Algebraic Iterative Methods with Noisy Data
Semi-Convergence and Stopping Rules

Per Christian Hansen

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

Plan for Today

1. Enter the noise $\rightarrow$ semi-convergence.
2. SVD analysis $\rightarrow$ iteration error and noise error.
3. Analysis of Landweber and Kaczmarz with projections.
4. The need for stopping rules.
5. Fit to noise level; min. of prediction error; extract all information.
6. Estimation of trace term and noise level.

Points to take home today:

- For noisy data we rely on semi-convergence of the iterative methods.
- We have a good theoretical understanding of this phenomenon.
- We need to terminate iterations at smallest reconstruction error.
- Several stopping rules are available.
- We can estimate the crucial parameters in these rules.

Real Problems Have Noisy Data

So far we have discussed how to solve $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$ by iterative methods. But when noise is present in the data, we don't quite want to do that!

We assume that the data, in the form of the right-hand side $\boldsymbol{b}$, is a sum of "clean" noise-free data $\boldsymbol{A}\,\bar{\boldsymbol{x}}$ from the ground-truth image plus a noise component $\boldsymbol{e}$:

$$\boldsymbol{b} = \boldsymbol{A}\,\bar{\boldsymbol{x}} + \boldsymbol{e}, \qquad \bar{\boldsymbol{x}} = \text{ground truth}, \qquad \boldsymbol{e} = \text{noise}.$$

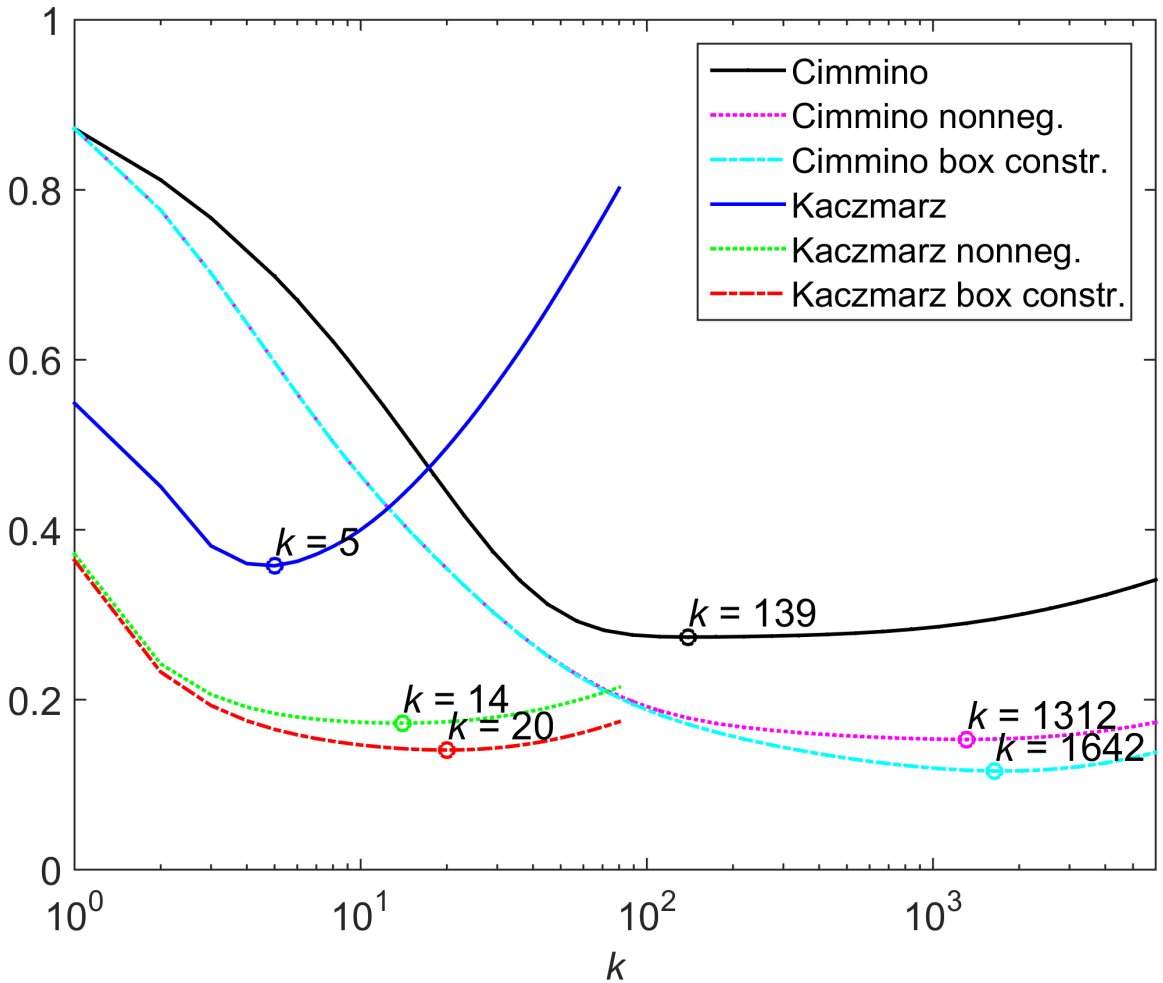
The plain-vanilla or naïve solution $\boldsymbol{x}^{\text{naïve}} = \boldsymbol{A}^{-1}\boldsymbol{b}$ is undesired, because it has a large component coming from the noise in the data:

$$\boldsymbol{x}^{\text{naïve}} = \boldsymbol{A}^{-1}\boldsymbol{b} = \boldsymbol{A}^{-1}(\boldsymbol{A}\,\bar{\boldsymbol{x}} + \boldsymbol{e}) = \bar{\boldsymbol{x}} + \boldsymbol{A}^{-1}\boldsymbol{e}.$$

The component $\boldsymbol{A}^{-1}\boldsymbol{e}$ typically dominates over $\bar{\boldsymbol{x}}$, because $\boldsymbol{A}$ is an ill conditioned matrix.

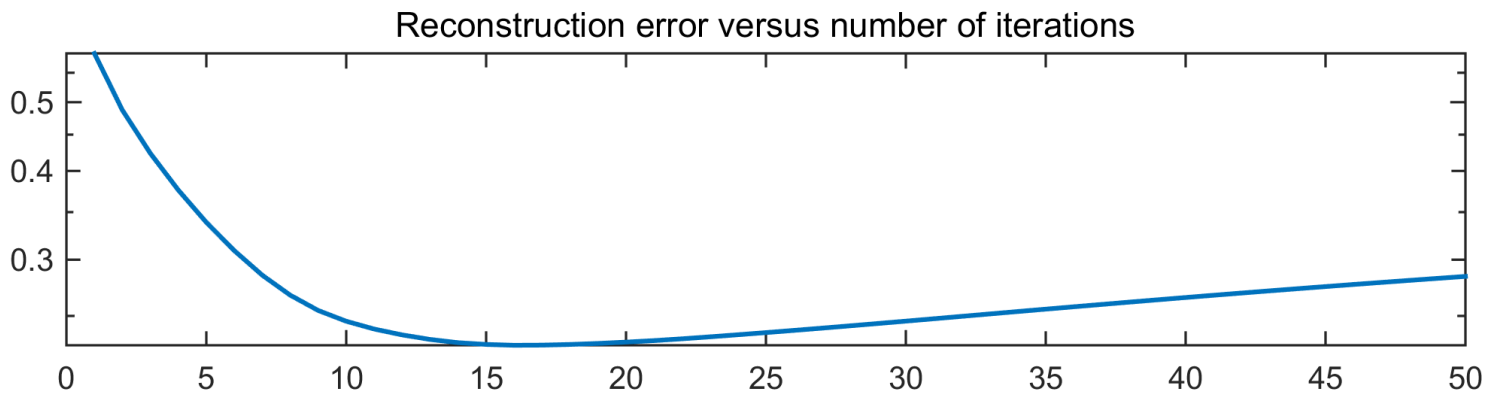
But something interesting happens during the iterations ...

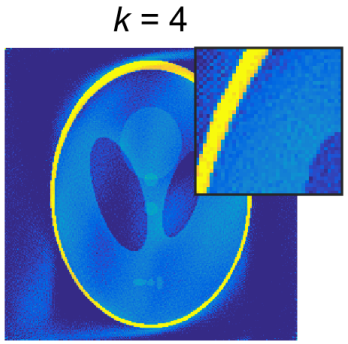
Convergence for Noisy Data

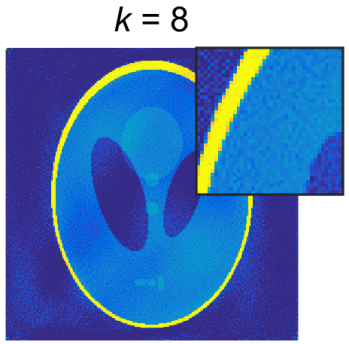


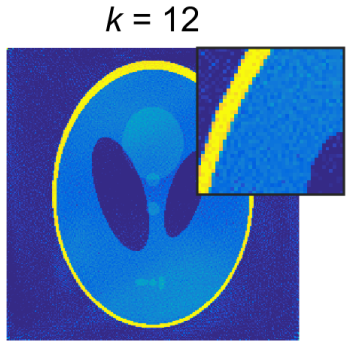
For all six methods the error $\|\bar{\boldsymbol{x}} - \boldsymbol{x}^{(k)}\|_2$ decreases until it reaches a minimum, shown by the circles, after which it starts to increase again.

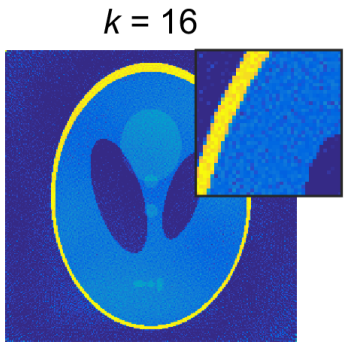
Semi-Convergence For Kaczmarz's Method

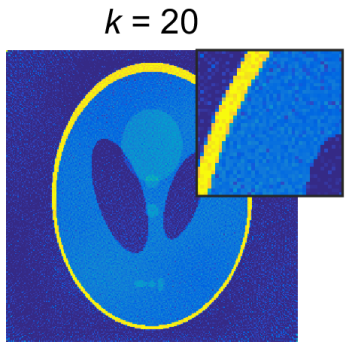


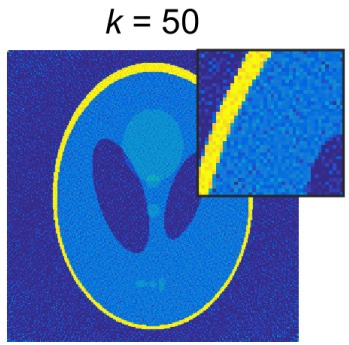












Semi-Convergence

This behavior is often referred to as semi-convergence:

- During the initial iterations, the iteration vector $\boldsymbol{x}^{(k)}$ approaches the desired – but un-obtainable – solution $\bar{\boldsymbol{x}}$ to the noise-free problem.
- During later iterations, $\boldsymbol{x}^{(k)}$ converges to the undesired naïve solution associated with the particular AIR method (i.e., $\boldsymbol{A}^{-1}\boldsymbol{b}$ if the system matrix is invertible).

We want to stop the iterations just when the convergence behavior changes from the former to the latter.

Then we achieve a regularized solution – an approximation to the noise-free solution which is not too perturbed by the noise in the data.

Today we explain

1. why we have semi-convergence for noisy data, and
2. how to stop the iterations at the right time.

Analysis of Landweber's Method I

With an arbitrary starting vector $\boldsymbol{x}^{(0)}$, the kth Landweber iterate is:

$$
\begin{aligned}
\boldsymbol{x}^{(k)} &= \boldsymbol{x}^{(k-1)} + \omega\, \boldsymbol{A}^T \big(\boldsymbol{b} - \boldsymbol{A}\, \boldsymbol{x}^{(k-1)}\big) \\
&= (\boldsymbol{I} - \omega\, \boldsymbol{A}^T \boldsymbol{A})\, \boldsymbol{x}^{(k-1)} + \omega\, \boldsymbol{A}^T \boldsymbol{b} \\
&= (\boldsymbol{I} - \omega\, \boldsymbol{A}^T \boldsymbol{A}) \left[(\boldsymbol{I} - \omega\, \boldsymbol{A}^T \boldsymbol{A})\, \boldsymbol{x}^{(k-2)} + \omega\, \boldsymbol{A}^T \boldsymbol{b} \right] + \omega\, \boldsymbol{A}^T \boldsymbol{b} \\
&= (\boldsymbol{I} - \omega\, \boldsymbol{A}^T \boldsymbol{A})^2\, \boldsymbol{x}^{(k-2)} + \big((\boldsymbol{I} - \omega\, \boldsymbol{A}^T \boldsymbol{A}) + \boldsymbol{I} \big)\, \omega\, \boldsymbol{A}^T \boldsymbol{b} \\
&= (\boldsymbol{I} - \omega\, \boldsymbol{A}^T \boldsymbol{A})^3\, \boldsymbol{x}^{(k-3)} + \big((\boldsymbol{I} - \omega\, \boldsymbol{A}^T \boldsymbol{A})^2 + (\boldsymbol{I} - \omega\, \boldsymbol{A}^T \boldsymbol{A}) + \boldsymbol{I} \big) \omega\, \boldsymbol{A}^T \boldsymbol{b} \\
&= \cdots \\
&= (\boldsymbol{I} - \omega\, \boldsymbol{A}^T \boldsymbol{A})^k \boldsymbol{x}^{(0)} + \\
&\qquad \left[(\boldsymbol{I} - \omega\, \boldsymbol{A}^T \boldsymbol{A})^{k-1} + (\boldsymbol{I} - \omega\, \boldsymbol{A}^T \boldsymbol{A})^{k-2} + \cdots + \boldsymbol{I} \right] \omega\, \boldsymbol{A}^T \boldsymbol{b} \\
&= (\boldsymbol{I} - \omega\, \boldsymbol{A}^T \boldsymbol{A})^k \boldsymbol{x}^{(0)} + \sum_{j=0}^{k-1} (\boldsymbol{I} - \omega\, \boldsymbol{A}^T \boldsymbol{A})^j\, \omega\, \boldsymbol{A}^T \boldsymbol{b}.
\end{aligned}
$$

Analysis of Landweber's Method II

For simplicity we now assume that $\boldsymbol{x}^{(0)} = 0$. We insert the SVD of the system matrix $\boldsymbol{A} = \boldsymbol{U}\,\boldsymbol{\Sigma}\,\boldsymbol{V}^T$ and use $\boldsymbol{I} = \boldsymbol{V}\,\boldsymbol{V}^T$:

$$\boldsymbol{x}^{(k)} = \boldsymbol{V} \sum_{j=0}^{k-1} (\boldsymbol{I} - \omega\,\boldsymbol{\Sigma}^2)^j\,\omega\,\boldsymbol{\Sigma}\,\boldsymbol{U}^T\boldsymbol{b} = \boldsymbol{V}\,\boldsymbol{\Phi}^{(k)}\,\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T\boldsymbol{b},$$

where we introduced the $n \times n$ diagonal matrix

$$\boldsymbol{\Phi}^{(k)} = \sum_{j=0}^{k-1} (\boldsymbol{I} - \omega\,\boldsymbol{\Sigma}^2)^j\,\omega\,\boldsymbol{\Sigma}^2 = \omega\,\boldsymbol{\Sigma}^2 \sum_{j=0}^{k-1} (\boldsymbol{I} - \omega\,\boldsymbol{\Sigma}^2)^j = \begin{pmatrix} \phi_1^{(k)} & & \\ & \phi_2^{(k)} & \\ & & \ddots \end{pmatrix}$$

with diagonal elements

$$\phi_i^{(k)} = \omega\,\sigma_i^2 \sum_{j=0}^{k-1} (1 - \omega\,\sigma_i^2)^j, \qquad i = 1, 2, \ldots, n.$$

Analysis of Landweber's Method III

The sum $\sum_{j=0}^{k-1} (1 - \omega\,\sigma_i^2)^j$ is a geometric series:

$$\sum_{j=0}^{k-1} z^j = (1 - z^k)/(1 - z),$$

and thus for $i = 1, 2, \ldots, n$ we have:

$$\phi_i^{(k)} = \omega\,\sigma_i^2 \sum_{j=0}^{k-1} (1 - \omega\,\sigma_i^2)^j = \omega\,\sigma_i^2\, \frac{1 - (1 - \omega\,\sigma_i^2)^k}{1 - (1 - \omega\,\sigma_i^2)} = 1 - (1 - \omega\,\sigma_i^2)^k$$

leading to a simple expression for the diagonal Landweber filter matrix

$$\boldsymbol{\Phi}^{(k)} = \begin{pmatrix} 1 - (1 - \omega\,\sigma_1^2)^k & & \\ & 1 - (1 - \omega\,\sigma_2^2)^k & \\ & & \ddots \end{pmatrix}.$$

SVD Expression of Landweber Iteration Vectors

After k iterations we have obtained a regularized solution

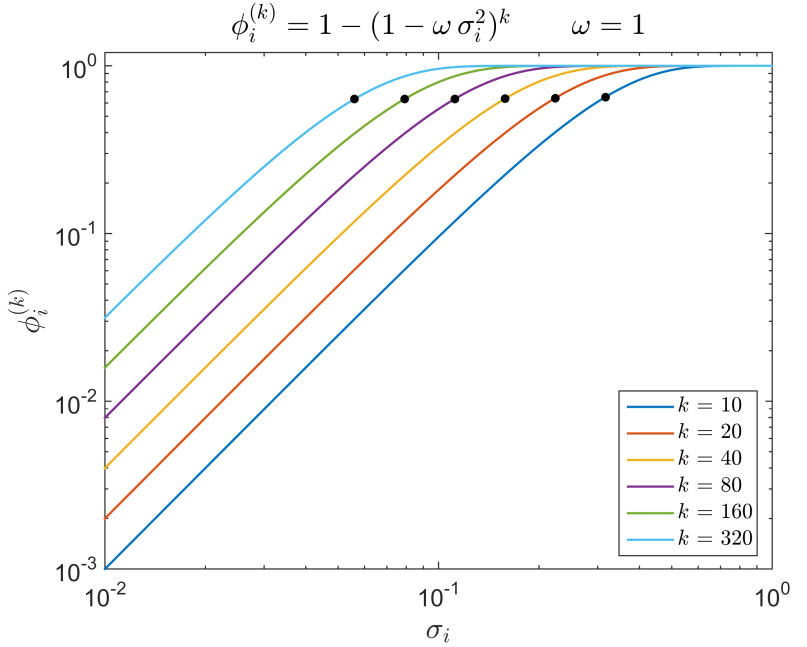
$$\boldsymbol{x}^{(k)} = \boldsymbol{V}\,\boldsymbol{\Phi}^{(k)}\,\boldsymbol{\Sigma}^{-1}\,\boldsymbol{U}^T\boldsymbol{b} = \sum_{i=1}^{n} \phi_i^{(k)}\,\frac{\boldsymbol{u}_i^T\boldsymbol{b}}{\sigma_i}\,\boldsymbol{v}_i$$

$$\phi_i^{(k)} = 1 - (1 - \omega\,\sigma_i^2)^k \approx \begin{cases} 1 & \text{for} \quad \sigma_i \gg \omega, \\ k\,\omega\,\sigma_i^2 & \text{for} \quad \sigma_i \ll \omega, \end{cases}$$

SVD components for large σ_i are essentially unfiltered; those for small σ_i are damped by a factor $\propto \sigma_i^2$.

The breakpoint is approximately for $\sigma_i \approx 1/\sqrt{k\,\omega}$, see black dots $\bullet \rightarrow$

More SVD components are included as we perform more iterations ($\sim$ TSVD).



About The Relaxation Parameter

This analysis also provides an asymptotic convergence analysis for Landweber's method as $k \to \infty$.

For the geometric series to converge we must require that the relaxation parameter ω satisfies

$$|1 - \omega\,\sigma_i^2| < 1, \qquad i = 1, 2, \ldots, m,$$

which implies that we must have

$$\omega < 2/\sigma_1^2 = 2/\|\boldsymbol{A}\|_2^2 = 2/\|\boldsymbol{A}^T\boldsymbol{A}\|_2.$$

When this is satisfied then $\phi_i^{(k)} \to 1$ for all i and thus $\boldsymbol{\Phi}^{(k)} \to \boldsymbol{I}$ for $k \to \infty$.

Hence $\boldsymbol{x}^{(k)}$ converges to the naïve noisy solution $\boldsymbol{V}\,\boldsymbol{\Sigma}^{-1}\,\boldsymbol{U}^T\boldsymbol{b} = \boldsymbol{A}^{-1}\boldsymbol{b}$.

Iteration Error and Noise Error

Get ready for semi-convergence analysis!

We can always split the reconstruction error for $\boldsymbol{x}^{(k)}$ into two components:

$$\bar{\boldsymbol{x}} - \boldsymbol{x}^{(k)} = \left(\bar{\boldsymbol{x}} - \bar{\boldsymbol{x}}^{(k)}\right) + \left(\bar{\boldsymbol{x}}^{(k)} - \boldsymbol{x}^{(k)}\right).$$

The "clean" iteration vector $\bar{\boldsymbol{x}}^{(k)}$ is defined as the iteration vector obtained when we apply k steps of Landweber's method to the noise-free data $\boldsymbol{A}\,\bar{\boldsymbol{x}}$.

- The first component $\bar{\boldsymbol{x}} - \bar{\boldsymbol{x}}^{(k)}$ is the iteration error which is an approximation error caused by the finite number of iterations, and which is independent of the noise in the data.

- The second component $\bar{\boldsymbol{x}}^{(k)} - \boldsymbol{x}^{(k)}$ is the noise error which is due to the presence of the data errors and causing the actual iteration vector $\boldsymbol{x}^{(k)}$ to differ from the "clean" iteration vector $\bar{\boldsymbol{x}}^{(k)}$.

SVD Analysis of Iteration Error and Noise Error

$$\begin{aligned} \bar{\boldsymbol{x}} &= \sum_{i=1}^{n} \boldsymbol{v}_i^T \bar{\boldsymbol{x}}\, \boldsymbol{v}_i, \\ \boldsymbol{u}_i^T \boldsymbol{b} &= \boldsymbol{u}_i^T (\boldsymbol{A}\bar{\boldsymbol{x}} + \boldsymbol{e}) = \boldsymbol{u}_i^T (\boldsymbol{A}\bar{\boldsymbol{x}}) + \boldsymbol{u}_i^T \boldsymbol{e}, \\ \boldsymbol{u}_i^T (\boldsymbol{A}\bar{\boldsymbol{x}}) &= \boldsymbol{u}_i^T \sum_{j=1}^{n} \boldsymbol{u}_j \sigma_j \boldsymbol{v}_j^T \bar{\boldsymbol{x}} = \sigma_i \boldsymbol{v}_i^T \bar{\boldsymbol{x}}. \end{aligned}$$

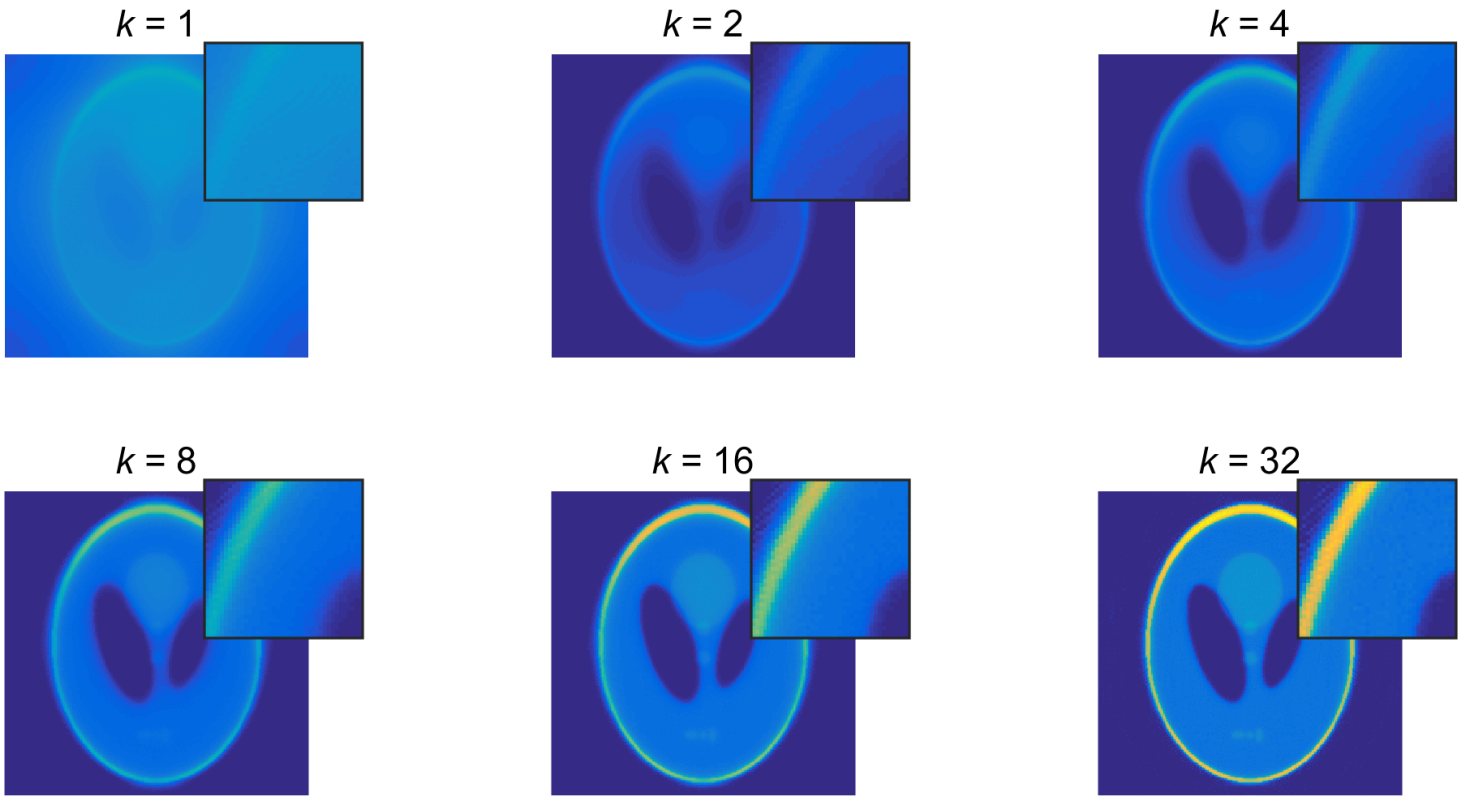
It follows that

$$\bar{\boldsymbol{x}} - \bar{\boldsymbol{x}}^{(k)} = \sum_{i=1}^{n} \boldsymbol{v}_i^T \bar{\boldsymbol{x}}\, \boldsymbol{v}_i - \sum_{i=1}^{n} \phi_i^{(k)} \frac{\boldsymbol{u}_i^T (\boldsymbol{A}\bar{\boldsymbol{x}})}{\sigma_i} \boldsymbol{v}_i = \sum_{i=1}^{n} (1 - \phi_i^{(k)}) \boldsymbol{v}_i^T \bar{\boldsymbol{x}}\, \boldsymbol{v}_i,$$

$$\boldsymbol{x}^{(k)} - \bar{\boldsymbol{x}}^{(k)} = \sum_{i=1}^{n} \phi_i^{(k)} \frac{\boldsymbol{u}_i^T \boldsymbol{b}}{\sigma_i} \boldsymbol{v}_i - \sum_{i=1}^{n} \phi_i^{(k)} \frac{\boldsymbol{u}_i^T (\boldsymbol{A}\bar{\boldsymbol{x}})}{\sigma_i} \boldsymbol{v}_i = \sum_{i=1}^{n} \phi_i^{(k)} \frac{\boldsymbol{u}_i^T \boldsymbol{e}}{\sigma_i} \boldsymbol{v}_i.$$

As k increases, and more $\phi_i^{(k)}$ approach 1, the iteration error tends to zero while the noise error increases because more noise components are included.

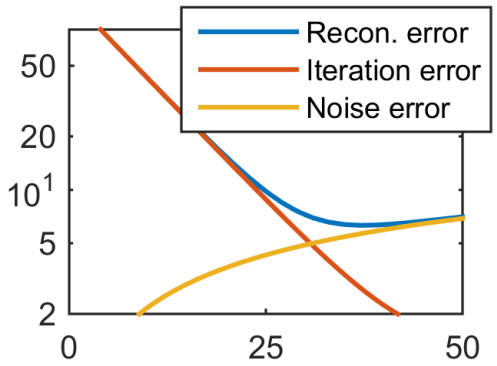
Progression of the Iteration Error



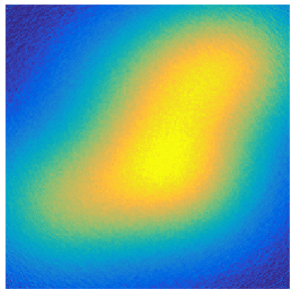
"Clean" iteration vectors $\bar{\boldsymbol{x}}^{(k)}$ for Landweber applied to noise-free data $\boldsymbol{A}\,\bar{\boldsymbol{x}}$.

As we take more iterations we include more SVD components with higher frequencies and we obtain sharper edges in the images.

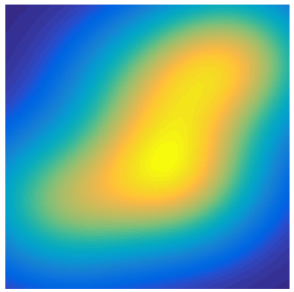
Progression of All Error Types

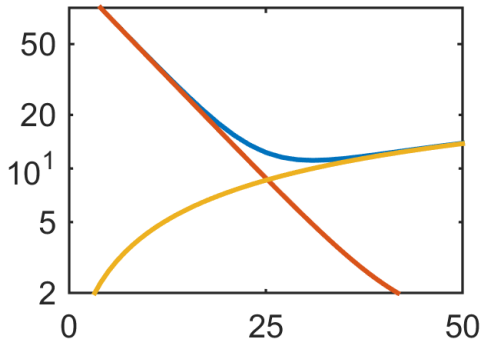


$\rho = 0.01$, $k = 37$

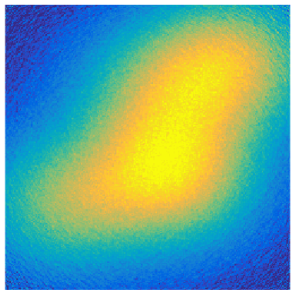


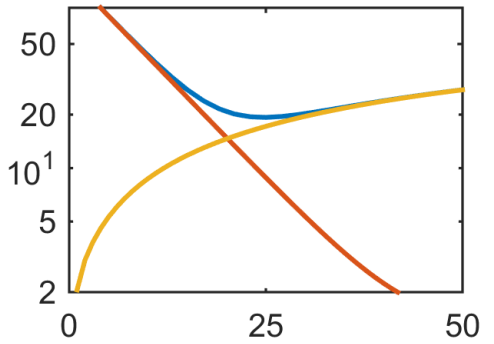
Phantom



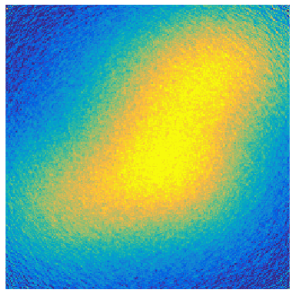


$\rho = 0.02$, $k = 31$





$\rho = 0.04$, $k = 25$



The iteration error $\|\bar{\boldsymbol{x}} - \bar{\boldsymbol{x}}^{(k)}\|_2$ is independent of the noise.

The noise error $\|\bar{\boldsymbol{x}}^{(k)} - \boldsymbol{x}^{(k)}\|_2$ increases when the noise increases.

The combined reconstruction error $\|\bar{\boldsymbol{x}} - \boldsymbol{x}\ (k)\|_2$ has a minimum $\mapsto$ semi-convergence.

Landweber's Method with Projections

When projections are incorporated in the iterative algorithm, we can no longer perform an SVD analysis. But it can be shown that:

$$\begin{aligned} \|\bar{\boldsymbol{x}} - \bar{\boldsymbol{x}}^{(k)}\|_2 &\leq \left(1 - \omega\,\sigma_n^2\right)^k \|\bar{\boldsymbol{x}}\|_2, \\ \|\bar{\boldsymbol{x}}^{(k)} - \boldsymbol{x}^{(k)}\|_2 &\leq \frac{1 - (1 - \omega\sigma_n^2)^k}{\sigma_n^2}\, \|\boldsymbol{A}\|_2\, \|\boldsymbol{e}\|_2. \end{aligned}$$

Here σ_n is the smallest singular value of $\boldsymbol{A}$; hence $\omega\,\sigma_n^2$ is quite small, but the iteration error usually decreases faster.

From $(1-\epsilon)^k = 1 - k\epsilon + {}^1\!/_2 k(k+1)\epsilon^2 + \cdots$ it follows that

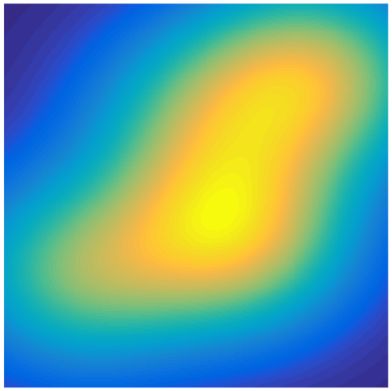
$$\frac{1 - (1 - \omega\sigma_n^2)^k}{\sigma_n^2} = \frac{1 - (1 - k\,\omega\sigma_n^2 + O(\sigma_n^4))}{\sigma_n^2} = k\,\omega + O(\sigma_n^2)$$

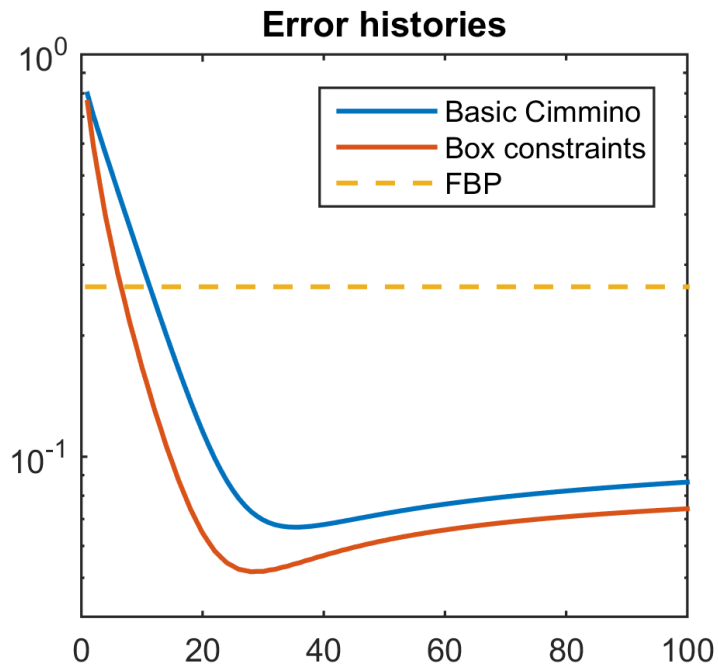
and we obtain the approximate bound for the noise error:

$$\boxed{\|\bar{\boldsymbol{x}}^{(k)} - \boldsymbol{x}^{(k)}\|_2 \lesssim k\,\omega\,\|\boldsymbol{A}\|_2\,\|\boldsymbol{e}\|_2.}$$

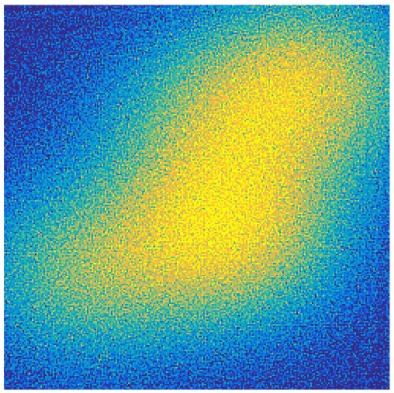
Projected Cimmino, Example I

Ground truth

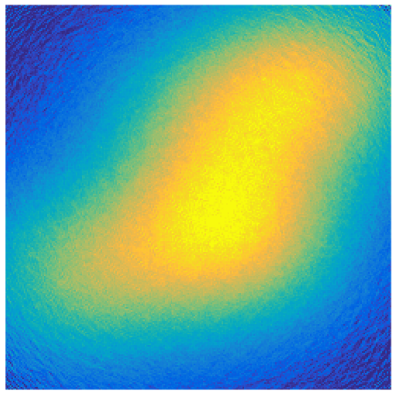




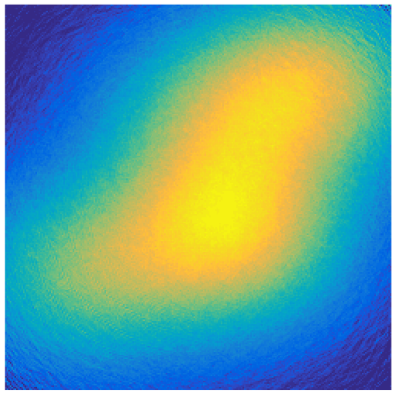
FBP, 0.26442



Basic, 0.066812

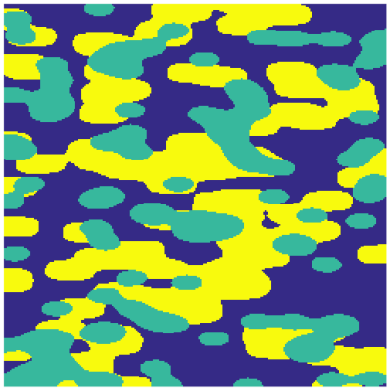


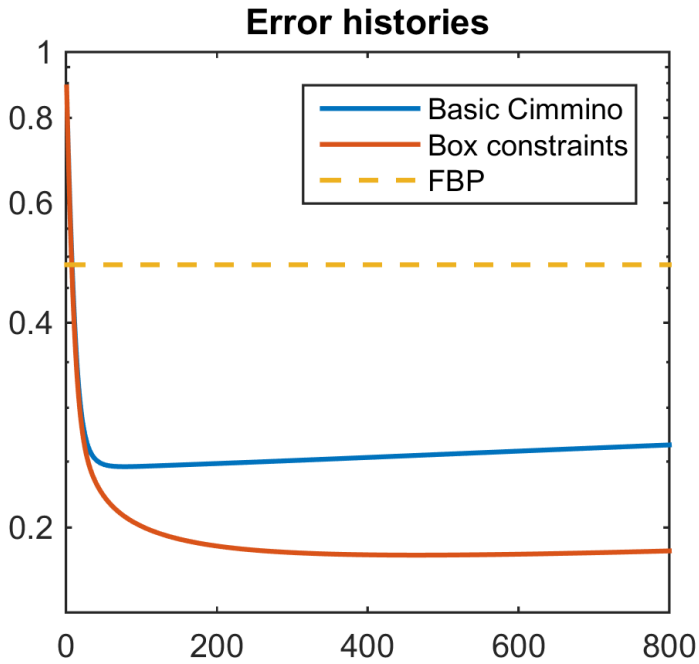
Box, 0.051752



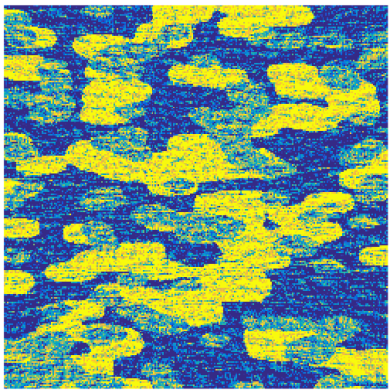
Projected Cimmino, Example II

Ground truth

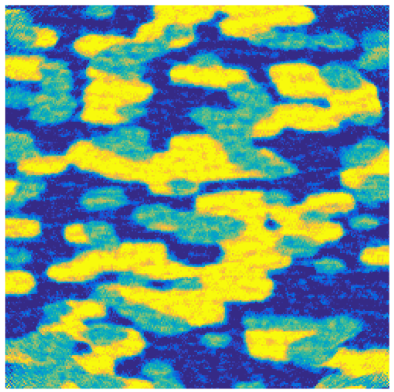




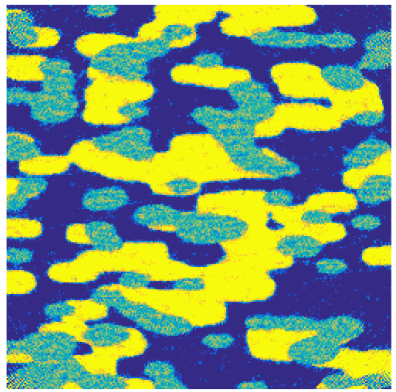
FBP, 0.48662



Basic, 0.24574



Box, 0.18214



Analysis of Kaczmarz's Method

To study the noise error in Kaczmarz's method we take this approach.

Define the "splitting"

$$\boldsymbol{A}\,\boldsymbol{A}^T = \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{L}^T,$$

$\boldsymbol{D}$ = diagonal matrix with diagonal elements of $\boldsymbol{A}\,\boldsymbol{A}^T$

$\boldsymbol{L}$ = strictly lower triangular matrix (zeros on the diagonal).

Also define the lower triangular matrix

$$\widehat{\boldsymbol{L}} = (\boldsymbol{D} + \omega\,\boldsymbol{L})^{-1}.$$

Then one iteration of Kaczmarz's method can be written as

$$\boldsymbol{x}^{(k)} = P_{\mathcal{C}}\big(\boldsymbol{x}^{(k-1)} + \omega\,\boldsymbol{A}^T\,\widehat{\boldsymbol{L}}\,(\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}^{(k-1)})\big)$$

This is for *purely theoretical use;* it should not be used for computations!

Upper Bound for Iteration Error

Following the same approach as for Landweber, we obtain:

$$\boldsymbol{x}^{(k)} - \bar{\boldsymbol{x}}^{(k)} = \sum_{j=0}^{k-1} (\boldsymbol{I} - \omega\, \boldsymbol{A}^T\, \widehat{\boldsymbol{L}}\, \boldsymbol{A})^j\, \boldsymbol{A}^T \widehat{\boldsymbol{L}}\, \boldsymbol{b},$$

and it can be shown that for both the un-constrained and the constrained problem, the noise error is bounded as

$$\|\bar{\boldsymbol{x}}^{(k)} - \boldsymbol{x}^{(k)}\|_2 \leq \frac{1 - (1 - \omega\, \varsigma^2)^k}{\varsigma^2}\, \|\boldsymbol{A}^T\, \widehat{\boldsymbol{L}}\, \boldsymbol{e}\|_2 + O(\varsigma^2),$$

where ς is the smallest nonzero singular value of the matrix $\boldsymbol{D}^{1/2}\, \widehat{\boldsymbol{L}}\, \boldsymbol{A}$.

Following the previous arguments, we again obtain an approximate upper bound of the form

$$\boxed{\|\bar{\boldsymbol{x}}^{(k)} - \boldsymbol{x}^{(k)}\|_2 \lesssim k\, \omega\, \|\boldsymbol{A}^T\, \widehat{\boldsymbol{L}}\, \boldsymbol{e}\|_2.}$$

Can We Do Something About $\|\boldsymbol{A}^T \widehat{\boldsymbol{L}}\, \boldsymbol{e}\|_2$?

Can we replace $\alpha = \|\boldsymbol{A}^T \widehat{\boldsymbol{L}}\, \boldsymbol{e}\|_2$ with the upper bound $\beta = \|\boldsymbol{A}\|_2 \, \|\widehat{\boldsymbol{L}}\|_2 \, \|\boldsymbol{e}\|_2$?

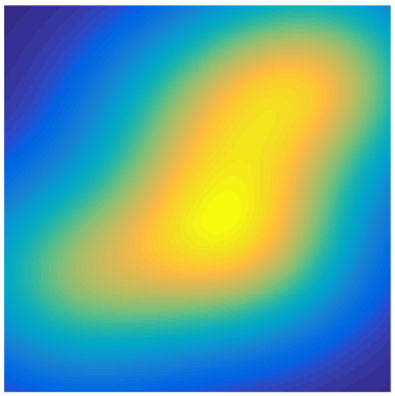
Example: $N = 64$, no. detector pixels $= 90$, length of detector $= L$ and $\sqrt{2}L$ where L is the object's side length.

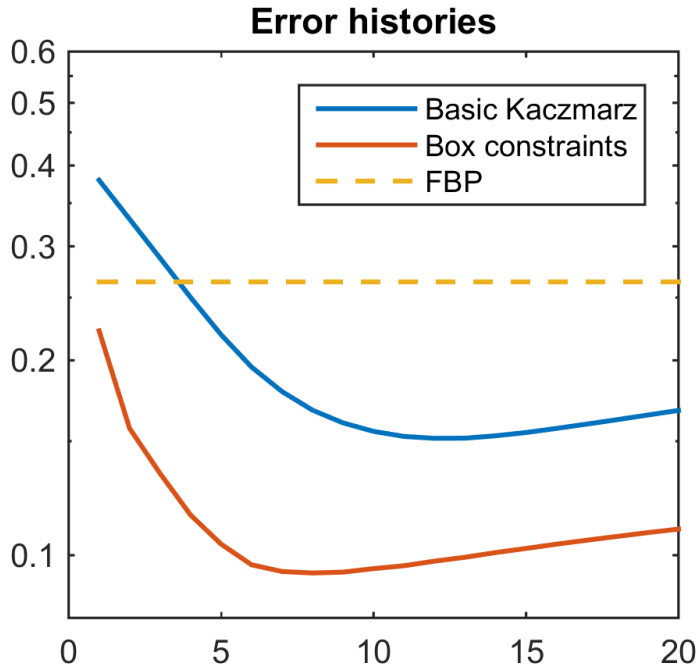
projection angles	$\|\boldsymbol{e}\|_2$	short detector		long detector	
		α	β	α	β
$\theta = 1°, 2°, 3°, \ldots, 180°$	49	7.6	1251	28.2	$6.4 \cdot 10^7$
$\theta = 3°, 6°, 9°, \ldots, 180°$	70	7.4	403	22.6	$2.1 \cdot 10^7$
$\theta = 6°, 12°, 18°, \ldots, 180°$	128	6.1	181	7.1	2232

Conclusion: $\beta \gg \alpha$; not a good idea to use β.

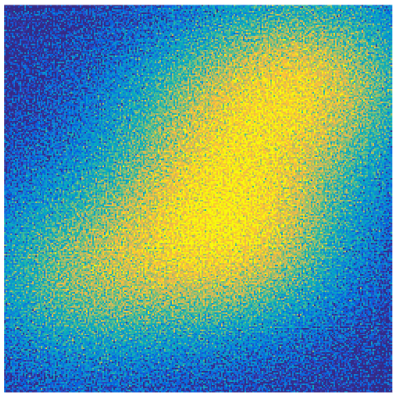
Kaczmarz Semi-Convergence, Example I

Ground truth

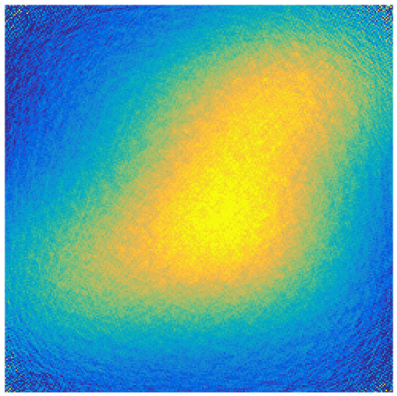




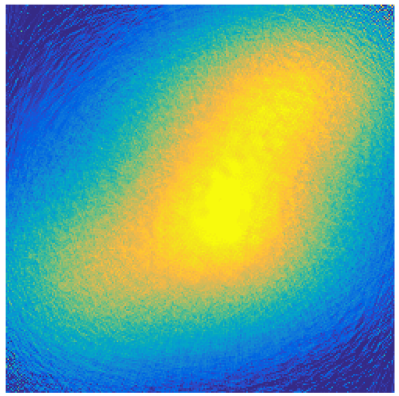
FBP, 0.26442



Basic, 0.15157

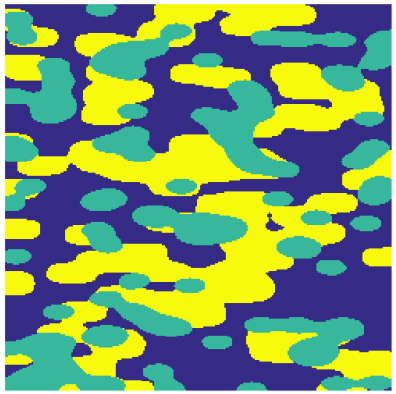


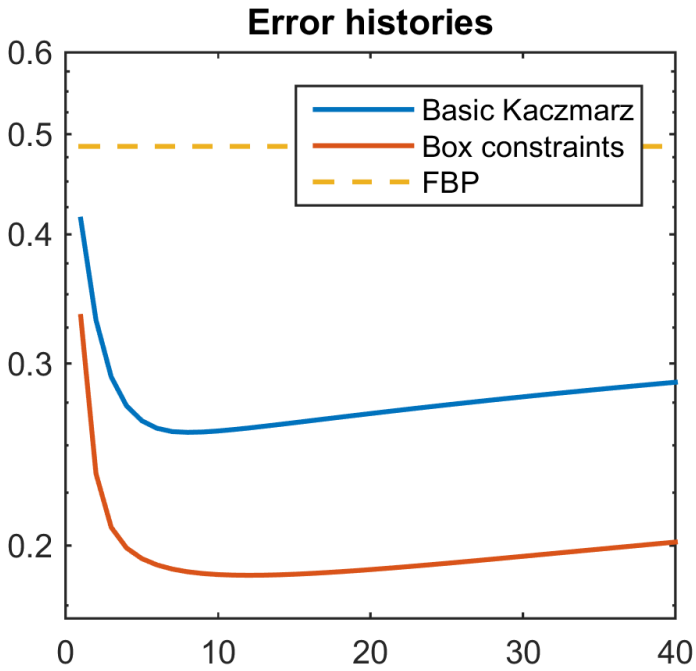
Box, 0.093865



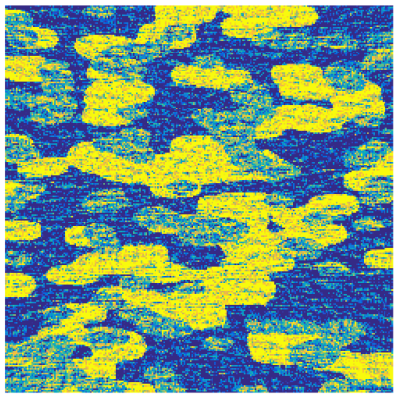
Kaczmarz Semi-Convergence, Example II

Ground truth

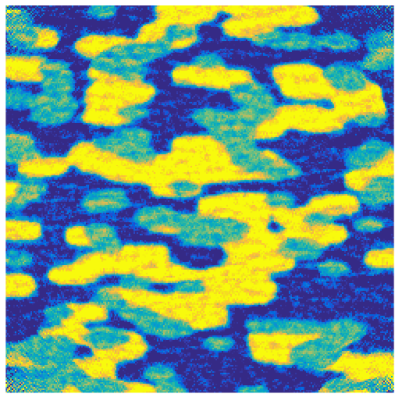




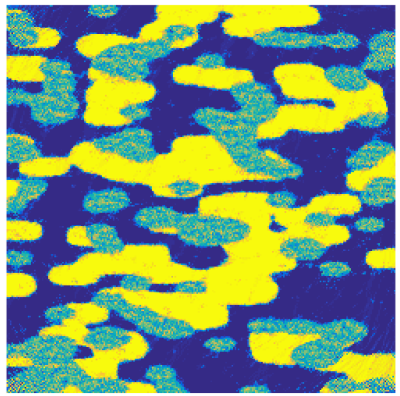
FBP, 0.48662



Basic, 0.25735



Box, 0.18716



Stopping Rules

To successfully use the iterative methods for noisy data, we obviously need an automatic method – a stopping rule – for terminating the iterations at, or near, the point of semi-convergence where the reconstruction error $\bar{\boldsymbol{x}} - \boldsymbol{x}^{(k)}$ is as small as possible.

We must be able to do so without knowing the ground truth $\bar{\boldsymbol{x}}$.

The decision must be made from available information, such as the kth iterate $\boldsymbol{x}^{(k)}$ and/or its corresponding residual $\boldsymbol{\varrho}^{(k)} = \boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}^{(k)}$.

Such stopping rules are studied frequently by mathematicians, but they do not achieve the same attention in the tomographic reconstruction communities ...

Our discussion of stopping rules is based on:
P. C. Hansen, J. S. Jørgensen, and P. W. Rasmussen, *Stopping rules for algebraic iterative reconstruction methods in computed tomography;* in 21st International Conference on Computational Science and Its Applications (ICCSA), IEEE (2021), pp. 60–70, doi 10.1109/ICCSA54496.2021.00019.

On the Need for Stopping Rules

- The noise error $\bar{\boldsymbol{x}}^{(k)} - \boldsymbol{x}^{(k)}$ often grows slowly with the number of iterations k. Hence the error history exhibits a *flat minimum*, and it is not crucial to stop at a very specific number of iterations.
- There are many applications for which the users have build a very *good intuition* of approximately how many iterations are needed to obtain a satisfactory reconstruction.
- When very many iterations are needed and the minimum is very flat, the iterations are often terminated with one's *patience runs out* – and hence one may not observe the semi-convergence effect.

Developing a robust stopping rule that works on many types of problems and for many kinds of data is difficult/impossible.

Here we give an overview of successful stopping rules, and then the user can try these methods on a given problem.

A Bit of Statistical Notation

To make precise statements in this section, we need a small amount of statistical framework and notation.

The exact noise-free data $\bar{\boldsymbol{b}}$ that correspond to the ground truth image:

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

The elements of the noise vector $\boldsymbol{e} \in \mathbb{R}^m$ are random variables, i.e., their values depend on a set of well-defined random events. The vector of expected values $\mathcal{E}(\boldsymbol{e})$ and the covariance matrix $\mathcal{V}(\boldsymbol{e})$ are defined as

$$\mathcal{E}(\boldsymbol{e}) = \begin{pmatrix} \mathcal{E}(e_1) \\ \mathcal{E}(e_2) \\ \vdots \end{pmatrix}, \qquad \mathcal{V}(\boldsymbol{e}) = \mathcal{E}\Big(\big(\boldsymbol{e} - \mathcal{E}(\boldsymbol{e})\big)\,\big(\boldsymbol{e} - \mathcal{E}(\boldsymbol{e})\big)^T \Big).$$

We restrict our analysis to *white Gaussian noise* with zero mean:

$$\mathcal{E}(\boldsymbol{e}) = 0, \qquad \mathcal{V}(\boldsymbol{e}) = \eta^2 \boldsymbol{I}, \qquad \mathcal{E}(\|\boldsymbol{e}\|_2^2) = m\,\eta^2,$$

where η is the standard deviation of the noise.

The Discrepancy Principle (DP)

DP: a model's output should fit the data "to the noise level."

This translates into a stopping rule where we choose the number of iterations k such that the residual $\boldsymbol{\varrho}^{(k)} = \boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}^{(k)}$ is "of the same size" as the noise vector $\boldsymbol{e}$:

$$\|\boldsymbol{\varrho}^{(k)}\|_2^2 \approx m\,\eta^2.$$

We return to methods for estimating the standard deviation η from data.

Most authors include a constant $\tau \geq 1$ such that the above condition takes the form

$$\|\boldsymbol{\varrho}^{(k)}\|_2^2 \approx \tau\,m\,\eta^2.$$

The constant τ can be useful when we have only a rough estimate of η and there is a risk that we take too many or too few iterations.

As we shall see, this stopping rule is quite dubious.

How We Really Should Fit to the Noise Level I

To learn more about this principle, consider the TSVD solution

$$\boldsymbol{x}_k = \sum_{i=1}^{k} \frac{\boldsymbol{u}_i^T \boldsymbol{b}}{\sigma_i} \boldsymbol{v}_i \qquad \text{with} \qquad \boldsymbol{A} = \sum_{i=1}^{n} \boldsymbol{u}_i \, \sigma_i \, \boldsymbol{v}_i^T .$$

The corresponding residual takes the form

$$\boldsymbol{\varrho}_k \equiv \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k = \sum_{i=k+1}^{m} (\boldsymbol{u}_i^T \boldsymbol{b}) \, \boldsymbol{u}_i = \boldsymbol{P}_k \, \boldsymbol{b} = \boldsymbol{P}_k \, \bar{\boldsymbol{b}} + \boldsymbol{P}_k \, \boldsymbol{e}.$$

where the projection matrix $\boldsymbol{P}_k = \sum_{i=k+1}^{m} \boldsymbol{u}_i \boldsymbol{u}_i^T$ projects onto the subspace spanned by $\boldsymbol{u}_{k+1}, \ldots, \boldsymbol{u}_m$.

The two components of $\boldsymbol{\varrho}_k$ take the form

$$\boldsymbol{P}_k \, \bar{\boldsymbol{b}} = \sum_{i=k+1}^{m} (\boldsymbol{u}_i^T \bar{\boldsymbol{b}}) \, \boldsymbol{u}_i \qquad \text{and} \qquad \boldsymbol{P}_k \, \boldsymbol{e} = \sum_{i=k+1}^{m} (\boldsymbol{u}_i^T \boldsymbol{e}) \, \boldsymbol{u}_i$$

How We Really Should Fit to the Noise Level II

The noise-free right-hand side's SVD components $\boldsymbol{u}_i^T \bar{\boldsymbol{b}}$ have a decaying magnitude as k increases. Hence $\|\boldsymbol{P}_k \bar{\boldsymbol{b}}\|_2$, which always decreases with k, will decrease quite fast because it's largest SVD components are extracted first (for the small values of k).

The noise component's norm is given by

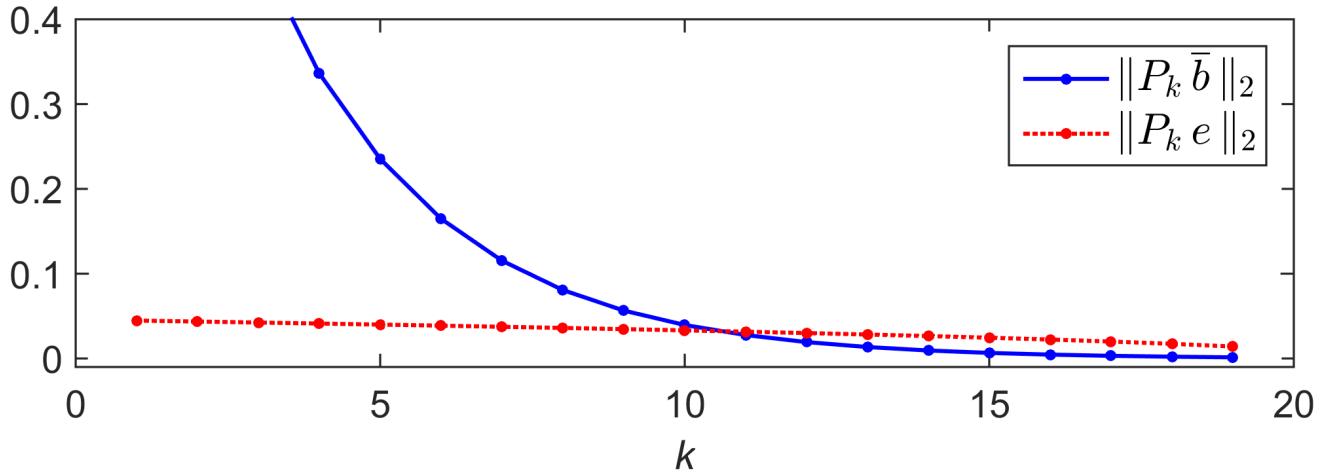
$$\|\boldsymbol{P}_k\, \boldsymbol{e}\|_2^2 = \sum_{i=k+1}^{m} (\boldsymbol{u}_i^T \boldsymbol{e})^2.$$

Since $\boldsymbol{e}$ is zero-mean white Gaussian noise, the quantities $\boldsymbol{u}_i^T \boldsymbol{e}$ also follow a Gaussian distribution with standard deviation η; hence:

$$\mathcal{E}\left(\|\boldsymbol{P}_k\, \boldsymbol{e}\|_2^2\right) = \mathcal{E}\left(\sum_{i=k+1}^{m} (\boldsymbol{u}_i^T \boldsymbol{e})^2\right) = \sum_{i=k+1}^{m} \mathcal{E}\left((\boldsymbol{u}_i^T \boldsymbol{e})^2\right) = (m-k)\,\eta^2.$$

The factor $m-k$ reflects the fact that the vector $\boldsymbol{P}_k\, \boldsymbol{e}$ lies in a subspace of that dimension and thus has $m-k$ degrees of freedom. The norm $\|\boldsymbol{P}_k\, \boldsymbol{e}\|_2$ also decays with k, and compared to $\|\boldsymbol{P}_k\, \bar{\boldsymbol{b}}\|_2$ it decays rather slowly.

How We Really Should Fit to the Noise Level III



- k too small: we have not captured enough SVD components; $\boldsymbol{A}\,\boldsymbol{x}_k$ is not a good approximation the exact data $\bar{\boldsymbol{b}}$, ϱ_k is dominated by $\boldsymbol{P}_k\,\bar{\boldsymbol{b}}$, and $\|\boldsymbol{P}_k\,\bar{\boldsymbol{b}}\|_2$ is larger than $\|\boldsymbol{P}_k\,\boldsymbol{e}\|_2$.
- k "just about right": $\boldsymbol{A}\,\boldsymbol{x}_k$ approximates $\bar{\boldsymbol{b}}$ as well as possible; the norm $\|\boldsymbol{P}_k\,\bar{\boldsymbol{b}}\|_2$ has now become smaller and is of the same size as $\|\boldsymbol{P}_k\,\boldsymbol{e}\|_2$.
- k too large: the residual ϱ_k is still dominated by the noise component $\boldsymbol{P}_k\,\boldsymbol{e}$, and hence $\|\boldsymbol{P}_k\,\boldsymbol{e}\|_2$ dominates the residual norm.

Strategy: choose k such that $\|\boldsymbol{P}_k\,\bar{\boldsymbol{b}}\|_2 \approx \|\boldsymbol{P}_k\,\boldsymbol{e}\|_2$. But both are unknown, so in practise we should choose k such that $\|\varrho_k\|_2^2 \approx (m-k)\,\eta^2$.

Formalization of The Heuristic

This heuristic reasoning was formalized by several authors; we summarize the main results our iterative methods. From the previous analysis it follows that we can write the kth Landweber iterate as

$$\boldsymbol{x}^{(k)} = \boldsymbol{A}_k^{\#} \boldsymbol{b} \qquad \text{with} \qquad \boldsymbol{A}_k^{\#} = \boldsymbol{V}\, \boldsymbol{\Phi}^{(k)}\, \boldsymbol{\Sigma}^{-1}\, \boldsymbol{U}^T.$$

The data predicted by the $\boldsymbol{x}^{(k)}$ is given by $\boldsymbol{b}_k = \boldsymbol{A}\, \boldsymbol{x}^{(k)} = \boldsymbol{A}\boldsymbol{A}_k^{\#} \boldsymbol{b}$. The matrix $\boldsymbol{A}\boldsymbol{A}_k^{\#}$ that transform the given, noisy data into this prediction is called the influence matrix.

With white Gaussian noise, it can be shown that at the optimal k we have

$$\mathcal{E}(\|\boldsymbol{\varrho}^{(k)}\|_2^2) = \eta^2\,(m - t_k), \qquad t_k = \text{trace}(\boldsymbol{A}\boldsymbol{A}_k^{\#}) = \sum_{i=1}^{n} \phi_i^{(k)},$$

where $\phi_i^{(k)}$ are the filter factors. The real number $m - t_k$ is the effective (or equivalent) degrees of freedom in the residual.

For TSVD the filter factors are 0's and 1's, we simply have $t_k = k$.

Stop Rule: Fit to Noise Level

Stop at the smallest k for which $\|\boldsymbol{\varrho}^{(k)}\|_2^2 \leq \eta^2\,(m - t_k)$.

For our iterative methods, this is particularly convenient because the residual norm $\|\boldsymbol{\varrho}^{(k)}\|_2$ decreases monotonically with k. To see this, we write the residual vector in terms of the SVD:

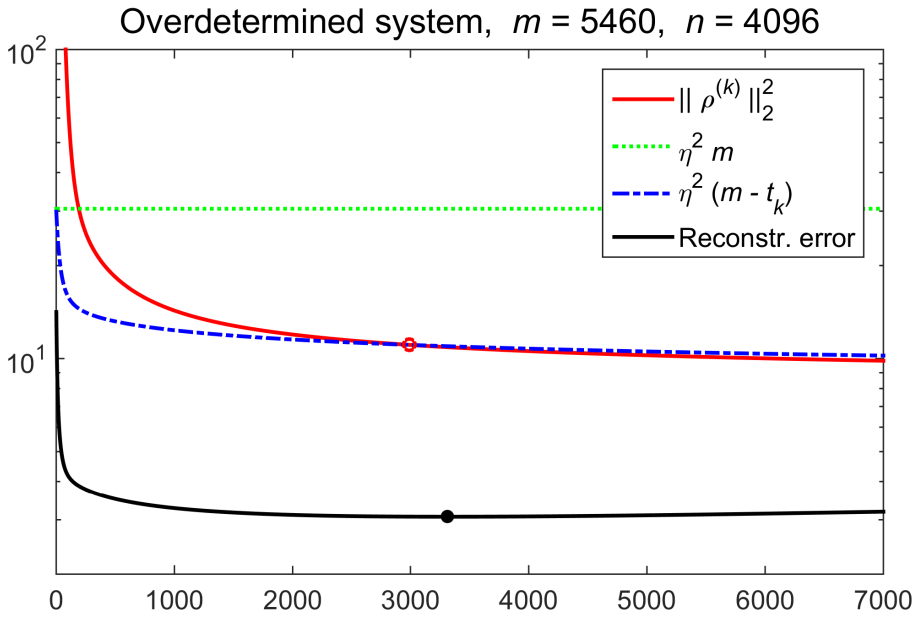
$$\boldsymbol{\varrho}^{(k)} = \boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}^{(k)} = \boldsymbol{U}\,(\boldsymbol{I} - \boldsymbol{\Phi}^{(k)})\boldsymbol{U}^T\boldsymbol{b} = \boldsymbol{U}\begin{pmatrix} (1-\phi_1^{(k)})\,\boldsymbol{u}_1^T\boldsymbol{b} \\ (1-\phi_2^{(k)})\,\boldsymbol{u}_2^T\boldsymbol{b} \\ \vdots \end{pmatrix}.$$

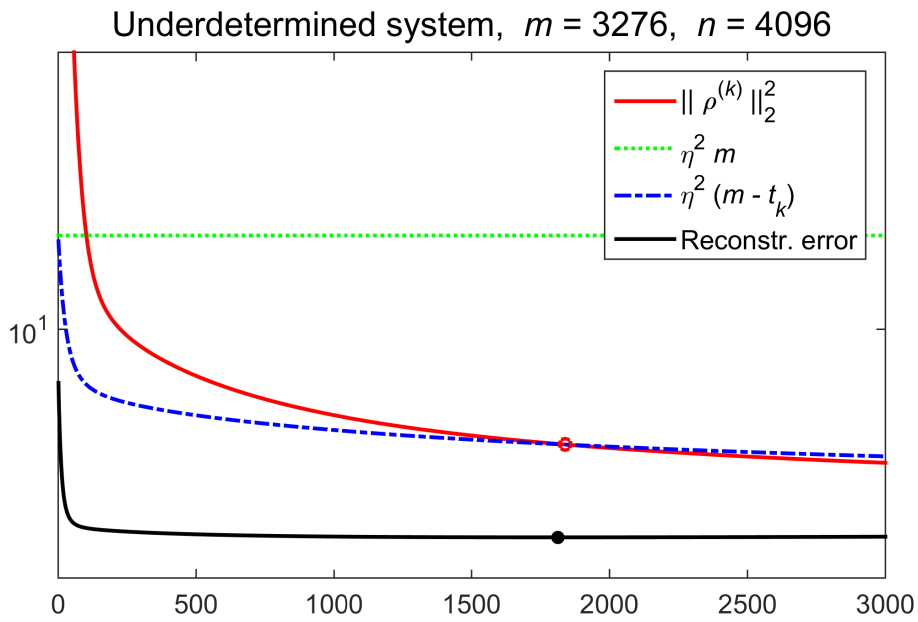
Hence, for Landweber's method,

$$\|\boldsymbol{\varrho}^{(k)}\|_2^2 = \sum_{i=1}^{m}\big(1-\phi_i^{(k)}\big)^2(\boldsymbol{u}_i^T\boldsymbol{b})^2 = \sum_{i=1}^{m}(1-\omega\,\sigma_i^2)^{2k}(\boldsymbol{u}_i^T\boldsymbol{b})^2.$$

Since ω is always chosen such that $|1-\omega\,\sigma_i^2| < 1$ the factors $(1-\omega\,\sigma_i^2)^{2k}$ – and hence the squared residual norm – decrease monotonically with k.

Illustration of Fit-to-Noise-Level Rule for Landweber





Parallel-beam example: image size $= 64 \times 64$; no. detector pixels $= 91$; projection angles $= 3°, 6°, 9°, \ldots, 180°$ and $8°, 16°, 24°, \ldots, 180°$.

Figures show reconstruction error $\|\bar{\boldsymbol{x}} - \boldsymbol{x}^{(k)}\|_2$ and residual norm $\|\boldsymbol{\varrho}^{(k)}\|_2$ versus k, together with threshold $\eta\sqrt{m}$ and the function $\eta\sqrt{m - t_k}$.

We stop near the optimal number of iterations. DP stops much too early.

Minimization of the Prediction Error

Instead of fitting to the noise level (as described above) we can find the number of iterations that minimizes the prediction error, i.e., the difference between the noise-free data $\bar{\boldsymbol{b}} = \boldsymbol{A}\,\bar{\boldsymbol{x}}$ and the predicted data $\boldsymbol{A}\,\boldsymbol{x}^{(k)}$:

$$\min_k \|\boldsymbol{A}\,\boldsymbol{x}^{(k)} - \bar{\boldsymbol{b}}\|_2.$$

Statisticians refer to various measures of this difference as the predictive risk, and the resulting method for choosing k is often called the *unbiased predictive risk estimation (UPRE)* method.

Again we present the results specifically in the framework of iterative reconstruction methods.

Derivation of the UPRE Rule

The expected squared norm of the prediction error (the risk) is

$$\mathcal{E}\big(\|\bar{\boldsymbol{b}} - \boldsymbol{A}\,\boldsymbol{x}^{(k)}\|_2^2\big) = \|(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}_k^{\#})\,\bar{\boldsymbol{b}}\|_2^2 + \eta^2\,\mathrm{trace}\big((\boldsymbol{A}\boldsymbol{A}_k^{\#})^2\big)$$

while the expected squared norm of the residual can be written as

$$\mathcal{E}\big(\|\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}^{(k)}\|_2^2\big) = \|(\boldsymbol{I} - \boldsymbol{A}\,\boldsymbol{A}_k^{\#})\,\bar{\boldsymbol{b}}\|_2^2 + \eta^2\,\mathrm{trace}\big((\boldsymbol{A}\boldsymbol{A}_k^{\#})^2\big) - 2\eta^2\,\mathrm{trace}(\boldsymbol{A}\boldsymbol{A}_k^{\#}) + \eta^2\,m.$$

Combining these two equations we can eliminate one of the trace terms and arrive at the following expression for the risk:

$$\mathcal{E}\big(\|\bar{\boldsymbol{b}} - \boldsymbol{A}\,\boldsymbol{x}^{(k)}\|_2^2\big) = \mathcal{E}\big(\|\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}^{(k)}\|_2^2\big) + 2\eta^2\,\mathrm{trace}(\boldsymbol{A}\boldsymbol{A}_k^{\#}) - \eta^2\,m.$$

Substituting the actual squared residual norm $\|\boldsymbol{\varrho}^{(k)}\|_2^2 = \|\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}^{(k)}\|_2^2$ for its expected value, we define the UPRE risk as a function of k:

$$U_k = \|\boldsymbol{\varrho}^{(k)}\|_2^2 + 2\,\eta^2\,t_k - \eta^2\,m.$$

Minimizer of $U_k \rightarrow$ approximation to a minimizer of the prediction error.

Stop Rule: UPRE

Note that U_k may not have a unique minimizer, and we therefore choose the smallest k at which U_k has a local minimum.

Stop Rule: UPRE

Find the smallest k that minimizes $U_k = \| \boldsymbol{\varrho}^{(k)} \|_2^2 + 2\,\eta^2\,t_k - \eta^2\,m$.

This rule also depends on an estimate of the standard deviation η of the noise – which may or may not be a problem in practise.

We shall therefore describe an alternative method for minimization of the prediction error that does not depend on knowledge of η.

Cross Validation

Assume that we remove the ith element b_i from the right-hand side (the noisy data), compute a reconstruction $\boldsymbol{x}_{[i]}^{(k)}$, and then use this vector to compute a prediction $\hat{b}_i = \boldsymbol{r}_i^T \boldsymbol{x}_{[i]}^{(k)}$ of the missing data b_i.

The goal is then to choose the k that minimizes the following measure of all the prediction errors:

$$\widehat{G}_k = \frac{1}{m} \sum_{i=1}^{m} \left(b_i - \hat{b}_i \right)^2 = \frac{1}{m} \sum_{i=1}^{m} \left(b_i - \boldsymbol{r}_i^T \boldsymbol{x}_{[i]}^{(k)} \right)^2 .$$

We can avoid the vectors $\boldsymbol{x}_{[i]}^{(k)}$ and write $\widehat{G}_k$ directly in terms of $\boldsymbol{x}^{(k)}$:

$$\widehat{G}_k = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{b_i - \boldsymbol{r}_i^T \boldsymbol{x}^{(k)}}{1 - \alpha_i^{(k)}} \right)^2 ,$$

where $\alpha_i^{(k)}$ = ith diagonal element of the influence matrix $\boldsymbol{A}\boldsymbol{A}_k^{\#}$ for $\boldsymbol{x}^{(k)}$.

Generalized Cross Validation (GCV)

The minimizer of $\widehat{G}_k$ depends on the particular ordering of the data. ☹

Generalized cross validation (GCV) circumvents this problem by replacing all $\alpha_i^{(k)}$ with their average

$$\mu^{(k)} = \frac{1}{m} \sum_{i=1}^{m} \alpha_i^{(k)} = \frac{1}{m} \operatorname{trace}(\boldsymbol{A}\boldsymbol{A}_k^{\#}) = \frac{t_k}{m},$$

leading to the modified measure

$$\widetilde{G}_k = \frac{1}{m} \frac{1}{(1-\mu^{(k)})^2} \sum_{i=1}^{m} \left(b_i - \boldsymbol{r}_i^T \boldsymbol{x}^{(k)}\right)^2 = \frac{\|\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}^{(k)}\|_2^2}{m(1-t_k/m)^2} = m \frac{\|\boldsymbol{\varrho}^{(k)}\|_2^2}{(m-t_k)^2}.$$

The minimizer of $\widetilde{G}_k$ is, of course, independent of the factor m and hence we choose to define the GCV risk as a function of k as

$$G_k = \|\boldsymbol{\varrho}^{(k)}\|_2^2 \,/\, (m-t_k)^2.$$

Stop Rue: GCV

We arrived at an η-free stopping rule:

Stop Rue: GCV

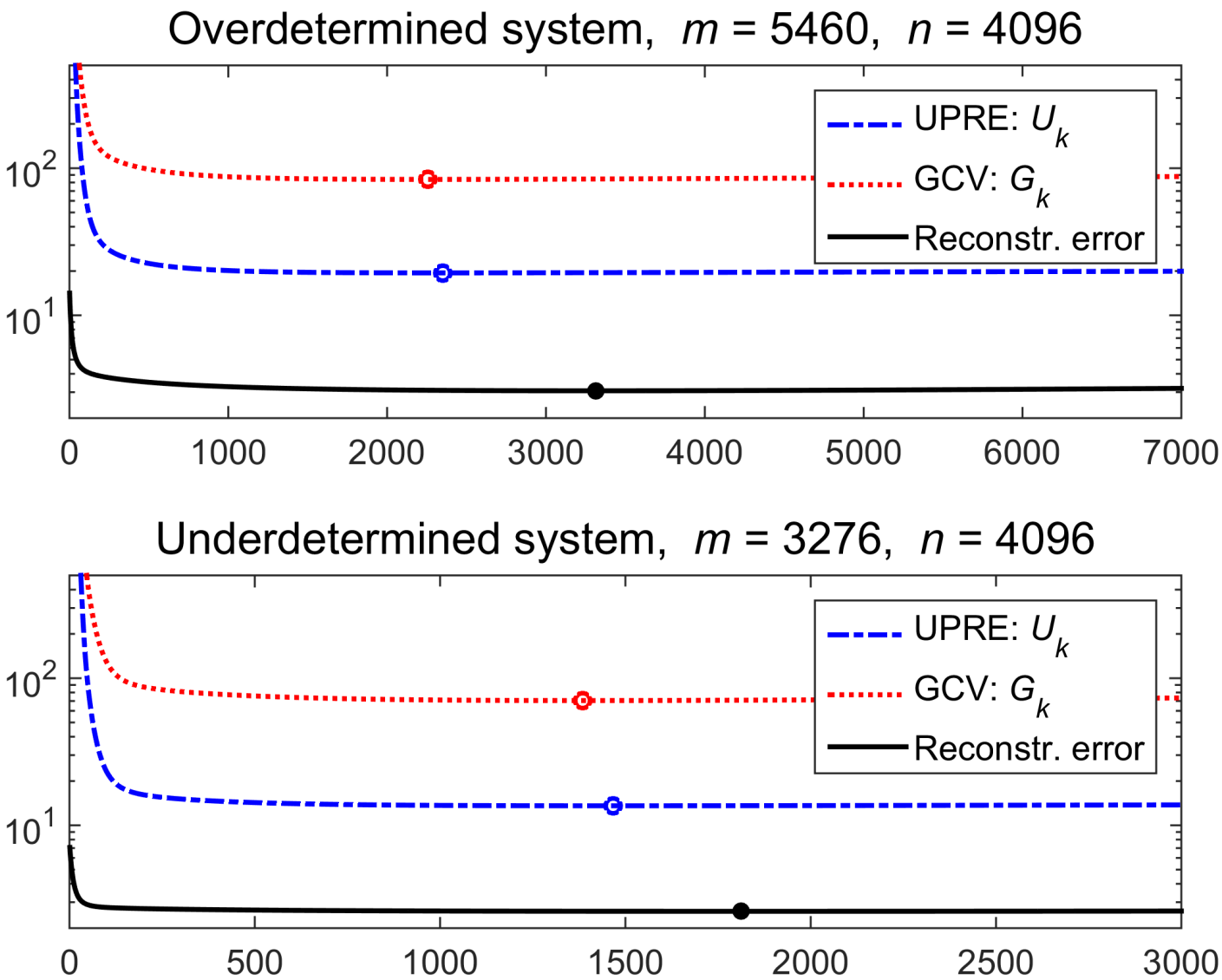
Find the k that minimizes $G_k = \|\boldsymbol{\varrho}^{(k)}\|_2^2 \,/\, (m - t_k)^2$.

The value of k which minimizes G_k is also an estimate of the value that minimizes the prediction error. Specifically, if k_{GCV} minimizes the GCV risk G_k and k_{PE} minimizes the prediction error $\|\bar{\boldsymbol{b}} - \boldsymbol{A}\,\boldsymbol{x}^{(k)}\|_2^2$, then

$$\mathcal{E}\big(\|\bar{\boldsymbol{b}} - \boldsymbol{A}\,\boldsymbol{x}^{(k_{\mathrm{GCV}})}\|_2^2\big) \to \mathcal{E}\big(\|\bar{\boldsymbol{b}} - \boldsymbol{A}\,\boldsymbol{x}^{(k_{\mathrm{PE}})}\|_2^2\big) \quad \text{for} \quad m \to \infty.$$

A practical note. UPRE and GCV need a few iterations too many detect a minimum of U_k and G_k. But the minimum of $\|\bar{\boldsymbol{x}} - \boldsymbol{x}^{(k)}\|_2$ is usually very flat, and it hardly makes any difference if terminate the algorithm a few iterations after the minimum of U_k or G_k.

Illustration of UPRE and GCV Rules for Landweber



Estimation of the Trace Term I

We must estimate the trace term t_k efficiently – without the SVD of $\boldsymbol{A}$ or the influence matrix $\boldsymbol{A}\,\boldsymbol{A}_k^{\#}$. The most common way to compute this estimate is via a Monte Carlo approach.

If $\overline{\boldsymbol{w}} \in \mathbb{R}^m$ is a random vector with $\overline{w}_i \sim \mathcal{N}(0,1)$, and $\boldsymbol{S} \in \mathbb{R}^{m \times m}$ is a symmetric matrix, then $\overline{\boldsymbol{w}}^T \boldsymbol{S}\,\overline{\boldsymbol{w}}$ is an unbiased estimate of trace$(\boldsymbol{S})$.

Hence $\overline{t}_k^{\,\text{est}} = \overline{\boldsymbol{w}}^T \boldsymbol{A}\,\boldsymbol{A}_k^{\#}\,\overline{\boldsymbol{w}}$ is an unbiased estimator of $t_k = \text{trace}(\boldsymbol{A}\,\boldsymbol{A}_k^{\#})$.

We need to compute the matrix-vector product $\boldsymbol{A}_k^{\#}\overline{\boldsymbol{w}}$ efficiently.

Apply the algebraic iterative method to the system $\boldsymbol{A}\,\overline{\boldsymbol{\xi}} = \overline{\boldsymbol{w}}$ which, after k iterations, produces the vector $\overline{\boldsymbol{\xi}}^{(k)} = \boldsymbol{A}_k^{\#}\overline{\boldsymbol{w}}$. The resulting estimate

$$\overline{t}_k^{\,\text{est}} = \overline{\boldsymbol{w}}^T \boldsymbol{A}\,\overline{\boldsymbol{\xi}}^{(k)},$$

is the standard Monte Carlo trace estimate.

Code for Trace Term Estimator I

Basic Landweber algorithm with MC trace estimator

$\overline{\boldsymbol{w}}$ = random m-vector for trace estimation
$\boldsymbol{x}^{(0)}$ = initial vector
$\overline{\boldsymbol{\xi}}^{(0)} = 0$ initial zero vector for trace estimation
$\boldsymbol{z} = \boldsymbol{A}^T \overline{\boldsymbol{w}}$
for $k = 0, 1, 2, \ldots$

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \omega\, \boldsymbol{A}^T (\boldsymbol{b} - \boldsymbol{A}\, \boldsymbol{x}^{(k)})$$

$$\overline{\boldsymbol{\xi}}^{(k+1)} = \overline{\boldsymbol{\xi}}^{(k)} + \omega\, \boldsymbol{A}^T (\overline{\boldsymbol{w}} - \boldsymbol{A}\, \overline{\boldsymbol{\xi}}^{(k)})$$

$\bar{t}_{k+1}^{\,\mathrm{est}} = \boldsymbol{z}^T \overline{\boldsymbol{\xi}}^{(k+1)}$ trace estimate

stopping rule goes here

end

Estimation of the Trace Term II

We can avoid the expensive multiplication with $\boldsymbol{A}$ for each k with an approach that applies to unprojected iterative methods of the general form

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \omega\, \boldsymbol{A}^T \boldsymbol{B}\, (\boldsymbol{b} - \boldsymbol{A}\, \boldsymbol{x}^{(k)}),$$

where $\boldsymbol{B}$ is a general $m \times m$ matrix (it is not required to be symmetric). This includes Landweber's, Cimmino's and Kaczmarz's methods.

When we apply such a method with arbitrary nonzero starting vector $\boldsymbol{\xi}^{(0)}$ to the system $\boldsymbol{A}\,\boldsymbol{\xi} = 0$ then the iterates are $\boldsymbol{\xi}^{(k)} = (\boldsymbol{I} - \omega \boldsymbol{A}^T \boldsymbol{B}\, \boldsymbol{A})^k\, \boldsymbol{\xi}^{(0)}$.

Specifically, if we use a random starting vector $\boldsymbol{\xi}^{(0)} = \boldsymbol{w} \in \mathbb{R}^n$ with elements $w_i \sim \mathcal{N}(0,1)$, and if $\boldsymbol{\xi}^{(k)}$ denote the corresponding iterates for $\boldsymbol{A}\,\boldsymbol{\xi} = 0$, then it can be shown that $\boldsymbol{w}^T \boldsymbol{\xi}^{(k)}$ is an unbiased estimator of $n - \mathrm{trace}(\boldsymbol{A}\, \boldsymbol{A}_k^{\#})$.

This leads to the alternative Monte Carlo trace estimate

$$t_k^{\mathrm{est}} = n - \boldsymbol{w}^T \boldsymbol{\xi}^{(k)}.$$

Code for Trace Term Estimator II

Basic Landweber algorithm with alternative MC trace estimator

$\boldsymbol{w}$ = random n-vector
$\boldsymbol{x}^{(0)}$ = initial vector
$\boldsymbol{\xi}^{(0)} = \boldsymbol{w}$ initial vector for for trace estimation
for $k = 0, 1, 2, \ldots$

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \omega\, \boldsymbol{A}^T (\boldsymbol{b} - \boldsymbol{A}\, \boldsymbol{x}^{(k)})$$

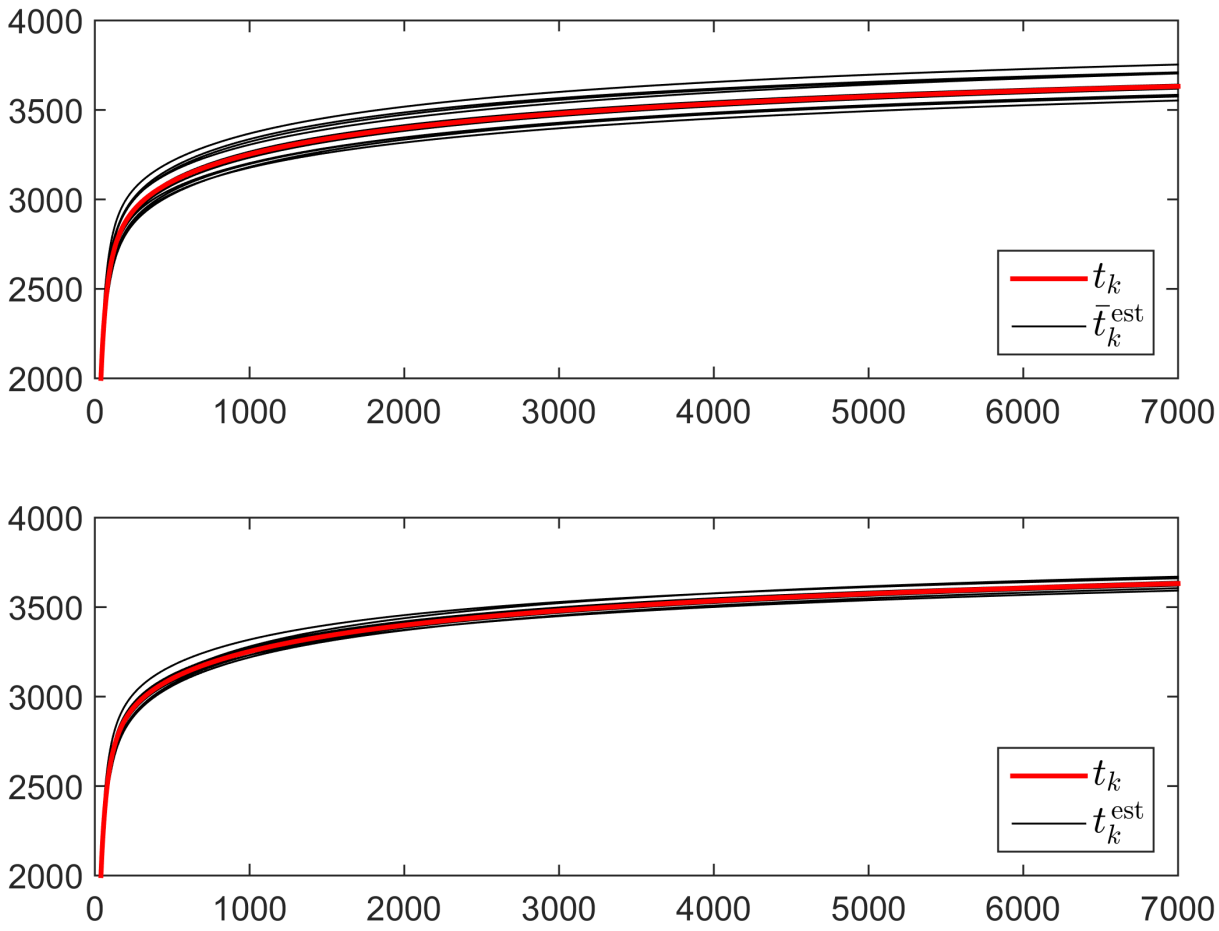
$$\boldsymbol{\xi}^{(k+1)} = \boldsymbol{\xi}^{(k)} + \omega\, \boldsymbol{A}^T (0 - \boldsymbol{A}\, \boldsymbol{\xi}^{(k)})$$

$t_{k+1}^{\text{est}} = n - \boldsymbol{w}^T \boldsymbol{\xi}^{(k+1)}$ trace estimate

stopping rule goes here

end

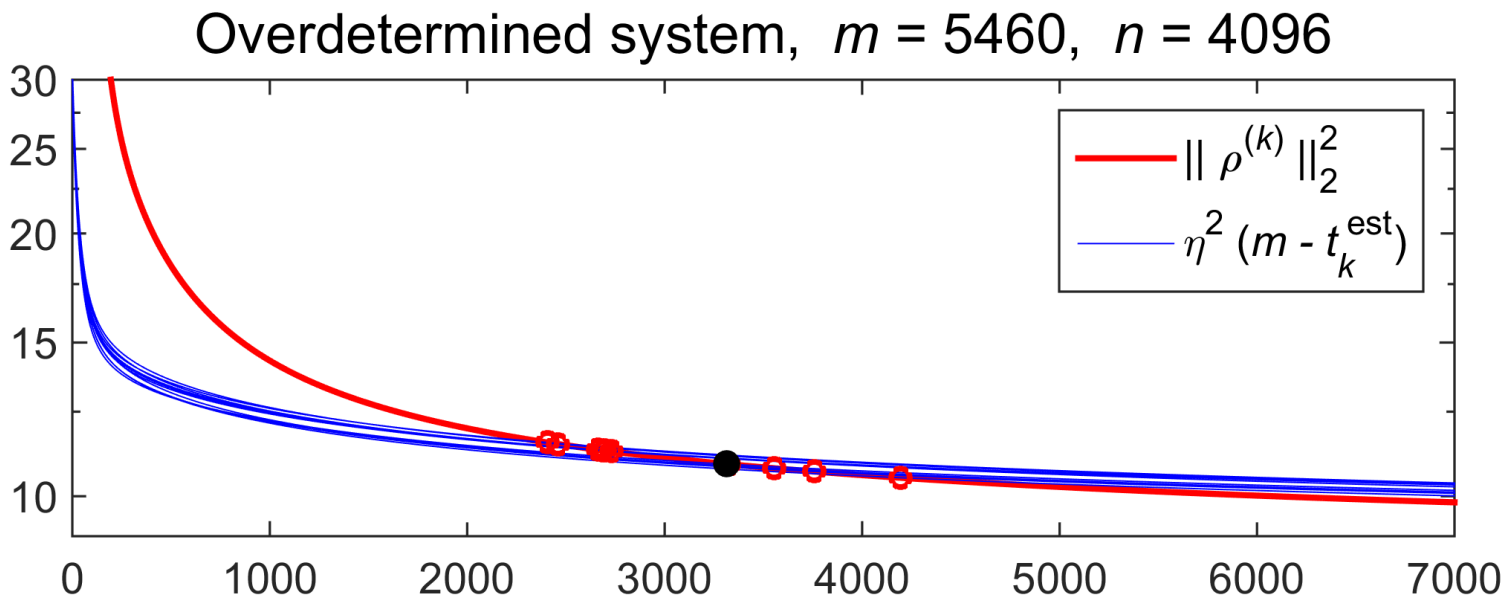
Comparison of the Two Trace Estimates for Landweber



Thick **red line**: exact trace t_k.

Thin black lines: trace estimates for 10 different random vectors $\overline{\boldsymbol{w}}$ and $\boldsymbol{w}$.

Applying t_k^{est} to the Fit-to-Noise-Level Stopping Rule



We used 10 different random vectors $\boldsymbol{w}$. The corresponding 10 intersections between $\|\boldsymbol{\varrho}^{(k)}\|_2^2$ (thick **red line**) and $\eta^2\,(m - t_k^{\text{est}})$ (thin blue lines) are shown by the red circles.

The black dot • shows the intersection with the exact $\eta^2\,(m - t_k)$.

Estimation of the Noise Level, In Our Framework

When we use the trace estimate t_k^{est} in the GCV stopping rule, then we seek a minimum for the approximate GCV risk given by

$$G_k^{\text{est}} = \|\boldsymbol{\varrho}^{(k)}\|_2^2 \,/\, (m - t_k^{\text{est}})^2.$$

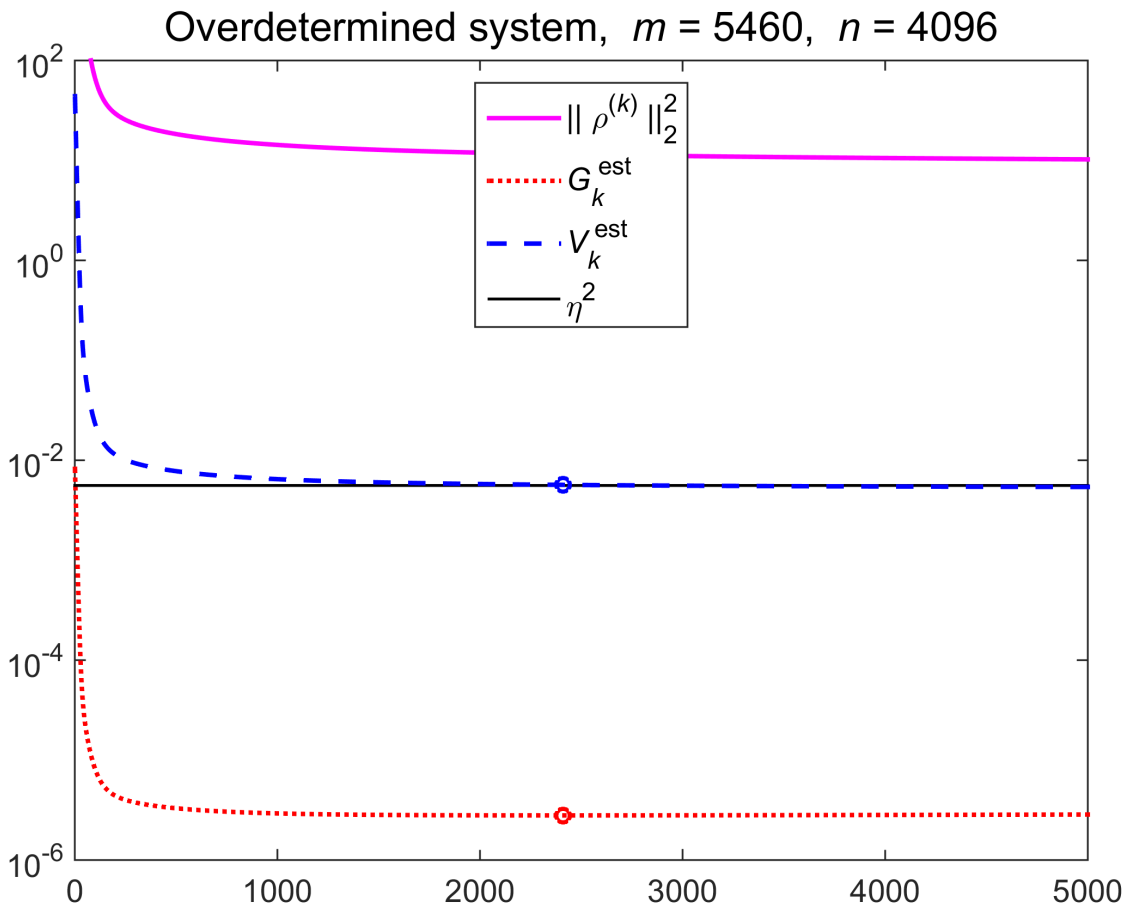
Now define the function

$$V_k^{\text{est}} = \|\boldsymbol{\varrho}^{(k)}\|_2^2 \,/\, (m - t_k^{\text{est}}) = G_k^{\text{est}}\,(m - t_k^{\text{est}}).$$

According to the fit-to-noise-level stopping rule it follows that when we stop the iterations, the ratio $\|\boldsymbol{\varrho}^{(k)}\|_2^2 \,/\, (m - t_k)$ is approximately equal to the noise variance η^2.

Hence, if we terminate at iteration $k = \hat{k}$ for which G_k^{est} is minimum, then the corresponding value $V_{\hat{k}}^{\text{est}}$ is an inexpensive estimate of η^2.

Illustration of G_k^{est} and $V_{\hat{k}}^{\text{est}}$



The approximate GCV risk G_k^{est} has a minimum at $\hat{k} = 2410$.

Circles represent $G_{\hat{k}}^{\text{est}}$ and $V_{\hat{k}}^{\text{est}}$, the latter being a good estimate of η^2.