solution to the noise-free problem, and let $\bar{x}^k$ denote the iterates when applying ART to $\bar{b}$. Then

$$\|x_k - \bar{x}\|_2 \leq \|x_k - \bar{x}_k\|_2 + \|\bar{x}_k - \bar{x}\|_2 \ .$$

Noise error

Iteration error

The convergence theory for ART is well established and ensures that the iteration error $\bar{x}_k - \bar{x}$ goes to zero.

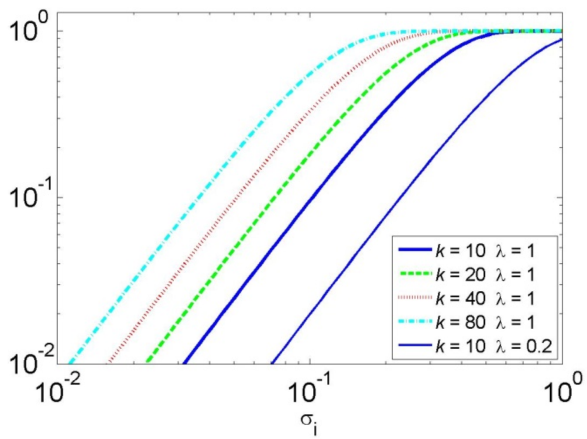
Our concern here is the noise error $e_k^{\mathrm{N}} = x_k - \bar{x}_k$. We wish to establish that it increases, and how fast.

Sidetrack: Noise Error for Landweber

Steepest descent for LSQ problem: $x_{k+1} = P_C\big(x_k + \omega A^T(b - A\,x_k)\big)$.

The unprojected case:
x_k is a filtered SVD solution:

$$
\begin{aligned}
x_k &= \textstyle\sum_{i=1}^n \varphi_i^{[k]} \frac{u_i^T b}{\sigma_i} v_i \\
\varphi_i^{[k]} &= 1 - \left(1 - \omega\,\sigma_i^2\right)^k .
\end{aligned}
$$



With projection an SVD analysis is not possible; we obtain:

$$\|x_k - \bar{x}_k\|_2 \leq \frac{\sigma_1}{\sigma_n} \frac{(1 - \omega\sigma_n^2)^k}{\sigma_n} \|b\|_2$$

and for $\omega\,\sigma_n^2 \ll 1$ we have:

$$\|x_k - \bar{x}_k\|_2 \approx \omega\,k\,\|A\|_2\,\|b\|_2.$$

Elfving, H, Nikazad, 2012

Noise Error for ART – No Projection

ART is equivalent to applying SOR to $A\,A^T y = b$, $x = A^T y$. Splitting:

$$AA^T = L + D + L^T, \qquad M = (D + \omega L)^{-1},$$

where L is strictly lower triangular and $D = \mathrm{diag}(\|a_i\|_2^2)$. Then:

$$\boxed{x_{k+1} = x_k + \omega A^T M\,(b - A\,x_k)\ .}$$

We introduce: $\quad e = b - \bar{b} = \text{noise in data}, \quad Q = I - \omega A^T M\,A.$

Then simple manipulations show that the noise error is given by

$$e_k^{\mathrm{N}} = x_k - \bar{x}_k = Q\ e_{k-1}^{\mathrm{N}} + \omega A^T M\,e = \omega \sum_{j=1}^{k-1} Q^j A^T M\,e\ .$$

After some work (see the paper) we obtain the bound

$$\|e_k^{\mathrm{N}}\|_2 \leq \omega\delta\,\frac{1-q^k}{1-q} = \omega\,k\,\|A^T M\,e\|_2 + O(\sigma_r^2).$$

Noise Error Analysis – A Tighter Bound

Further analysis (see the paper) shows that the noise error in ART is bounded above as:

$$\|e_k^{\mathrm{N}}\|_2 \le \frac{\|A^T M\, e\|_2}{\sigma_r}\, \Psi_k + \mathcal{O}(\sigma_r^2), \qquad \Psi_k = \frac{1-(1-\omega\sigma_r^2)^k}{\sigma_r}.$$

As long as $\omega\sigma_r^2 < 1$ we have

$$\Psi_k \le \sqrt{\omega}\sqrt{k}$$

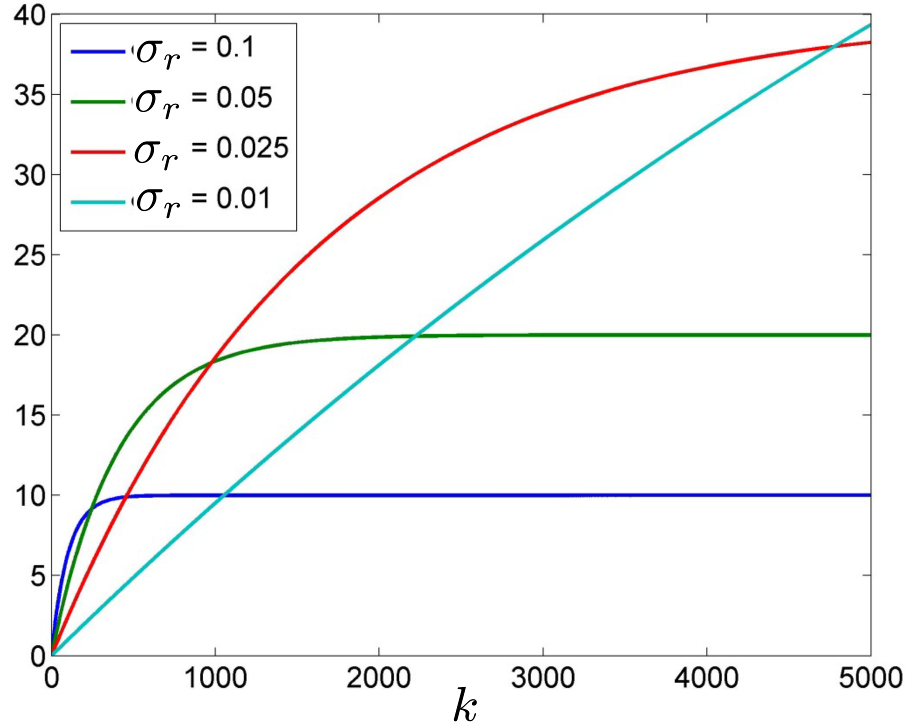
and thus

$$\boxed{\|e_k^{\mathrm{N}}\|_2 \le \frac{\sqrt{\omega}\|A^T M e\|_2}{\sigma_r}\sqrt{k} + \mathcal{O}(\sigma_r^2).}$$

This also holds for *projected* ART provided that A and P satisfy

$$y \in \mathcal{R}(A^T) \Rightarrow \mathcal{P}(y) \in \mathcal{R}(A^T).$$

Ψ_k for $\omega = 1$



Numerical Results ('paralleltomo' from AIR Tools)

The point of **semi-convergence** arises when noise error ≈ iteration error.

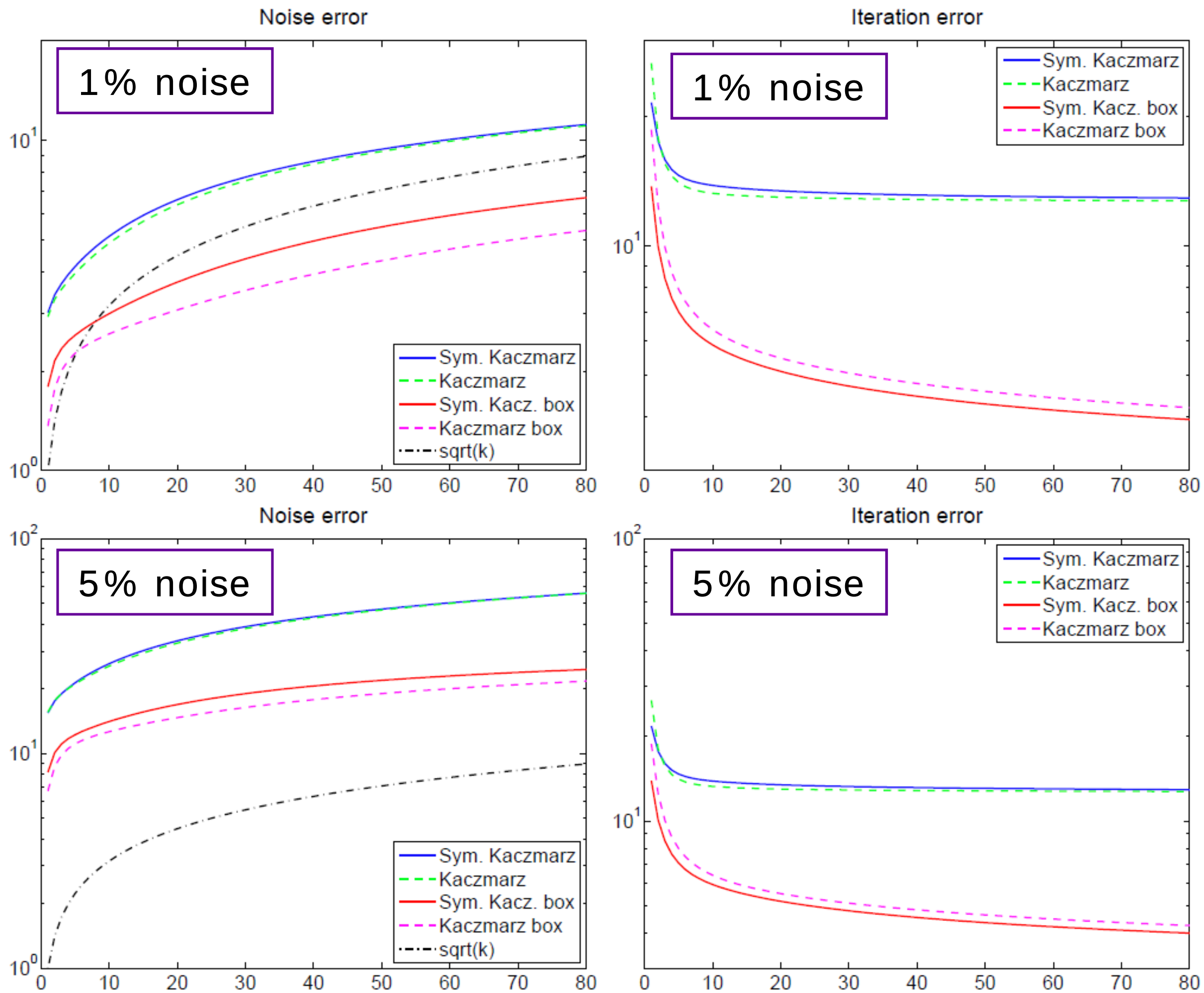
Test problem:

- 200×200 phantom,
- 60 projections at
- 3°,6°,9°,…,180°,
- m = 15,232,
- n = 40,000.

We estimate

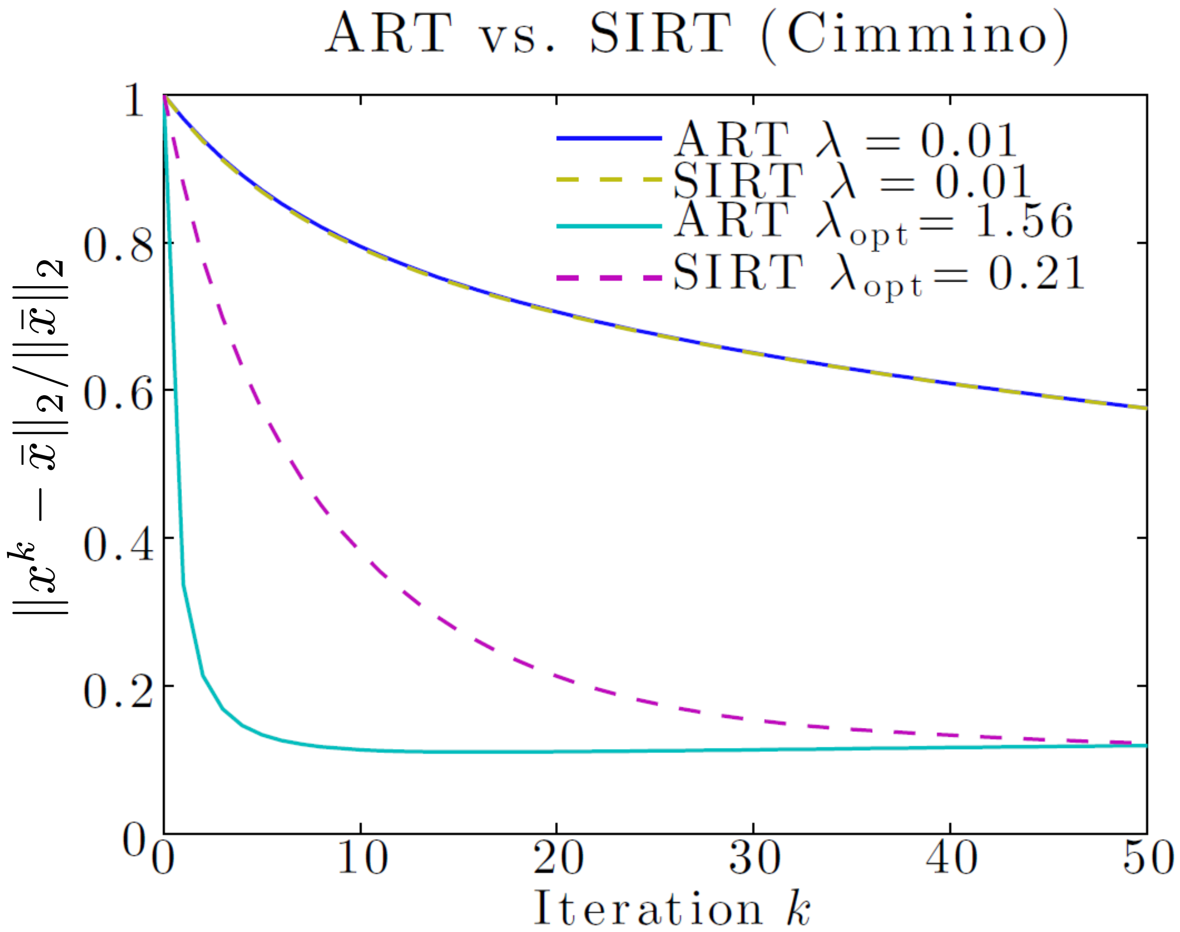
$$\frac{\sqrt{\omega}\|A^T M e\|_2}{\sigma_r} \approx 10^7.$$

Hence our bound is a wild over-estimate but it correctly *tracks* the noise error.



Implementation Issues

SIRT (Cimmino): $x \leftarrow \mathcal{P}\big(x + \omega\, A^T D^{-2} (b - A\, x)\big)$

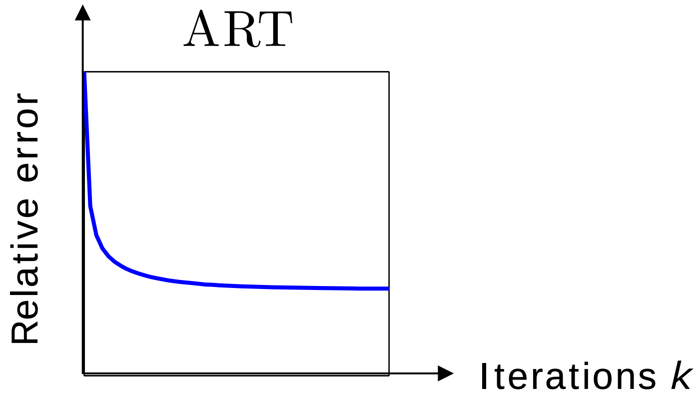


Slow convergence.

ART can converge a lot *faster* than SIRT.

Performance
1 core

$$\|x^k - \bar{x}\|_2 / \|\bar{x}\|_2$$



$m \times n$	ART t/iter
$13 \cdot 128^2 \times 64^3$	0.08 s
$13 \cdot 256^2 \times 128^3$	0.93 s
$13 \cdot 512^2 \times 256^3$	10.8 s

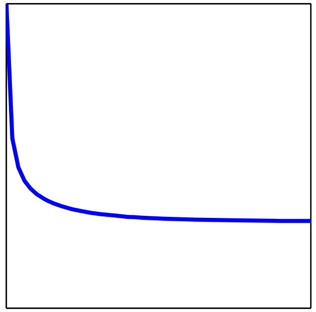
Intel Xeon E5620
2.40 GHz (1 core)

Test Problem:

- Parallel-beam tomography.
- 13 projections.
- 3D Shepp-Logan phantom, Schabel (2006).

Performance
1 core

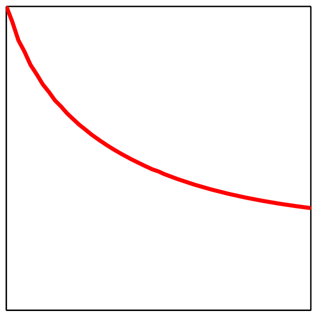
ART



$m \times n$	ART t/iter	SIRT t/iter
$13 \cdot 128^2 \times 64^3$	0.08 s	0.08 s
$13 \cdot 256^2 \times 128^3$	0.93 s	1.02 s
$13 \cdot 512^2 \times 256^3$	10.8 s	14.7 s

Intel Xeon E5620
2.40 GHz (1 core)

SIRT

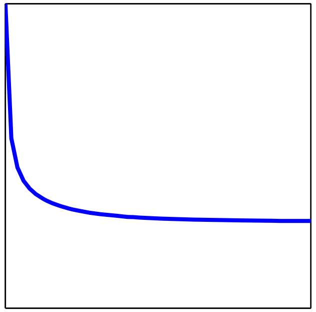


Same number of flops!

The difference is due to the cache: ART uses row a_i twice once it is loaded.

Performance
4 cores

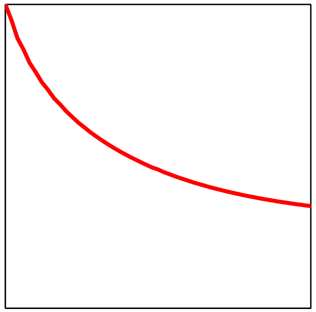
ART



$m \times n$	ART t/iter	SIRT 1 core t/iter	SIRT 4 cores t/iter
$13 \cdot 128^2 \times 64^3$	0.08 s	0.08 s	0.04 s
$13 \cdot 256^2 \times 128^3$	0.93 s	1.02 s	0.41 s
$13 \cdot 512^2 \times 256^3$	10.8 s	14.7 s	4.12 s

Intel Xeon E5620
2.40 GHz (4 cores)

SIRT



Four cores are better suited for block matrix-vector operations.

Our Dilemma

ART has *faster convergence* than SIRT – i.e., more reduction of the error per iteration.

SIRT can better take advantage of *multi-core architecture* than ART.

How to achieve the ”best of both worlds?” $\rightarrow$ Block methods!

H. H. B. Sørensen and P. C. Hansen, *Multi-core performance of block algebraic iterative reconstruction methods*, SIAM J. Sci. Comp., 36 (2014), pp. C524–C546.

Block Methods

$$A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_p \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix}, \qquad A_\ell \in \mathbb{R}^{m_\ell \times n}, \quad \ell = 1, \ldots, p,$$

In each iteration we can:

- Treat the blocks sequentially or simultaneously (i.e., in parallel).
- Treat each block by an iterative or by a direct computation.

We obtain several methods:

- Sequential processing + ART on each block $\rightarrow$ classical ART
- Sequential processing + SIRT on each block
- Sequential processing + pseudoinverse of A_ℓ
- Parallel processing + ART on each block
- Parallel processing + SIRT on each block $\rightarrow$ classical SIRT
- Parallel processing + pseudoinverse of A_ℓ

Block-Sequential Methods

Algorithm: Block-Sequential

Initialization: choose an arbitrary $x^0 \in \mathbb{R}^n$

Iteration: for $k = 0, 1, 2, \ldots$

$$x^{k,0} = x^{k-1}$$
$$x^{k,\ell} = P\big(x^{k,\ell-1} + \omega\, A_\ell^T\, M_\ell\, (b_\ell - A_\ell\, x^{k,\ell-1})\big), \quad \ell = 1, 2, \ldots, p$$
$$x^k = x^{k-1,p}$$

SART: Andersen, Kak (1984)
Block-Iteration: Censor (1988)

The convergence depends on the number of blocks p:

- If p = 1, we recover SIRT
- If p = m, we recover ART

Parallelism given by the tradeoff: m/p rows

Variant by Elfving (1980): $M_\ell = (A_\ell A_\ell^T)^\dagger \;\Rightarrow\; A_\ell^T M_\ell = A_\ell^\dagger$

Block-Parallel Methods

Algorithm: Block-Parallel

Initialization: choose an arbitrary $x^0 \in \mathbb{R}^n$

Iteration: for $k = 0, 1, 2, \ldots$

for $\ell = 1, \ldots, p$ execute in parallel

$$x^{k,\ell} = \text{ART-sweep}(\omega, A_\ell, b_\ell, x^{k-1})$$

$$x^k = {}^1\!/_p \sum_{\ell=1}^{p} x^{k,\ell}.$$

String-Averaging:
Censor, Elfving, Herman (2001)

The convergence depends on p:

- If p = 1, we recover ART
- If p = m, we recover SIRT

Parallelism is given by: p blocks

Variants:

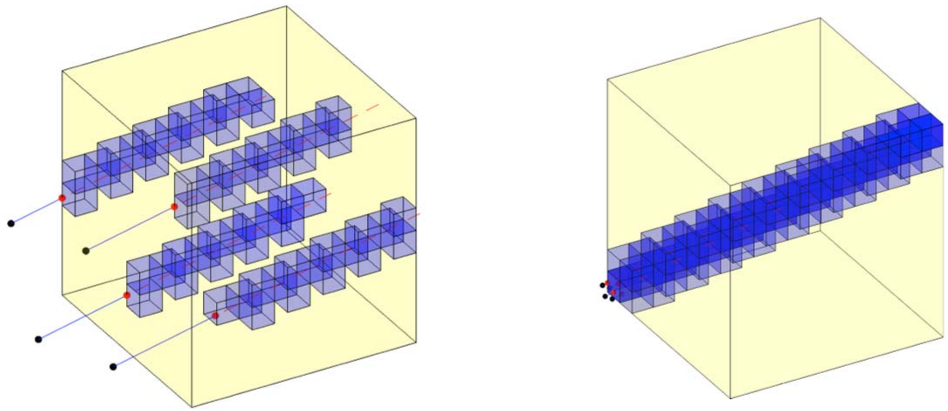
- Elfving (1980) – inner step: $x^{k,\ell} = P\big(x^{k-1,\ell} + \omega\, A_\ell^\dagger (b_\ell - A_\ell\, x^{k-1,\ell})\big)$
- CARP algorithm, Gordon & Gordon (2005):

$$x^k = \sum_{\ell=1}^{p} D_\ell\, x^{k,\ell}, \qquad D_\ell \text{ depends on sparsity structure}$$

Blocks of Structurally Orthogonal Rows

In *3D tomography*, it is easy to find sets of rows that are orthogonal due to the structure of zeros/nonzeros.

Thus, a re-ordering of the rows can produce blocks with mutually orthogonal rows (= the traces of rays are non-overlapping).

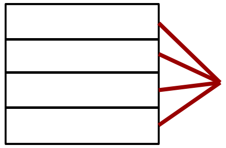


When a block has structurally orthogonal rows then ART, SIRT and "pinv" are *equivalent*. It is worthwhile to utilize this!

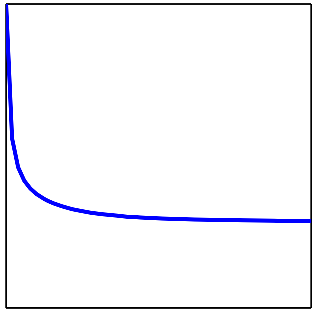
- PART algorithm, Gordon (2006)

Block Sequential

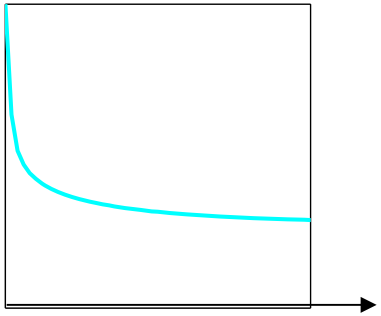
Intel Xeon
E5620
2.40 GHz
(4 cores)



ART

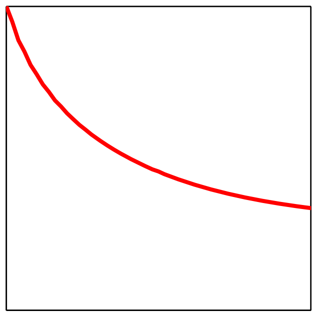


Block-Seq.



	ART	SIRT	Block-Seq.
$m \times n$	t/iter	t/iter	t/iter
$13 \cdot 128^2 \times 64^3$	0.08 s	0.04 s	0.05 s
$13 \cdot 256^2 \times 128^3$	0.93 s	0.41 s	0.48 s
$13 \cdot 512^2 \times 256^3$	10.8 s	4.12 s	4.36 s

SIRT



4 blocks

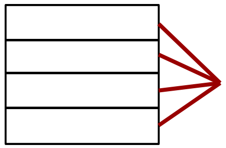
The ”building blocks” are SIRT iterations, suited for multicore.

The blocks are treated sequentailly!

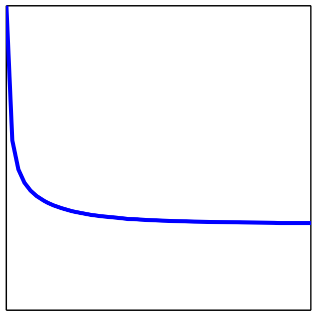
Hence the error reduction per iteration is close to that of ART.

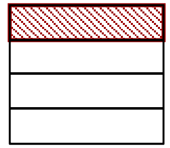
Block Parallel

Intel Xeon
E5620
2.40 GHz
(4 cores)

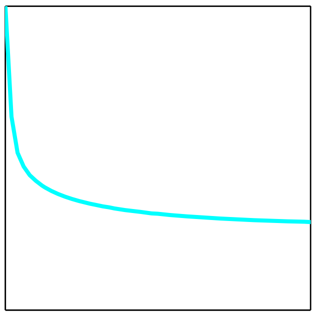


ART



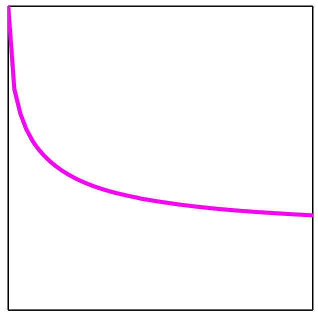


Block-Seq.

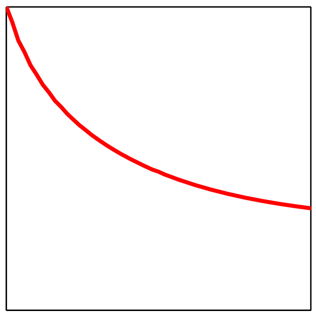


$m \times n$	ART t/iter	SIRT t/iter	Block Seq. t/iter	Block Par. t/iter
$13 \cdot 128^2 \times 64^3$	0.08 s	0.04 s	0.05 s	0.10 s
$13 \cdot 256^2 \times 128^3$	0.93 s	0.41 s	0.48 s	0.37 s
$13 \cdot 512^2 \times 256^3$	10.8 s	4.12 s	4.36 s	5.41 s

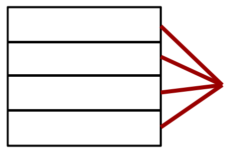
Block-Par.



SIRT

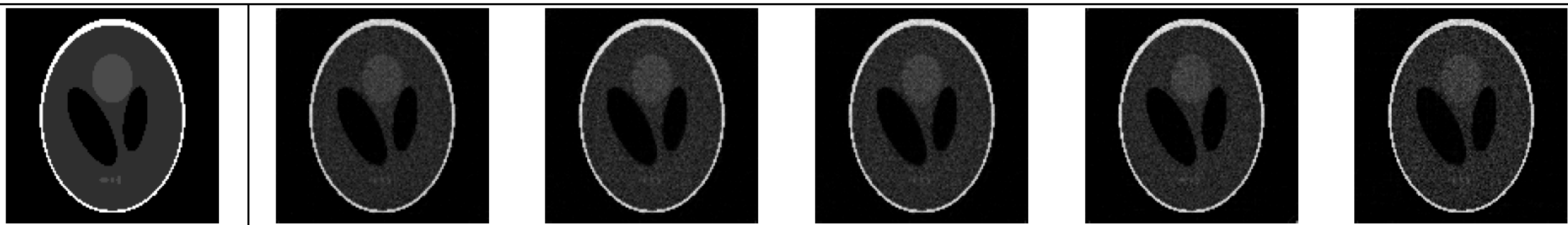


4 blocks



Multi-Core Results – 4 Cores

Multi-core: 4 cores



Method	Block-Seq	Block-Par	CARP	PART	ART
Blocks	64	4	4	460	8
Iterations	2	3	3	2	2
Time (s)	2.54	**1.89**	2.19	**1.90**	3.92

Intel Core i7-3820
3.60 GHz (4 cores)

The advantage of PART over standard ART is due to the improved use of multicore architecture.

Block-Seq: block-sequential-SIRT
Block-Par: block-parallel-ART (Censor, Elfving, Herman)
CARP: block-parallel-ART (Gordon, Gordon)
PART – utilizes struct. orthog.
ART (1 thread)

128^3 voxels
115 projections of
128×128 pixels

Multi-Core Results – 32 Cores

Multi-core: 32 cores					
					
Method	Block-Seq	Block-Par	CARP	PART	ART
Blocks	64	2	4	460	115
Iterations	2	2	3	2	2
Time (s)	4.77	5.98	7.60	**2.50**	11.29

4 socket AMD Opteron 6282 SE
2.60 GHz (32 cores)

With many cores,
PART is a clear winner.

Block-Seq: block-sequential-SIRT
Block-Par: block-parallel-ART (Censor, Elfving, Herman)
CARP: block-parallel-ART (Gordon, Gordon)
PART – utilizes struct. orthog.
ART (1 thread)

256^3 voxels
133 projections of
256×256 pixels

Conclusions

- Block algebraic iterative reconstruction techniques are able to achieve *initial convergence rate* similar to that of ART,
- and with the *smaller computing time* of SIRT, because we can utilize the multicore architecture.
- With a suitable row ordering and choice of blocks, we can produce blocks of structurally orthogonal rows.
- PART has identical convergence to ART and very good scaling properties in practice.
- Next step: target GPUs (joint work with ASTRA group).