Convergence and Non-Convergence of Algebraic Iterative Reconstruction Methods

Per Christian Hansen

DTU Compute, Technical University of Denmark

Joint work with

Yiqiu Dong & Nicolai A. B. Riis – DTU
Tommy Elfving – Linköping University
Michiel E. Hochstenbach – TU Eindhoven

Published in SISC

Vol. 40 (2018) pp. A573–A591 + Vol. 41 (2019) pp. A1822–A1839

Prologue

Q: Why must we implement backprojection $\mathcal{B}$?

A: Otherwise we cannot implement FBP – doh!

Q: Why do we implement (forward) projection $\mathcal{R}$ (the Radon transform)?

- A1: For simulation studies, to generate artificial data.
- A2: To implement algebraic iterative reconstruction methods.

By definition, $\mathcal{B} = \text{adjoint}(\mathcal{R})$.

So who in their right mind would write software where $\mathcal{B} \neq \text{adjoint}(\mathcal{R})$?

All good HPC-programmers!

Today we will study the implications of this fact.

Linear Least Squares Problems (*familiar stuff*)

We consider noisy, ill-conditioned systems of linear equations

$$A\,x \simeq b\,, \qquad A \in \mathbb{R}^{m\times n}\,, \qquad b \in \mathbb{R}^{m}\,,$$

where A is a discretization of the forward projection $\mathcal{R}$.

We focus on the **least squares problem** $\boxed{\min_x f(x)}$ with

$$\boxed{\begin{array}{c} f(x) = \frac{1}{2}\|A\,x - b\|_M^2 = \frac{1}{2}\,(A\,x-b)^T M\,(A\,x-b) \\ \nabla f(x) = A^T M\,(A\,x-b)\,, \qquad M = \text{SPD weight matrix.} \end{array}}$$

We use first-order iterative methods (Landweber/Cimmino/etc.) with steps

$$\boxed{x^{k+1} = x^k - \omega_k \nabla f(x^k) = x^k + \omega_k A^T M\,(b - A\,x^k)\,, \quad k = 1, 2, 3, \ldots}$$

Convergence (*more familiar stuff*)

The first-order method

$$x^{k+1} = x^k + \omega_k \nabla f(x^k) \ , \qquad k = 1, 2, 3, \ldots$$

converges to a local minimum if

$$\omega_k < \frac{1}{L} \ , \qquad L \equiv \sup \frac{\|\nabla f(x) - \nabla f(y)\|_2}{\|x - y\|_2} = \text{Lipschitz constant.}$$

If A has full rank then $f(x) = \frac{1}{2}\|A\,x - b\|_M^2$ is convex, $L = \|A^T M\,A\|_2$, and the method converges to the unique weighted least squares solution

$$x_{\text{LS}} = (A^T M\,A)^{-1} A^T M\,b \ .$$

The convergence rate is **linear**, i.e., $\|x^k - x_{\text{LS}}\|_2 \leq \text{const}^k \, \|x^0 - x_{\text{LS}}\|_2$

Interpretation of A and its transpose A^T

The step $x^{k+1} = x^k + \omega_k A^T M\,(b - A\,x^k)$ involves to basic operations:

Multiplication with A $\leftrightsquigarrow$ (forward) projector.
Multiplication with A^T $\leftrightsquigarrow$ backprojector.

But many software packages implement the backprojector in such a way that it is *not* the exact transposed of the projector ($\rightarrow$ appendix slide).

- Philosophy: different discretization schemes may be appropriate for projection and backprojection.
- Practicality: HPC software should make the *most efficient use* of multi-core processors, GPUs and other hardware accelerators.

Today: Study the influence of unmatched projector/backprojector pairs on the computed solutions and the convergence of the iterations.

Convergence Analysis for Unmatched Pairs

To set the stage we consider the generic **BA Iteration**

$$x^{k+1} = x^k + \omega\, B\,(b - A\,x^k)\ , \qquad \omega > 0$$

Generally not related to solving a minimization problem!

It is a *fixed-point iteration* whose convergence depends on the product BA.

- Any fixed point x^* satisfies $BA\,x^* = B\,b$ (*unmatched normal eq.*).
- If BA is invertible then $x^* = (BA)^{-1} B\,b$.

Shi, Wei, Zhang (2011); Elfving, H (2018)

The **BA Iteration** converges to a solution of $BA\,x = B\,b$ if and only if

$$0 < \omega < \frac{2\,\mathrm{Re}\,\lambda_j}{|\lambda_j|^2} \quad \text{and} \quad \mathrm{Re}\,\lambda_j > 0, \qquad \{\lambda_j\} = \mathtt{eig}(BA)\ .$$

Dong, H, Hochstenbach, Riis (2019) – for the nerds

The following requirements for a *unique fixed point* are equivalent:

1. $BA : \mathcal{R}(B) \to \mathcal{R}(B)$ is nonsingular.
2. For every $b \in \mathbb{R}^m$, $BA\,x = B\,b$ has a unique solution $x \in \mathcal{R}(B)$.
3. $\mathcal{R}(B) \cap \mathcal{N}(BA) = \{0\}$.
4. $\mathcal{N}(BA\,B) = \mathcal{N}(B)$.
5. $\mathcal{R}(BA\,B) = \mathcal{R}(B)$.
6. $\mathrm{rank}(BA\,B) = \mathrm{rank}(B)$.
7. A is nonsingular on $\mathcal{R}(B)$ and B is nonsingular on $\mathcal{R}(AB)$.
8. $\mathcal{R}(B) \cap \mathcal{N}(A) = \{0\}$ and $\mathcal{R}(AB) \cap \mathcal{N}(B) = \{0\}$.

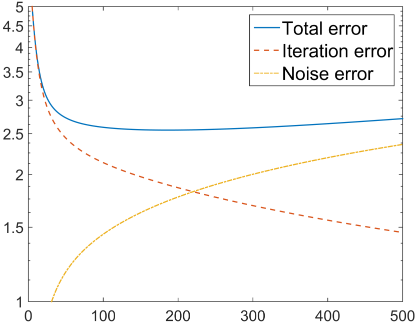
Here $\mathcal{R}(\cdot)$ = range and $\mathcal{N}(\cdot)$ = null space.

Convergence Analysis: Split the Error

Let $\bar{x}^k$ and $\bar{x}^*$ denote the iterates and the fixed point, respectively, for a noise-free right-hand side. We consider:

$$\underbrace{x^k - \bar{x}^*}_{\text{reconstruction error}} = \underbrace{x^k - \bar{x}^k}_{\text{noise error}} + \underbrace{\bar{x}^k - \bar{x}^*}_{\text{iteration error}}$$

We expect the iteration error to decrease and the noise error to increase.



Iteration Error and Noise Error

Elfving, H (2018)

The *iteration error* is given by

$$\bar{x}^k - \bar{x}^* = T^k(\bar{x}^0 - \bar{x}) \ , \quad \bar{x}^0 = \text{initial vector} \ , \quad T = I - \omega B A \ ,$$

and it follows that we have **linear** convergence:

$$\|\bar{x}^k - \bar{x}\|_2 \ \leq \ \|T^k\|_2 \, \|\bar{x}^0 - \bar{x}\|_2 \ \leq \ \|T\|_2^k \, \|\bar{x}^0 - \bar{x}\|_2 .$$

With $b = A\,\bar{x} + e$ the *noise error* satisfies

$$\boxed{\|x^k - \bar{x}^k\|_2 \ \leq \ (\omega\, c \|B\|_2)\, k\, \|e\|_2}$$

where we define the constant c by: $\sup_j \|(I - \omega B A)^j\|_2 \leq c$.

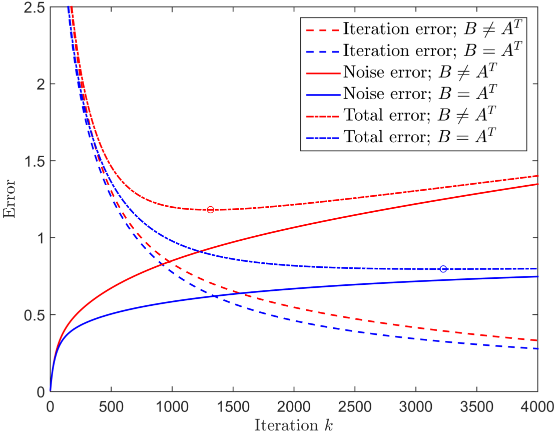
I.e., the upper bound grows linearly with the number of iterations k.

Example

Cimmino's method.

Test problem

- ▷ 64×64 phantom
- ▷ 180 projections at $1^\circ, 2^\circ, 3^\circ, \ldots, 180^\circ$
- ▷ $m = 16\,380$
- ▷ $n = 4\,096$
- ▷ $\mathrm{Re}\,\lambda_j(BA) > 0 \ \forall j$



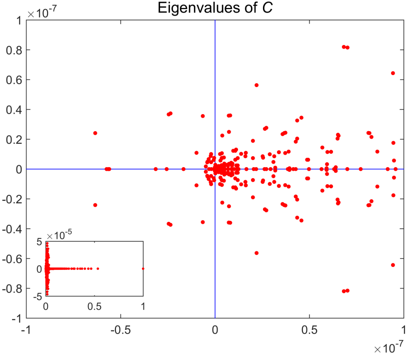
Iteration error: both versions converge to $\bar{x}$; the one with $B \neq A^T$ is slower.

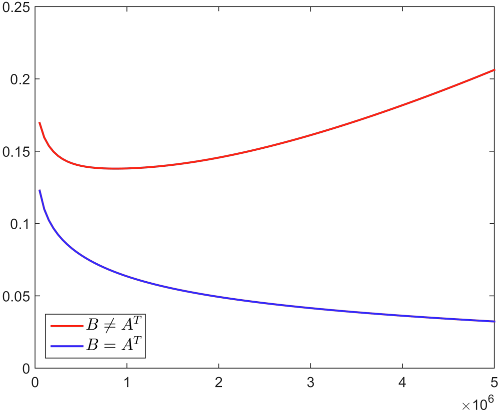
Noise error: the one for $B \neq A^T$ increases faster.

Total error: semi-convergence, the iteration with $B \neq A^T$ reaches the min. error ○ 1.181 after 1314 iterations. This error is 48% larger than the min. error ○ 0.796 for the iterations with A^T, achieved after 3225 iterations.

The Challenge: Eigenvalues with Negative Real Parts

Parallel-beam CT, unmatched pair from *ASTRA*, 64×64 Shepp-Logan phantom, 90 projection angles, 60 detector pixels, $\min \mathrm{Re}\, \lambda_j = -6.4 \cdot 10^{-8}$.

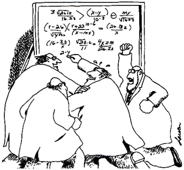




No asymptotic convergence ☹

What To Do?

1. Ask the software developers to change their implementation of $\mathcal{B}$ and/or $\mathcal{R}$? $\rightarrow$ Significant loss of comput. efficiency.



2. Use mathematics to *fix* the nonconvergence.

Take inspiration from the Tikhonov problem

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

for which a gradient step takes the form

$$\begin{aligned} x^{k+1} &= x^k - \omega\,(A^T(b - A\,x) + \alpha\,x^k) \\ &= (1 - \alpha\,\omega)\,x^k + \omega\,A^T(b - A\,x^k) \ . \end{aligned}$$

Note the factor $(1 - \alpha\,\omega)$.

The Shifted BA Iteration

Many thanks to Tommy Elfving for originally suggesting this.

We define the **shifted** version of the BA Iteration:

$$x^{k+1} = (1 - \alpha\,\omega)\,x^k + \omega\,B\,(b - A\,x^k)\ , \qquad \omega > 0$$

with just one extra factor $(1 - \alpha\,\omega)$; simple to implement.

This Shifted BA Iteration is equivalent to applying the BA Iteration with the substitutions

$$A \to \begin{bmatrix} A \\ \sqrt{\alpha}\,I \end{bmatrix}, \qquad B \to \left[\,B\,,\ \sqrt{\alpha}\,I\,\right]\ , \qquad b \to \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

Hence it is “easy” to perform the convergence analysis . . .

Convergence Results

Dong, H, Hochstenbach, Riis (2019)

Let λ_j denote those eigenvalues of BA that are different from $-\alpha$.

Then the Shifted BA Iteration converges to a fixed point if and only if α and ω satisfy

$$0 < \omega < 2\,\frac{\operatorname{Re}\lambda_j + \alpha}{|\lambda_j|^2 + \alpha\,(\alpha + 2\operatorname{Re}\lambda_j)} \qquad \text{and} \qquad \operatorname{Re}\lambda_j + \alpha > 0\ .$$

The fixed point x_α^* satisfies

$$(BA + \alpha\, I)\, x_\alpha^* = B\, b\ .$$

This result tells us how to choose the shift parameter α:

Just large enough that $\operatorname{Re}\lambda_j + \alpha > 0$ for all j.

"Perturbation" Result

How much do we perturb the *solution* when we introduce $\alpha > 0$?

Dong, H, Hochstenbach, Riis (2019)

Assume that $B\,A + \alpha\,I$ is nonsingular and the right-hand side is noise-free with $\bar{b} = A\,\bar{x}$. Then the corresponding fixed point $\bar{x}_\alpha^*$ satisfies

$$\bar{x} - \bar{x}_\alpha^* = \alpha\,(B\,A + \alpha\,I)^{-1}\bar{x}\ .$$

Notice the factor α.

With a small α – just large enough to ensure convergence – we compute a slightly perturbed solution (instead of computing nothing).

Eigenvalue Estimates (See Paper for Details)

We need to compute an estimate of the **leftmost** eigenvalue of BA, i.e., the eigenvalue with the minimal real part.

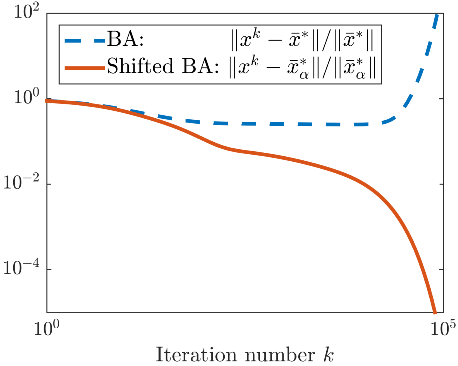
In our paper we discuss five different iterative algorithms:

- Matlab's `eigs(_,_,'smallestreal')` (calls ARPACK): baseline algorithm.
- Algorithms by Meerbergen and coauthors: robust but need too many matrix-vector multiplications.
- Krylov-Schur method by Stewart ($\sim$ implicitly restarted Arnoldi): 30% faster than Matlab's `eigs`.
- Jacobi-Davidson: slower than Krylov-Schur.
- Our own "field-of-values approximation algorithm": competitive with Krylov-Schur.

Numerical Results – Divergence and Convergence

Parallel-beam CT, 90 projections in the range 0°–180°, 80 detector pixels; 128×128 Shepp-Logan phantom; $m = 7\,200$ and $n = 16\,384$.
Both A and B are generated with the GPU-version of the ASTRA toolbox.

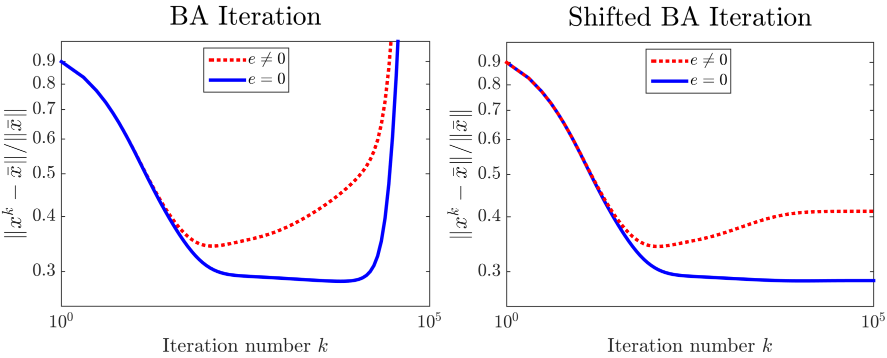
$\rho(BA) = 1.76 \cdot 10^4$
$\alpha = 1.85$



The BA Iteration diverges from $\bar{x}^* = (BA)^{-1}Bb$.

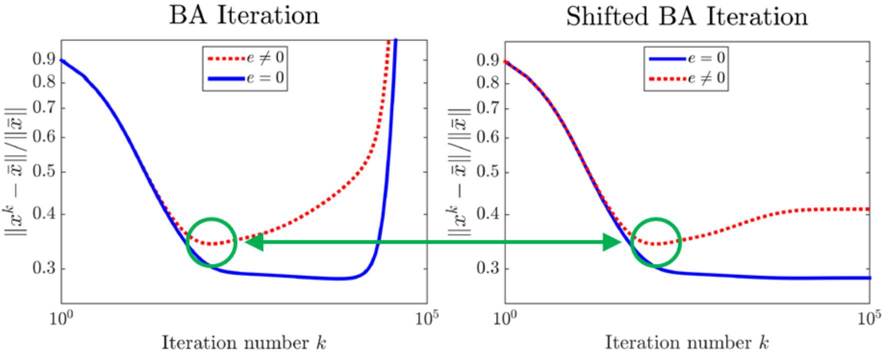
The Shifted BA Iteration converges to fixed point $\bar{x}^*_\alpha = (BA + \alpha I)^{-1}Bb$.

Numerical Results – Reconstruction Errors



- The BA Iteration diverges from the ground truth $\bar{x}$.
- The Shifted BA Iteration
  - Without noise: converges to a solution $\bar{x}_\alpha^*$ that approximates $\bar{x}$.
  - With noise: first semi-convergence, then convergence to x_α^*.

Does It Matter?



- For noisy data, the solutions at semi-convergence are almost the same.
- But is this always the case? More research is necessary.
- Also, we prefer iterative methods that converge with or without noise.

Conclusions

- We studied the influence of an unmatched pair of matrices for which backprojection $\neq$ adjoint(projection).
- Focus on SIRT method; also a concern for Karzmarz-type methods.
- Iterative methods based on unmatched pairs do not solve an optimization problem, but may still converge to a fixed point.
- The main criterion for convergence is that all eigenvalues of the iteration matrix must have positive real part.
- If violated, we introduce a small shift that ensures convergence ...
- to a fixed point that is a slightly perturbed solution ($\sim$ Tikhonov).
- The shift is computed via estimation of the leftmost eigenvalue.
- Numerical results confirm our convergence results.

Appendix: Kaczmarz and Block Sequential Iteration?

Kaczmarz (a_i^T = row of A):

$$x^{k+1} = x^k + \omega \, \frac{b_i - a_i^T x^k}{\|a_i\|_2^2} \, a_i \ .$$

Here $a_i^T x^k$ is backprojection, multiplication with a_i is forward projection.

Block Sequential Iteration (R_ℓ = block row of A):

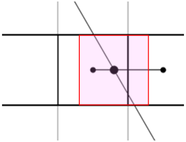
$$x^{k+1} = x^k + \omega \, R_\ell^T M_\ell \, (b_\ell - R_\ell \, x^k) \ .$$

Here we clearly see forward and back projections with blocks.

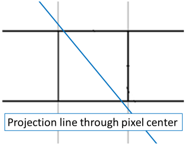
Hence our concerns from the first-order gradient methods carry over to the Karzmarz-type methods.

Appendix: ASTRA's Discretization Methods

Forward projection uses Joseph's model, also known as the interpolation model. It is identical to using the simple line model on an artificial pixel whose value is obtained by linear interpolation of two neighbouring pixels.



Backprojection projects the location of the pixel center to the detector, interpolates between the values of the two closest detector pixels, and assigns this value to the image pixel weighted by the projection line's length within the pixel. The interpolation is done on the GPU and is restricted to 256 values.



Thanks to W. J. Palenstein