Semi-Convergence and Stopping Rules

We will use the AIR TOOLS software that you already downloaded. Throughout we define the relative noise level as

\[ \rho = \frac{\|e\|_2}{\|\bar{b}\|_2} \quad \text{with} \quad \bar{b} = A \bar{x}. \]  

(1)

Given \( \rho \) and a noise-free right hand side \( \text{bex} \), we can generate such a noise vector \( e \) and the corresponding noisy data \( b \) by means of the following code:

\[
\begin{align*}
&>> e = \text{randn(size(bex))}; \\
&>> e = \rho*\text{norm(bex)}*e/\text{norm(e)}; \\
&>> b = \text{bex} + e;
\end{align*}
\]

Part 1 – Semi-Convergence of AIR Methods

Exercise 1. A Simple Illustration

This exercise illustrates how the noise in the data gives rise to semi-convergence of the iterates \( x^{(k)} \) of Kaczmarz’s method. We use a rather small system to keep the computing times reasonably small. We will use a parallel-beam test problem with the Shepp-Logan phantom, generated by means of:

\[
\begin{align*}
&>> N = 64; \\
&>> \text{theta} = 4:4:180; \\
&>> [A,\text{bex},\text{xex}] = \text{parallelTomo}(N,\text{theta});
\end{align*}
\]

Run \( k_{\text{max}} = 800 \) iterations of Kaczmarz’s method with these relative noise levels:

\[
\rho = 0, \; 0.0015, \; 0.0020, \; 0.0025 \quad (2)
\]

(you can try others, as well, depending on your patience).

Plot the error history, i.e., the relative error \( \| \bar{x} - x^{(k)} \|_2 / \| \bar{x} \|_2 \) as a function of \( k \), for all four noise levels. You should see the characteristic semi-convergence behavior for all \( \rho \neq 0 \). At what iteration number \( k \) do we reach the smallest error, for each \( \rho \)?

Note: you will get a different answer each time you run your code, because the results depend on the actual realization of the noise.

Exercise 2. Semi-Convergence is Associated with Inverse Problems

Semi-convergence is the fundamental mechanism that makes the algebraic iterative reconstruction methods suited for solving noisy tomography problems (and inverse problems in general). As we have seen, semi-convergence relies on the fact that the exact solution to an inverse problem is dominated by the initial SVD components –
and hence semi-convergence does not manifest itself for an arbitrary ill-conditioned system.

To illustrate this, we will perform a numerical experiment to see if we experience semi-convergence for a linear system \( A \mathbf{x} = \mathbf{b} \) with a synthetic random ill-conditioned matrix generated by:

\[
\begin{align*}
& N = 32; \\
& n = N^2; \\
& [U, S, V] = \text{svd}(\text{randn}(n)); \\
& S = \text{diag}(\text{logspace}(0, -12, n)); \\
& A = U \cdot S \cdot V';
\end{align*}
\]

The condition number of this matrix is \( \kappa = 10^{12} \). Perform an experiment similar to above, with a Shepp-Logan phantom of appropriate size:

\[
\begin{align*}
& \text{Xex} = \text{phantom}(N); \\
& \text{xex} = \text{Xex}(:); \\
& \text{bex} = A \cdot \text{xex};
\end{align*}
\]

Use the relative noise level \( \rho = 0.02 \) and perform \( k_{\text{max}} = 2000 \) iterations. Do you see any useful semi-convergence? Show the best reconstruction – is it good?

**Exercise 3. Semi-Convergence for Constrained Problems**

The theory says that we should also observe semi-convergence when we apply the algebraic iterative reconstruction methods to constrained problems. We will investigate this for Kaczmarz’s method with box constraints, which we impose by setting \( \text{options.box} = 1 \).

Repeat Exercise 1, still with \( k_{\text{max}} = 800 \) and now with box constraints. Discuss the error histories, and compare the results with those from Exercise 1.

**Exercise 4. Surprising Semi-Convergence?**

In this exercise we also use box constraints. We use a parallel-beam test problem with a special phantom with binary pixel values, generated by means of:

\[
\begin{align*}
& N = 64; \\
& \theta = 4:4:180; \\
& A = \text{parallelto}m(N, \theta); \\
& \text{Xex} = \text{phantomgallery}('binary', N); \\
& \text{xex} = \text{Xex}(:); \\
& \text{bex} = A \cdot \text{xex};
\end{align*}
\]

Use the relative noise level \( \rho = 0.002 \); you only need perform \( k_{\text{max}} = 100 \) Kaczmarz iterations. What is so special about the behavior of the error history for this problem? Can you explain this behavior?
Exercise 5. SVD Analysis of Cimmino’s Method

The goals of the exercise are twofold: to illustrate how to perform SVD analysis of an iterative method, and to demonstrate how to so for a method with \( M \neq I \). We consider Cimmino’s method with relaxation parameter \( \omega \):

\[
x^{(k+1)} = x^{(k)} + \omega A^T M^{-1} (b - Ax^{(k)})
\]

with \( M = \text{diag}(m\|r_i\|_2^2) \), and where we defined

\[
\hat{A} = M^{-1/2} A \quad \text{and} \quad \hat{b} = M^{-1/2} b.
\]

Note that we compute the square root of the diagonal matrix \( M^{-1} \) simply by computing the square roots of its diagonal elements. Then it follows that the SVD analysis of Cimmino’s method follows that of Landweber’s method, but with \( A \) and \( b \) replaced by \( \hat{A} \) and \( \hat{b} \). In particular we should use the SVD of \( \hat{A} \) for the analysis.

Theory question: write down the expression for the \( k \)th iterate of Cimmino’s method in terms of \( \hat{b} \) and the SVD of \( \hat{A} \). Ditto for the iteration error and the noise error.

Now generate a parallel-beam test problem, and the corresponding \( Z = M^{-1/2} \):

\[
\begin{align*}
\text{>> } & N = 64; \\
\text{>> } & \text{theta} = 3:3:180; \\
\text{>> } & [A,bex] = \text{paralleltomo}(N,\text{theta}); \\
\text{>> } & [A,bex] = \text{rzr}(A,bex,3); \quad m = \text{size}(A,1); \\
\text{>> } & d = \sqrt{m*\text{sum}(A.*2,2)}; \\
\text{>> } & Z = \text{spdiags}(1./d(:),0,\text{sparse}(m,m)); \% \text{Sparse diagonal matrix.}
\end{align*}
\]

The statement \([A,bex] = \text{rzr}(A,bex,3)\) removes all zero rows, as well as rows with less than 4 nonzero elements; this is to ensure a numerically stable computation of \( M^{-1/2} \). You must use the statement \([U,S,V] = \text{svd}(\text{full}(Z*A))\) to compute the SVD of the sparse matrix. Save the SVD for later use: \text{save SavedSVD U S V}. Add noise with relative noise level \( \rho = 0.02 \), and run Cimmino’s method on the noisy problem with relaxation parameter \( \omega = 1/\|\hat{A}\|_2^2 \) and with \( k_{\text{max}} = 5000 \) iterations. The relaxation parameter is specified via the \text{options} input. For each of the following iterations

\[ k = 10, 50, 100, 500, 1000, 5000 \]

use the SVD of \( \hat{A} \) to compute and display:

- The filter factors \( \phi_i^{(k)} \), \( i = 1, 2, \ldots, n \).
- The \( k \)th iterate \( x^{(k)} \). Compare your result with the output from \text{cimmino}.
- The iteration error of \( \bar{x} - x^{(k)} \), and its norm.
- The noise error of \( \hat{x}^{(k)} - x^{(k)} \), and its norm.

Your results should behave qualitatively like those for Landweber’s method. At which iteration \( k \) do the iteration error and the noise error have approximately the same norm?
Part 2 – Stopping Rules

To perform numerical experiments with the stopping rules you will use the following strategy. Choose a maximum number of iterations \(k_{\text{max}}\) that is (hopefully) large enough that you pass the minimum reconstruction error \(\|\hat{x} - x^{(k)}\|_2\), run the iterative method (say, Landweber) on your system, and compute the optimal number of iterations:

\[
\begin{align*}
\text{>> } X &= \text{landweber}(A,b,1:k_{\text{max}}); \\
\text{for } k = 1:k_{\text{max}}, \text{ err}(k) &= \text{norm}(x_{\text{ex}} - X(:,k)); \text{ end} \\
\text{>> } [\cdot,k_{\text{opt}}] &= \text{min}(\text{err});
\end{align*}
\]

Then compare the number of iterations found by the stopping rules with \(k_{\text{opt}}\).

When you are asked to use the trace term estimators, you need to run the iterative method again. For the first trace estimator, compute the estimates \(t_{\text{est}}^{(k)}\) as follows:

\[
\begin{align*}
\text{>> } \text{wbar} &= \text{randn}(m,1); \\
\text{>> } X\text{Ibar} &= \text{landweber}(A,\text{wbar},1:k_{\text{max}}); \% \text{Right-hand side is } \text{wbar}. \\
\text{>> } \text{for } k = 1:k_{\text{max}}, \text{ tbarkest}(k) &= \text{w}'A\text{XIbar}(:,k); \text{ end}
\end{align*}
\]

For the second trace estimator, compute the estimates \(t_{\text{est}}^{(k)}\) as follows:

\[
\begin{align*}
\text{>> } \text{w} &= \text{randn}(n,1); \\
\text{>> } X\text{I} &= \text{landweber}(A,\text{zeros}(m,1),1:k_{\text{max}},\text{w}); \% \text{Right-hand side is } 0 \text{ and } \text{starting vector is } \text{w}. \\
\text{>> } \text{for } k = 1:k_{\text{max}}, \text{ tkest}(k) &= \text{w}'X\text{I}( :, k); \text{ end}
\end{align*}
\]

Exercise 6. Deriving the Stopping Rules for Cimmino’s Method

(This is a purely theoretical exercise which you may skip if you prefer to do numerical experiments only.)

We derived the three stopping rules fit-to-noise-level, UPRE, and GCV specifically for Landweber’s method, and here we will extend them to Cimmino’s methods. Our main theoretical task is to formulate the stopping rules in terms of the matrix \(\hat{A}\) and its SVD

\[
\hat{A} = \hat{U} \hat{\Sigma} \hat{V}^T. \tag{7}
\]

According to Eqs. (3) and (5) we can write \(x^{(k)}\) in two different ways:

\[
x^{(k)} = A_k^# b \quad \text{and} \quad x^{(k)} = \hat{A}_k^# \hat{b} = \hat{A}_k^# M^{-1/2} b, \tag{8}
\]

where, similar to the Landweber algorithm,

\[
\hat{A}_k^# = \hat{V} \hat{\Phi}^{(k)} \hat{\Sigma}^{-1} \hat{U}^T, \quad \hat{\Phi}^{(k)} = \text{diag} \left( \hat{\phi}_i^{(k)} \right), \quad \hat{\Sigma} = \text{diag} \left( \hat{\sigma}_i \right) \tag{9}
\]

with

\[
\hat{\phi}_i^{(k)} = 1 - (1 - \omega \hat{\sigma}_i^2)^k, \quad i = 1, 2, \ldots, n. \tag{10}
\]
What we need is an expression for the trace of the influence matrix $AA_k^\#$, but expressed in terms of the singular values $\hat{\sigma}_i$ of the matrix $\hat{A}$. To do so, show that:

$$AA_k^\# = M^{1/2} \hat{\Phi}^{(k)} \hat{U}^T \hat{M}^{-1/2} = (M^{1/2} \hat{U}) \hat{\Phi}^{(k)} (M^{1/2} \hat{U})^{-1}.$$  

(11)

The multiplication from the left and from the right with $M^{1/2} \hat{V}$ and its inverse, respectively, is a similarity transform which leaves the eigenvalues unchanged. Using that the trace of a matrix is the sum of its eigenvalues, show that:

$$t_k = \text{trace}(AA_k^\#) = \sum_{i=1}^{n} \hat{\sigma}_i^{(k)}.$$  

(12)

This result allows us to compute the three functions $\eta \sqrt{m-t_k}$, $U_k$ and $G_k$ needed in the stopping rules for Cimmino’s method.

**Exercise 7. Using the Stopping Rules for Cimmino’s Method**

We shall now test the three stopping rules fit-to-noise-level, UPRE and GCV applied to Cimmino’s method and the test problem from Exercise 5. To use these stopping rules, you need to compute the trace term $t_k$ using the expression in (12). If you save the SVD of the matrix $\hat{A}$ in Exercise 5, you can load it again by means of `load SavedSVD`.

We suggest that you just run, say, $k_{\text{max}} = 2000$ iterations, and then analyse the three functions $\|\varrho^{(k)}\|_2 - \eta \sqrt{m-t_k}$ (find a sign change), $U_k$ and $G_k$ (find the minimum). Compare the number of iterations found by the three stopping rules with the optimal $k$, i.e., the one that minimizes the reconstruction error $\|\bar{x} - x^{(k)}\|_2$.

To get an idea of the robustness of these methods, try to repeat the experiments with different realizations of the noise.

**Exercise 8. And Now: The Trace Term Estimator**

As you might have guessed, in this exercise we replace the trace term $t_k$ – which requires the singular values – with the alternative Monte Carlo trace estimate $t_{\text{est}}^k$. Recall that you specify a nonzero starting vector as the fourth input argument to `cimmino`.

Repeat the experiments with $t_k$ replace by $t_{\text{est}}^k$ and compare the two approaches. Try different random starting vectors to get a feeling of the robustness of the stopping rules based on $t_{\text{est}}^k$.

**Exercise 9. Stopping Rules for Kaczmarz’s Method**

In today’s final exercise we consider the same test problem once again, but this time we solve it by means of Kaczmarz’s method; $k_{\text{max}} = 100$ iterations should suffice. For this method it is cumbersome to compute the exact trace term, so we only use the three stopping rules based on the estimated trace term $t_{\text{est}}^k$.

How good is the performance of the stopping rules? Are the associated reconstructions almost as good as the optimal one?