Algebraic Methods for Computed Tomography

Course 31545

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark

DTU

Plan for Today

1. A bit of motivation.
2. The algebraic formulation; matrix notation and interpretation.
3. Kaczmarz's method (also known as ART) – fully sequential.
4. Cimmino's method and vatiants – fully simultaneous.
5. Least squares problems.

Points to take home today:

- Algebraic formulations provide more flexibility than formulations based on the Radon transform.
- Linear algebra provides a concise framework for formulating the associated algorithms for algebraic formulations.
- Convergence analysis of iterative algebraic methods:
  - Kaczmarz's method = ART converges for consistent problems only.
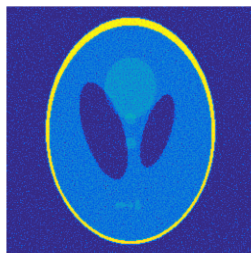 - Cimmino's method always converges.

FBP: Filtered Back Projection

- This is *the classical* method for 2D reconstructions.
- There are similar methods for 3D, such as FDK.
- Many year of use → lots of *practical experience*.
- The FBP method is *very fast* (it uses the Fast Fourier Transform)!
- The FBP method has *low memory requirements*.
- With many data, FBP gives very good results.
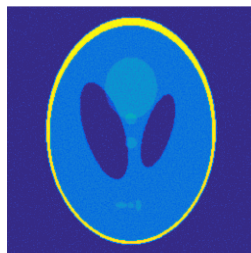- Example with 3% noise:

**Phantom**   **FBP 180 projections**   **FBP 1000 projections**

FBP Versus Algebraic Methods

- Limited data, or nonuniform distribution of projection angles or rays → *artifacts* appear in FBP reconstructions.
- Difficult to incorporate constraints (e.g., nonnegativity) in FBP.
- Algebraic methods are more flexible and adaptive.
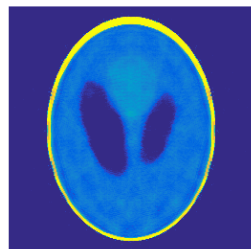- Same example with 3% noise and projection angles $15°, 30°, \ldots, 180°$:
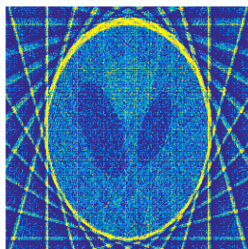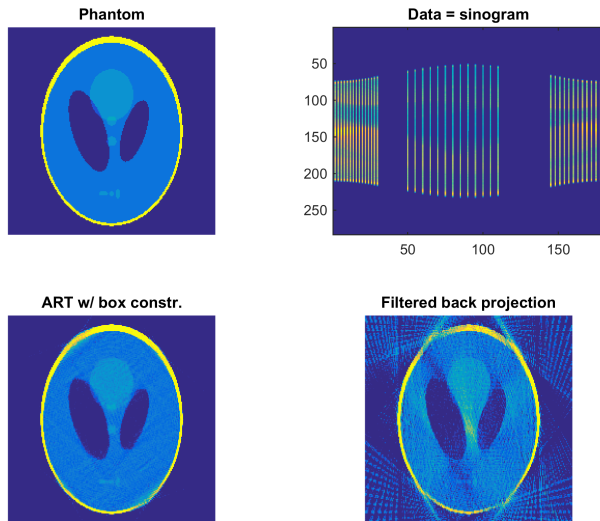
Phantom	FBP (`iradon`)	ART w/ box constraints

Algebraic Reconstruction Technique, box constraints (pixel values $\in [0,1]$).

Another Motivating Example: Missing Data

Irregularly spaced angles & "missing" angles also cause difficulties for FBP:

Phantom

Data = sinogram

ART w/ box constr.

Filtered back projection

Setting up the Algebraic Model

The damping of the ith X-ray through the object is a line integral of the attenuation coefficient $\xi(\boldsymbol{s})$ along the ray (from Lambert-Beer's law):

$$b_i = \int_{\text{ray}_i} \xi(\boldsymbol{s})\, d\ell, \qquad i = 1, 2, \ldots, m.$$

Assume that $\xi(\boldsymbol{s})$ is a constant x_j in pixel j. This leads to:

$$b_i = \sum_{j \sim \text{ray}_i} a_{ij}\, x_j, \qquad a_{ij} = \text{length of ray}_i \text{ in pixel } j,$$

where the sum is over those pixels j that are intersected by ray$_i$.

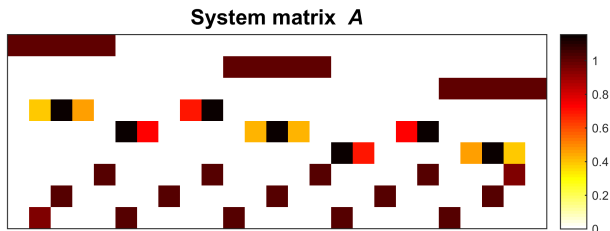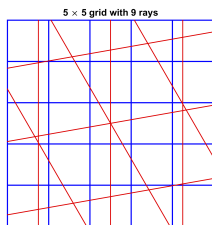If we define $a_{ij} = 0$ for those pixels *not* intersected by ray$_i$, then we have a simple sum

$$b_i = \sum_{j=1}^{n} a_{ij}\, x_j, \qquad n = \text{number of pixels.}$$

A Big and Sparse System

If we collect all m equations then we arrive at a system of linear equations

$$\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$$

with a very **sparse** *system matrix* $\boldsymbol{A}$. Example: 5 × 5 pixels and 9 rays:

5 × 5 grid with 9 rays

System matrix A

A really big advantage is that we only set up equations for the data that we actually have. In case of missing data, e.g., for certain projection angles or certain rays in a projection, we just omit those from the linear system.
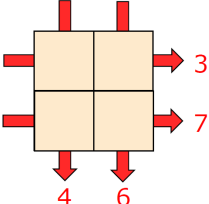
The System Matrix is Very Sparse

Another example: 256×256 pixels and 180 projections with 362 rays each. The system matrix $\boldsymbol{A}$ is $65,160 \times 65,536$ and has $\approx 4.27 \cdot 10^9$ elements. There are $15,018,524$ nonzero elements corresponding to a fill of 0.35%.
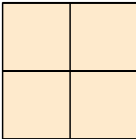

nnz(A) = 15018524,   1000×1000 zoom
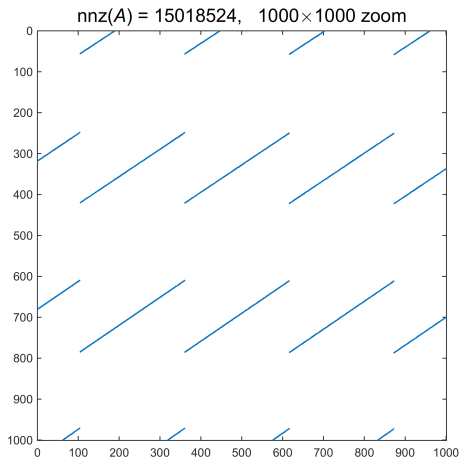
A "Sudoku" Problem

Four unknowns, four rays $\rightarrow$ system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$:

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

Unfortunately there are infinitely many solutions, with $k \in \mathbb{R}$:
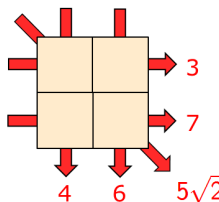
(There is an arbitrary component in the null space of the matrix $\boldsymbol{A}$.)

More Data Gives a Unique Solution

With *enough rays* the problem has a unique solution.

Here, one more ray is enough to ensure a full-rank matrix:

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \sqrt{2} & 0 & 0 & \sqrt{2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 3 \\ 7 \\ 4 \\ 6 \\ 5\sqrt{2} \end{pmatrix}$$

Algebraic Reconstruction Methods

- In principle, all we need to do in the algebraic formulation is to solve the large sparse linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$:

$$\text{Math: } \boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}, \qquad \text{MATLAB: } \mathtt{x = A\backslash b}.$$

  How hard can that be?

- Actually, this can be a formidable task if we try do use a traditional approach such as Gaussian elimination.

- Researchers in tomography have therefore focused on the use of *iterative solvers* – and they have rediscovered many methods developed by mathematicians . . .

- In tomography they are called **algebraic reconstruction methods**. They are much more flexible than FBP, but at a higher computational cost!

Some Algebraic Reconstruction Methods

Fully Sequential Methods

- Kaczmarz's method + variants.
- These are row-action methods: they update the solution using one row of $\boldsymbol{A}$ at a time.
- Fast convergence.

Fully Simultaneous Methods

- Landweber, Cimmino, CAV, DROP, SART, SIRT, ...
- These methods use all the rows of $\boldsymbol{A}$ simultaneously in one iteration (i.e., they are based on matrix multiplications).
- Slower convergence.

Block Methods

- These methods combine the best properties of the above two classes.
- They are not covered today.

Matrix Notation and Interpretation

Notation:

$$\boldsymbol{A} = \begin{pmatrix} | & | & & | \\ \boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_n \\ | & | & & | \end{pmatrix} = \begin{pmatrix} \text{---} & \boldsymbol{r}_1 & \text{---} \\ & \vdots & \\ \text{---} & \boldsymbol{r}_m & \text{---} \end{pmatrix},$$

The matrix $\boldsymbol{A}$ maps the discretized absorption coefficients (the vector $\boldsymbol{x}$) to the data in the detector pixels (the elements of the vector $\boldsymbol{b}$) via:

$$\boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} = \boldsymbol{A}\boldsymbol{x} = \underbrace{x_1\,\boldsymbol{c}_1 + x_2\,\boldsymbol{c}_2 + \cdots + x_n\,\boldsymbol{c}_n}_{\text{linear combination of columns}} = \begin{pmatrix} \boldsymbol{r}_1 \cdot \boldsymbol{x} \\ \boldsymbol{r}_2 \cdot \boldsymbol{x} \\ \vdots \\ \boldsymbol{r}_m \cdot \boldsymbol{x} \end{pmatrix}.$$

The ith row of $\boldsymbol{A}$ maps $\boldsymbol{x}$ to detector element i (through the ith ray) via the inner product:

$$b_i = \boldsymbol{r}_i \cdot \boldsymbol{x} = \sum_{j=1}^{n} a_{ij}\, x_j, \qquad i = 1, 2, \ldots, m.$$

Example of Column Interpretation

A 32×32 image has four nonzero pixels with intensities 1, 0.8, 0.6, 0.4. In the vector x these four pixels correspond to entries 468, 618, 206, 793. Hence the sinogram, represented as a vector b, takes the form

$$b = 0.6\, c_{206} + 1.0\, c_{468} + 0.8\, c_{618} + 0.4\, c_{793}.$$

Geometric Interpretation of $Ax = b$

$$
\begin{aligned}
r_1 \cdot x &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
r_2 \cdot x &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
&\qquad\qquad\qquad \vdots \\
r_m \cdot x &= a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m.
\end{aligned}
$$
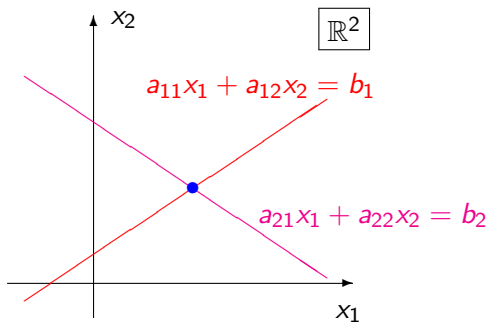
Each equation $r_i \cdot x = b_i$ defines an *affine hyperplane* in $\mathbb{R}^n$:
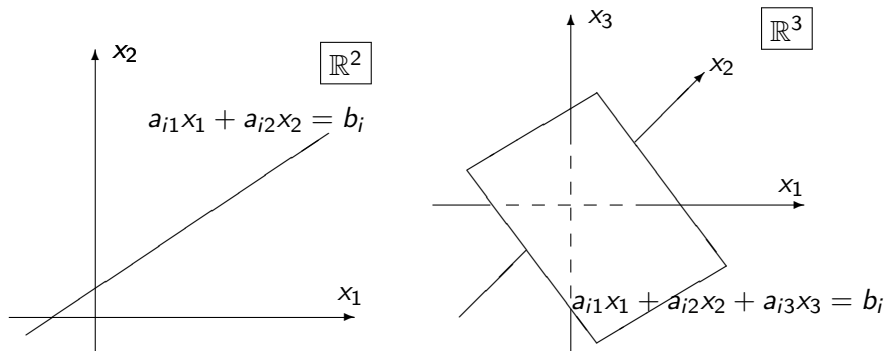
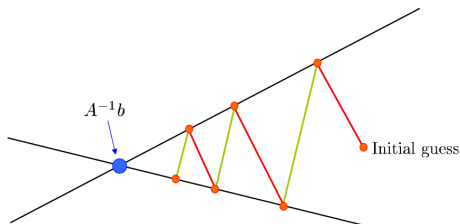# Geometric Interpretation of the Solution

Assuming that the solution to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is unique, it is the point $\boldsymbol{x} \in \mathbb{R}^m$ where all the m affine hyperplanes intersect.

Example with $m = n = 2$:

Kaczmarz's Method = Algebraic Reconstruction Technique

A simple iterative method based on the geometric interpretation.

In each iteration, and in a *cyclic fashion*, compute the new iteration vector such that precisely one of the equations is satisfied.

This is achieved by projecting the current iteration vector $\boldsymbol{x}$ on one of the hyperplanes $\boldsymbol{r}_i \cdot \boldsymbol{x} = b_i$ for $i = 1, 2, \ldots, m, 1, 2, \ldots, m, 1, 2, \ldots$

Originally proposed in 1937, and independently suggested under the name ART by Gordon, Bender & Herman in 1970 for tomographic reconstruction.

Orthogonal Projection on Affine Hyperplane
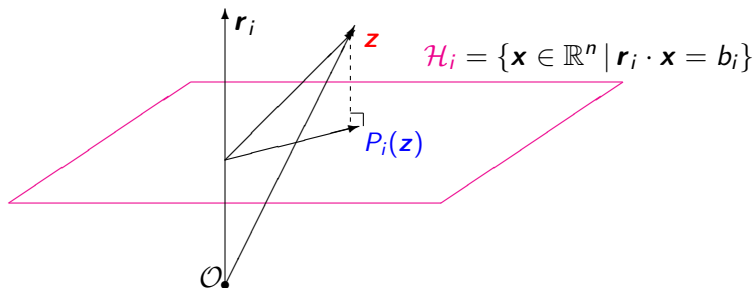
The orthogonal projection $P_i(\mathbf{z})$ of an arbitrary point $\mathbf{z}$ on the affine hyperplane $\mathcal{H}_i$ defined by $\mathbf{r}_i \cdot \mathbf{x} = b_i$ is given by:

$$P_i(\mathbf{z}) = \mathbf{z} + \frac{b_i - \mathbf{r}_i \cdot \mathbf{z}}{\|\mathbf{r}_i\|_2^2}\, \mathbf{r}_i, \qquad \|\mathbf{r}_i\|_2^2 = \mathbf{r}_i \cdot \mathbf{r}_i.$$

In words, we scale the row vector $\mathbf{r}_i$ by $(b_i - \mathbf{r}_i \cdot \mathbf{z})/\|\mathbf{r}_i\|_2^2$ and add it to $\mathbf{z}$.

Kaczmarz's Method

We thus obtain the following algebraic formulation:

Basic Kaczmarz algorithm

$\mathbf{x}^{(0)} =$ initial vector

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

$\quad i = k \pmod{m}$

$$\mathbf{x}^{(k+1)} = P_i(\mathbf{x}^{(k)}) = \mathbf{x}^{(k)} + \frac{b_i - \mathbf{r}_i \cdot \mathbf{x}^{(k)}}{\|\mathbf{r}_i\|_2^2} \, \mathbf{r}_i$$

end

Each time we have performed m iterations of this algorithm, we have performed one *sweep* over the rows of $\mathbf{A}$. Other choices of sweeps:

- **Symmetric Kaczmarz:** $i = 1, 2, \ldots, m{-}1, m, m{-}1, \ldots, 3, 2$.
- **Randomized Kaczmarz:** select row i randomly, possibly with probability proportional to the row norm $\|\mathbf{r}_i\|_2$.

Convergence Issues

The convergence of Kaczmarz's method is quite obvious from the graph on slide 17 – but can we say more?

Difficulty: the *ordering* of the rows of **A** influences the *convergence rate:*
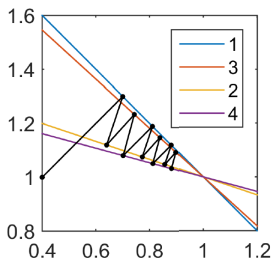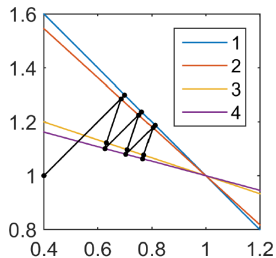
$$\begin{pmatrix} 1.0 & 1.0 \\ 1.0 & 1.1 \\ 1.0 & 3.0 \\ 1.0 & 3.7 \end{pmatrix} \boldsymbol{x} = \begin{pmatrix} 2.0 \\ 2.1 \\ 4.0 \\ 4.7 \end{pmatrix}$$

The ordering 1–3–2–4 is preferable: almost twice as fast.

Convergence of Kaczmarz's Method

One way to avoid the difficulty associated with influence of the ordering of the rows is to assume that we *select the rows randomly*.

For simplicity, assume that $\boldsymbol{A}$ is invertible and that all rows of $\boldsymbol{A}$ are scaled to unit 2-norm. Then the expected value $\mathcal{E}(\cdot)$ of the error norm satisfies:

$$\mathcal{E}\left(\|\boldsymbol{x}^{(k)} - \bar{\boldsymbol{x}}\|_2^2\right) \leq \left(1 - \frac{1}{n\,\kappa^2}\right)^k \|\boldsymbol{x}^{(0)} - \bar{\boldsymbol{x}}\|_2^2, \quad k = 1, 2, \dots ,$$

where $\bar{\boldsymbol{x}} = \boldsymbol{A}^{-1}\boldsymbol{b}$ and $\kappa = \|\boldsymbol{A}\|_2\,\|\boldsymbol{A}^{-1}\|_2$. This is **linear convergence**.

When κ is large we have

$$\left(1 - \frac{1}{n\,\kappa^2}\right)^k \approx 1 - \frac{k}{n\,\kappa^2}.$$

After $k = n$ steps, corresp. to one sweep, the reduction factor is $1 - 1/\kappa^2$. Note that there are often orderings for which the convergence is faster!

Cyclic Convergence

So far we have assumed that there is a unique solution $\bar{x} = A^{-1}b$ that satisfies $Ax = b$, i.e., all the affine hyperplanes associated with the rows of A intersect in a single point.

What happens when this is not true? $\rightarrow$ *cyclic convergence:*
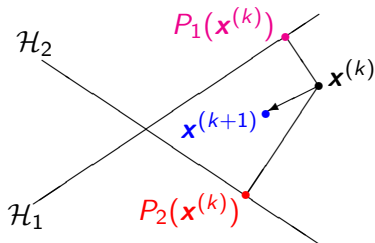

$m = 3$, $n = 2$
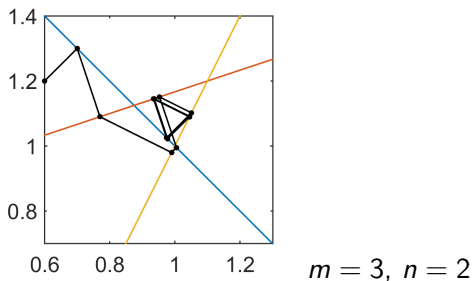
Kaczmarz's method can be brought to converge to a unique point, and we will discuss the modified algorithm later today.

From Sequential to Simultaneous Updates

Karzmarz's method accesses the rows sequentially. **Cimmino's method** accesses the rows *simultaneously* and computes the next iteration vector as the average of the all the projections of the previous iteration vector:

$$\boldsymbol{x}^{(k+1)} = \frac{1}{m} \sum_{i=1}^{m} P_i\big(\boldsymbol{x}^{(k)}\big) = \frac{1}{m} \sum_{i=1}^{m} \Big(\boldsymbol{x}^{(k)} + \frac{b_i - \boldsymbol{r}_i \cdot \boldsymbol{x}^{(k)}}{\|\boldsymbol{r}_i\|_2^2}\, \boldsymbol{r}_i\Big)$$

$$= \boldsymbol{x}^{(k)} + \frac{1}{m} \sum_{i=1}^{m} \frac{b_i - \boldsymbol{r}_i \cdot \boldsymbol{x}^{(k)}}{\|\boldsymbol{r}_i\|_2^2}\, \boldsymbol{r}_i.$$

Matrix formulation of Cimmino's Method

We can write the updating in our matrix-vector formalism as follows

$$
\begin{aligned}
\boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} + \frac{1}{m} \sum_{i=1}^{m} \frac{b_i - \boldsymbol{r}_i \cdot \boldsymbol{x}^{(k)}}{\|\boldsymbol{r}_i\|_2^2} \, \boldsymbol{r}_i \\
&= \boldsymbol{x}^{(k)} + \frac{1}{m} \begin{pmatrix} \dfrac{\boldsymbol{r}_1}{\|\boldsymbol{r}_1\|_2^2} & \cdots & \dfrac{\boldsymbol{r}_m}{\|\boldsymbol{r}_m\|_2^2} \end{pmatrix} \begin{pmatrix} b_1 - \boldsymbol{r}_1 \cdot \boldsymbol{x}^{(k)} \\ \vdots \\ b_m - \boldsymbol{r}_m \cdot \boldsymbol{x}^{(k)} \end{pmatrix} \\
&= \boldsymbol{x}^{(k)} + \frac{1}{m} \begin{pmatrix} \boldsymbol{r}_1 \\ \vdots \\ \boldsymbol{r}_m \end{pmatrix}^T \begin{pmatrix} \|\boldsymbol{r}_1\|_2^{-2} & & \\ & \ddots & \\ & & \|\boldsymbol{r}_m\|_2^{-2} \end{pmatrix} \left(\boldsymbol{b} - \begin{pmatrix} \boldsymbol{r}_1 \\ \vdots \\ \boldsymbol{r}_m \end{pmatrix} \boldsymbol{x}^{(k)} \right) \\
&= \boldsymbol{x}^{(k)} + \boldsymbol{A}^T \boldsymbol{M}^{-1} (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)}),
\end{aligned}
$$

where we introduced the diagonal matrix $\boldsymbol{M} = \operatorname{diag}\big(m\|\boldsymbol{r}_i\|_2^2\big)$.

Cimmino's Method

We thus obtain the following formulation:

> Basic Cimmino algorithm
>
> $x^{(0)} =$ initial vector
> for $k = 0, 1, 2, \ldots$
> $x^{(k+1)} = x^{(k)} + A^T M^{-1} (b - A x^{(k)})$
> end

Note that one iteration here involves all the rows of A, while one iteration in Kaczmarz's method involves a single row.

Therefore, the computational work in one Cimmino iteration is equivalent to m iterations (a sweep over all the rows) in Kaczmarz's basic algorithm.

The issue of finding a good row ordering is, of course, absent from Cimmino's method.

Convergence Study

Assume $x^{(0)} = 0$ and let I denote the $n \times n$ identity matrix; then:

$$
\begin{aligned}
x^{(k+1)} &= \sum_{j=0}^{k} (I - A^T M^{-1} A)^j A^T M^{-1} b \\
&= \left(I - (I - A^T M^{-1} A)^{k+1} \right) (A^T M^{-1} A)^{-1} A^T M^{-1} b.
\end{aligned}
$$

If A is invertible then

$$
(A^T M^{-1} A)^{-1} A^T M^{-1} b = A^{-1} M A^{-T} A^T M^{-1} b = A^{-1} b.
$$

Moreover, the largest eigenvalue of the symmetric matrix $I - A^T M^{-1} A$ is strictly smaller than one, and therefore

$$
\left(I - (I - A^T M^{-1} A)^{k+1} \right) \rightarrow I \qquad \text{for} \qquad k \rightarrow \infty.
$$

Hence the iterates $x^{(k)}$ converge to the solution $\bar{x} = A^{-1} b$.

Convergence of Cimmino's Method

To simplify the result, assume that $\boldsymbol{A}$ is invertible and that the rows of $\boldsymbol{A}$ are scaled such that $\|\boldsymbol{A}\|_2^2 = m$. Then

$$\|\boldsymbol{x}^{(k)} - \bar{\boldsymbol{x}}\|_2^2 \leq \left(1 - \frac{2}{1 + \kappa^2}\right)^k \|\boldsymbol{x}^{(0)} - \bar{\boldsymbol{x}}\|_2^2$$

where $\bar{\boldsymbol{x}} = \boldsymbol{A}^{-1}\boldsymbol{b}$, $\kappa = \|\boldsymbol{A}\|_2 \|\boldsymbol{A}^{-1}\|_2$, and we have **linear convergence**.

When $\kappa \gg 1$ then we have the approximate upper bound

$$\|\boldsymbol{x}^{(k)} - \bar{\boldsymbol{x}}\|_2^2 \lesssim (1 - 2/\kappa^2)^k \|\boldsymbol{x}^{(0)} - \bar{\boldsymbol{x}}\|_2^2,$$

showing that in each iteration the error is reduced by a factor $1 - 2/\kappa^2$.

This is almost the same factor as in one sweep through the rows of $\boldsymbol{A}$ in Kaczmarz's method.

Consistent and Inconsistent Systems

A system is *consistent* if there exists at least one x such that $Ax = b$, i.e., such that b is a linear combination of the columns c_i of A.

This is equivalent to the requirement $b \in \mathcal{R}(A)$.

Otherwise the system is *inconsistent*, $b \notin \mathcal{R}(A)$, as shown below.
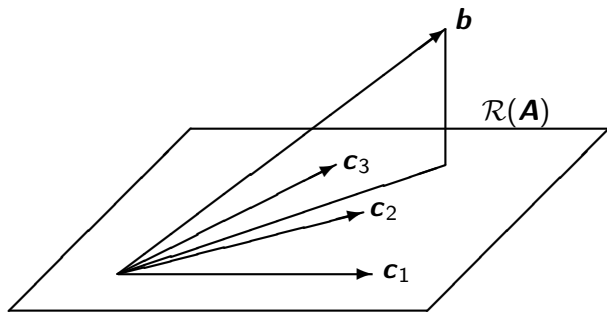
This is actually the normal situation in problems with measurement noise.

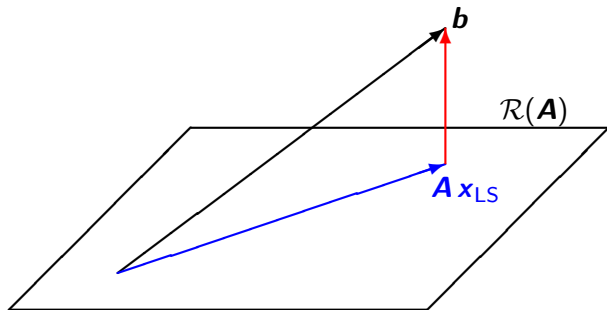# The Least Squares Solution

We must define a unique solution for inconsistent systems!

Assume that $\boldsymbol{b} = \boldsymbol{A}\bar{\boldsymbol{x}} + \boldsymbol{e}$ and $\boldsymbol{e}$ is zero-mean Gaussian noise. The best linear unbiased estimate of $\bar{\boldsymbol{x}}$ is the solution to the **least squares problem**:

$$\boldsymbol{x}_{\mathsf{LS}} = \arg\min_{\boldsymbol{x}} 1/2 \, \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2^2,$$

and $\boldsymbol{x}_{\mathsf{LS}}$ is unique when $r = n$. Geometrically, this corresponds to finding $\boldsymbol{x}_{\mathsf{LS}}$ such that $\boldsymbol{A}\boldsymbol{x}_{\mathsf{LS}}$ is orthogonal to the residual vector $\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{\mathsf{LS}}$.

Computing the Least Squares Solution

The requirement that $\boldsymbol{A}\,\boldsymbol{x}_{\text{LS}} \perp (\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}_{\text{LS}})$ leads to:

$$\left(\boldsymbol{A}\,\boldsymbol{x}_{\text{LS}}\right)^T(\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}_{\text{LS}}) = 0 \quad \Leftrightarrow \quad \boldsymbol{x}_{\text{LS}}^T(\boldsymbol{A}^T\boldsymbol{b} - \boldsymbol{A}^T\boldsymbol{A}\,\boldsymbol{x}_{\text{LS}}) = 0$$

which means that $\boldsymbol{x}_{\text{LS}}$ is the solution to the *normal equations*:

$$\boldsymbol{A}^T\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b} \quad \Rightarrow \quad \boldsymbol{x}_{\text{LS}} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}.$$

$\boldsymbol{x}_{\text{LS}}$ exists and is unique when $\boldsymbol{A}^T\boldsymbol{A}$ is invertible, which is the case when $r = n$ (i.e., the system is over-determined and $\boldsymbol{A}$ has full rank).

Bonus info: the matrix $\boldsymbol{A}^\dagger = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$ is called the *pseudoinverse* (or Moore-Penrose inverse) of $\boldsymbol{A}$.

Weighted Least Squares Solutions and Cimmino

Recall our definition of the diagonal matrix $\boldsymbol{M} = \mathrm{diag}\big(m\|\boldsymbol{r}_i\|_2^2\big)$.

We also define the *weighted least squares problem*

$$\min_{\boldsymbol{x}} \, 1/2 \, \|\boldsymbol{M}^{-1/2}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\|_2^2 \qquad \Leftrightarrow \qquad (\boldsymbol{A}^T\boldsymbol{M}^{-1}\boldsymbol{A})\,\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{M}^{-1}\boldsymbol{b}$$

and the corresponding solution $\boldsymbol{x}_{\mathrm{LS},\boldsymbol{M}} = (\boldsymbol{A}^T\boldsymbol{M}^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{M}^{-1}\boldsymbol{b}$.

Similarly we define the minimum-norm weighted least squares solution

$$\boldsymbol{x}_{\mathrm{LS},\boldsymbol{M}}^0 = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_2 \quad \text{subject to} \quad \boldsymbol{A}^T\boldsymbol{M}^{-1}\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{M}^{-1}\boldsymbol{b}.$$

<u>Cimmino's method:</u>
- $r = n = m$: convergence to $\boldsymbol{A}^{-1}\boldsymbol{b}$.
- $r = n < m$ and $\boldsymbol{b} \in \mathcal{R}(\boldsymbol{A})$: convergence to $\boldsymbol{x}_{\mathrm{LS}}$.
- $r = n < m$ and $\boldsymbol{b} \notin \mathcal{R}(\boldsymbol{A})$: convergence to $\boldsymbol{x}_{\mathrm{LS},\boldsymbol{M}}$.

Incorporating Simple Constraints

We can include constraints on the elements of the reconstructed image.

Assume that we can write the constraint as $\boldsymbol{x} \in \mathcal{C}$, where $\mathcal{C}$ is a convex set; this includes two very common special cases:
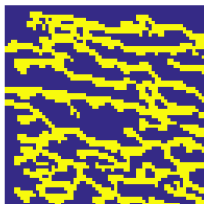
Non-negativity constraints. The set $\mathcal{C} = \mathbb{R}_+^n$ corresponds to
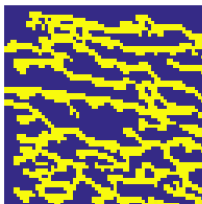
$$x_i \geq 0, \qquad i = 1, 2, \ldots, n.$$

Box constraints. The set $\mathcal{C} = [0, 1]^n$ (n-dimensional box) corresponds to
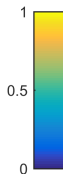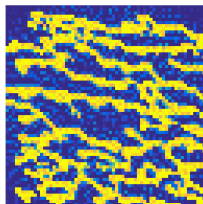
$$0 \leq x_i \leq 1, \qquad i = 1, 2, \ldots, n.$$
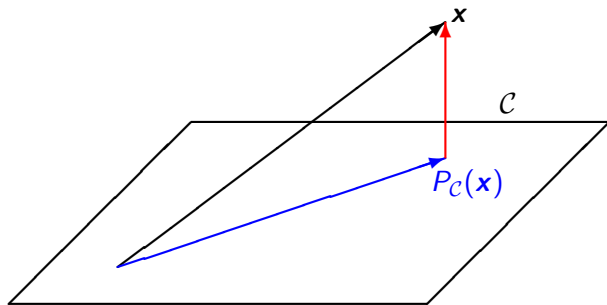

Ground truth   Box constraints   No constraints

Orthogonal Projections

Given a set $\mathcal{C}$, the orthogonal projection $P_{\mathcal{C}}(\boldsymbol{x})$ of an arbitrary vector $\boldsymbol{x} \in \mathbb{R}^n$ on $\mathcal{C}$ is the unique vector that satisfies: $P_{\mathcal{C}}(\boldsymbol{x}) \perp (\boldsymbol{x} - P_{\mathcal{C}}(\boldsymbol{x}))$.

If $\mathcal{C} = \mathbb{R}_{+}^n$ (non-negativity constraints) then, in MATLAB, we compute the corresponding projection of x as `max(x,0)`.

The Projected Algorithms

Both algorithms below solve $\min_{\mathbf{x} \in \mathcal{C}} \|\mathbf{M}^{-1/2}(\mathbf{b} - \mathbf{A}\mathbf{x})\|_2$.

<u>Projected gradient algorithm</u> $(\lambda_k < 2/\|\mathbf{A}^T \mathbf{M} \mathbf{A}\|_2)$

> $x^{(0)} = $ initial vector
> for $k = 0, 1, 2, \ldots$
> $\quad \mathbf{x}^{(k+1)} = P_{\mathcal{C}}\left(\mathbf{x}^{(k)} + \lambda_k \mathbf{A}^T \mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)})\right)$
> end

<u>Projected incremental gradient (Kaczmarz) algorithm</u> $(\lambda_k < 2)$

> $x^{(0)} = $ initial vector
> for $k = 0, 1, 2, \ldots$
> $\quad i = k \pmod{m}$
> $\quad \mathbf{x}^{(k+1)} = P_{\mathcal{C}}\left(\mathbf{x}^{(k)} + \lambda_k \frac{b_i - \mathbf{r}_i \cdot \mathbf{x}}{\|\mathbf{r}_i\|_2^2} \mathbf{r}_i\right)$
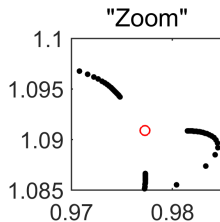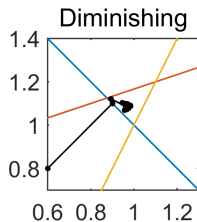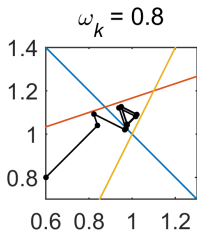> end

Iteration-Dependent Relaxation Parameter λ_k

The basic Kaczmarz algorithm gives a cyclic and non-convergent behavior.
Consider the example from slide 22 with:

$\lambda_k = 0.8$ (independent of k)   and $\lambda_k = 1/\sqrt{k}, \quad k = 0, 1, 2, \ldots$

The rightmost plot is a "zoom" of the middle plot.

- With a fixed $\lambda_k < 1$ we still have a cyclic non-convergent behavior.
- With the *diminishing relaxation parameter* $\lambda_k = 1/\sqrt{k} \to 0$ as $k \to \infty$ the iterates converge to the weighted least squares solution $x_{\mathrm{LS},M}$.

A Few References

- T. Elfving, T. Nikazad, and C. Popa, *A class of iterative methods: semi-convergence, stopping ryles, inconsistency, and constraining;* in Y. Censor, Ming Jiang, and Ge Wang (Eds.), "Biomedical Mathematics: Promising Directions in Imaging, Therapy Planning, and Inverse Problems," Medical Physics Publishing, Madison, Wisconsin, 2010.

- P. C. Hansen and J. S. Jørgensen, *AIR Tools II: algebraic iterative reconstruction methods, improved implementation*, Numer. Algo., 79 (2018), pp. 107–137.

- G. T. Herman, *Fundamentals of Computerized Tomography – Image Reconstruction from Projections*, Springer, New York, 2009.

- M. Jiang and G. Wang, *Convergence studies on iterative algorithms for image reconstruction*, IEEE Trans. Medical Imaging, 22 (2003), pp. 569–579.

- A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, IEEE Press, 1988.