Journal of Computational and Applied Mathematics 236 (2012) 2167–2178

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

AIR Tools — A MATLAB package of algebraic iterative reconstruction methods☆

Per Christian Hansen*, Maria Saxild-Hansen

Department of Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Lyngby, Denmark

ARTICLE INFO

Article history:
Received 29 October 2010

MSC:
65F22
65Y15

Keywords:
ART methods
SIRT methods
Semi-convergence
Relaxation parameters
Stopping rules
Tomographic imaging

ABSTRACT

We present a MATLAB package with implementations of several algebraic iterative reconstruction methods for discretizations of inverse problems. These so-called row action methods rely on semi-convergence for achieving the necessary regularization of the problem. Two classes of methods are implemented: Algebraic Reconstruction Techniques (ART) and Simultaneous Iterative Reconstruction Techniques (SIRT). In addition we provide a few simplified test problems from medical and seismic tomography. For each iterative method, a number of strategies are available for choosing the relaxation parameter and the stopping rule. The relaxation parameter can be fixed, or chosen adaptively in each iteration; in the former case we provide a new "training" algorithm that finds the optimal parameter for a given test problem. The stopping rules provided are the discrepancy principle, the monotone error rule, and the NCP criterion; for the first two methods "training" can be used to find the optimal discrepancy parameter.

1. Introduction

Iterative regularization methods for computing stable regularized solutions to discretizations of inverse problems have been used for decades in medical imaging, geophysics, materials science, and many other disciplines that involve 2D and 3D imaging [1–3]. There are many variants of these iterative methods, and they have in common that they rely on matrix-vector multiplications and therefore are well suited for large-scale problems.

In the beginning of the 20th century the Polish mathematician Stefan Kaczmarz [4] and the Italian mathematician Gianfranco Cimmino [5] independently developed iterative algorithms for solving linear systems of equations. In 1970 Gordon et al. rediscovered the Kaczmarz method applied in medical imaging [6]; they called the method ART (Algebraic Reconstruction Technique) and when Hounsfield patented the first CT-scanner in 1972 the classical methods found their practical purpose in tomography [7]. The iterative methods are routinely used for tomographic imaging problems [8–10].

This paper presents a MATLAB package containing a number of algebraic iterative methods for reconstruction, i.e., for solving large linear systems of the form

$$Ax \simeq b, \quad A \in \mathbb{R}^{m\times n} \tag{1}$$

used in tomography and many other inverse problems. We assume that the elements of A are nonnegative, and that A contains no zero rows or columns; there are no restrictions on the dimensions. The methods are summarized in a common framework using the same notation, and all the MATLAB functions have similar interfaces. Also included in our MATLAB

☆ This work is part of the project CSI: Computational Science in Imaging, supported by grant no. 274-07-0065 from the Danish Research Council for Technology and Production Sciences.

* Corresponding author.
E-mail address: pch@imm.dtu.dk (P.C. Hansen).

0377-0427/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2011.09.039

functions are several strategies for choosing the relaxation parameter as well as several stopping rules. A few tomography test problems are also included. The package requires MATLAB version 7.8 or later, and no additional toolboxes are needed; together with the manual it available from www.imm.dtu.dk/~pch/AIRtools.

We have included the most common algebraic iterative reconstruction methods in the package — but we left out block versions of the methods, which are better suited for other programming languages than MATLAB. The main part of this package was originally developed in [11]. Our main contribution is the design of new training algorithms for the optimal relaxation parameter, and the "packaging" of all the methods with identical calling sequences and functionality plus strategies for the various parameters and suitable stopping rules. We are not aware of other MATLAB packages with this functionality. The C++ package SNARK09 [12] includes some of the same methods as this package, as well as functions to generate more realistic medical test problems.

Our paper is organized as follows. Sections 2 and 3 survey the iterative methods included in the package, and Section 4 summarizes our software considerations. Section 5 introduces the concept of semi-convergence which sets the stage for the parameter-choice strategies presented in Section 6 (including a new training method for the optimal fixed parameter). In Section 7 we survey the stopping rules used in the package, Section 8 introduces nonnegativity constraints, and in Section 9 we present the package's three test problems. Finally, in Section 10 we give an overview of the package, and in 11 we present a few numerical examples.

Throughout the paper, all vectors are column vectors, a_j is the jth column of A, a^i is the transposed of the ith row of A, $\langle x, y\rangle = x^T y$ is the standard inner product, $\rho(\cdot)$ is the spectral radius (the largest positive eigenvalue), I is an identity matrix of appropriate dimensions, and $\mathcal{R}(M)$ is the range or column space of the matrix M. For each of the linear equations in (1) we define the affine hyperplane $\mathcal{H}_i = \{x \in \mathbb{R}^n \mid \langle a^i, x\rangle = b_i\}$, and the orthogonal projection of a vector z on $\mathcal{H}_i$ is given by $\mathcal{P}_i(z) = z + (b_i - \langle a^i, z\rangle)\|a^i\|_2^{-2} a^i$.

2. Algebraic Reconstruction Techniques (ART)

These row-action methods treat the equations one at a time during the iterations — hence the ART methods are said to be fully sequential. The typical step in these methods involves the ith row of A in the following update of the iteration vector:

$$x \leftarrow x + \lambda_k \frac{b_i - \langle a^i, x\rangle}{\|a^i\|_2^2} a^i, \tag{2}$$

where λ_k is a relaxation parameter. What distinguishes the methods is the order in which the rows are processed. The following convergence theorem is from [13].

Theorem 1. *Assume that $0 < \lambda_k < 2$. If the system (1) is consistent then the iteration (2) converges to a solution x^*, and if $x^0 \in \mathcal{R}(A^T)$ then x^* is the solution of minimum 2-norm. If the system is inconsistent then every subsequence associated with a^i converges, but not necessarily to a least squares solution.*

We note that if $\lambda_k \to 0$ for $k \to \infty$ then the iteration converges to a weighted least squares solution; this feature is only included for the symmetric Kaczmarz method.

2.1. The Kaczmarz method

This is undoubtedly the most well-known method of the ART class [8,14]; this method uses a fixed $\lambda_k = \lambda \in (0, 2)$, and $\lambda = 1$ was used in the original paper [4] in which case the next iterate is clearly the orthogonal projection of the old iterate on $\mathcal{H}_i$. In the literature the method is often referred to as ART, which can be confusing since ART is also the name of Algebraic Reconstruction Techniques in general. The kth iteration consists of a "sweep" through the m rows of A from top to bottom, i.e.,

$$i = 1, 2, \ldots, m.$$

2.2. Symmetric Kaczmarz

The symmetric Kaczmarz method [15] is a variant in which one "sweep" of the Kaczmarz method is followed by another "sweep" using the rows in reverse order, and one iteration therefore consists of $2m - 2$ steps. The kth iteration of the symmetric Kaczmarz method thus consists of the following "double sweep":

$$i = 1, 2, \ldots, m-1, m, m-1, \ldots, 3, 2.$$

This method can, in principle, be formulated as an SIRT method (these methods are covered in the next section) but this formulation is impractical for computations, and the method must be implemented by means of sequential row operations. We allow both a fixed $\lambda \in (0, 2)$ and an iteration-dependent λ_k.

2.3. Randomized Kaczmarz

As a way to accelerate the convergence of the Kaczmarz method for some problems, it has been proposed [16] to select the rows a^i of A randomly with probability proportional to $\|a^i\|_2^2$. For the randomized Kaczmarz method we cannot talk about iterations; but in order to compare all the methods in the package we define one "iteration" of this method to consist of m random steps.

The original randomized Kaczmarz method from [16] does not include a relaxation parameter, but in our implementation we introduced a constant $\lambda \in (0, 2)$; the default value is $\lambda = 1$ which was used in the original paper.

3. Simultaneous Iterative Reconstruction Techniques (SIRT)

These methods are "simultaneous" in the sense that all the equations are used at the same time in one iteration. The methods can be written in the general form:

$$x^{k+1} = x^k + \lambda_k T A^T M(b - Ax^k), \quad k = 0, 1, 2, \ldots, \tag{3}$$

where x^k denotes the current iteration vector, x^{k+1} is the new iteration vector, λ_k is a relaxation parameter, and the matrices M and T are symmetric positive definite. Different methods depend on the choice of these matrices. The following theorem regarding convergence summarizes the results from [17–21].

Theorem 2. *The iterates of the form (3) converge to a solution x^* of $\min_x \|Ax - b\|_M$ if and only if*

$$0 < \epsilon \le \lambda_k \le 2/\rho(TA^T MA) - \epsilon,$$

where ϵ is an arbitrarily small but fixed constant. If in addition $x^0 \in \mathcal{R}(TA^T)$ then x^* is the unique solution of minimum T^{-1}-norm (minimum 2-norm if $T = I$).

We incorporate positive weights $w_i > 0$, $i = 1, \ldots, m$ in three of the SIRT methods in this package (Cimmino, CAV, and DROP, see below), and if weights are not specified then all weights are set to 1.

3.1. Landweber's method

The classical Landweber method [22] takes the form:

$$x^{k+1} = x^k + \lambda_k A^T(b - Ax^k), \quad k = 0, 1, 2, \ldots, \tag{4}$$

which corresponds to setting $M = I$ and $T = I$ in (3).

3.2. Cimmino's method

The method was introduced in [5] where it was based on reflections on hyperplanes (see also [14]). It is often presented in a variant based on projections, which is also the version used in this package; the only difference is a factor 2 in the length of the step, which is absorbed in the relaxation parameter. The next iterate x^{k+1} is the average of the projections of the previous iterate x^k on all the hyperplanes $\mathcal{H}_i$ for $i = 1, \ldots, m$:

$$x^{k+1} = \frac{1}{m}\sum_{i=1}^m \mathcal{P}_i(x^k) = \frac{1}{m}\sum_{i=1}^m \left(x^k + \frac{b_i - \langle a^i, x^k\rangle}{\|a^i\|_2^2} a^i\right) = x^k + \frac{1}{m}\sum_{i=1}^m \frac{b_i - \langle a^i, x^k\rangle}{\|a^i\|_2^2} a^i.$$

The version of Cimmino's method included in this package is obtained by including a relaxation parameter λ_k as well as weights w_i:

$$x^{k+1} = x^k + \lambda_k \frac{1}{m}\sum_{i=1}^m w_i \frac{b_i - \langle a^i, x^k\rangle}{\|a^i\|_2^2} a^i, \quad k = 0, 1, 2, \ldots, \tag{5}$$

and using matrix notation Cimmino's method takes the form of (3) with $M = D$ and $T = I$, where we have defined

$$D = \frac{1}{m}\mathrm{diag}\left(\frac{w_i}{\|a^i\|_2^2}\right). \tag{6}$$

3.3. Component averaging (CAV)

Cimmino's original method uses equal weighting of the contributions from the projections, which seems fair when A is a dense matrix. Component Averaging (CAV) was introduced in [23] as an extension of Cimmino's method which incorporates information about the sparsity of A (if any), in a heuristic way. Let s_j denote the number of nonzero elements of

column j:

$$s_j = \text{NNZ}(a_j), \quad j = 1, \ldots, n, \tag{7}$$

and define the diagonal matrix $S = \text{diag}(s_1, \ldots, s_n)$ and the norm $\|a^i\|_S^2 = \langle a^i, Sa^i \rangle = \sum_{j=1}^n a_{ij}^2 s_j$ for $i = 1, \ldots, m$. Then the CAV algorithm takes the form:

$$x^{k+1} = x^k + \lambda_k \sum_{i=1}^m w_i \frac{b_i - \langle a^i, x^k \rangle}{\|a^i\|_S^2} a^i, \quad k = 0, 1, 2, \ldots, \tag{8}$$

and when A is dense then $S = mI$ and we get Cimmino's method. The CAV algorithm thus takes the matrix form (3) with $M = D_S$ and $T = I$, where we have defined

$$D_S = \text{diag}\left(\frac{w_i}{\|a^i\|_S^2} \right). \tag{9}$$

3.4. Diagonally Relaxed Orthogonal Projections (DROP)

This method is another extension of Cimmino's original method, in which the factors s_j from (7) are incorporated in a different manner, namely, by computing the next iterate as

$$x^{k+1} = x^k + \lambda_k S^{-1} \sum_{i=1}^m w_i \frac{b_i - \langle a^i, x^k \rangle}{\|a^i\|_2^2} a^i. \tag{10}$$

The DROP method thus has the form (3) with $T = S^{-1}$ and $M = mD$, with D from (6). Again we obtain Cimmino's method when A is dense in which case $S^{-1} = m^{-1}I$.

It is shown in [18] that $\rho(S^{-1}A^TMA) \le \max_i\{w_i\}$, which means that convergence is guaranteed if $\lambda_k \le (2-\epsilon)/\max_i\{w_i\}$ where ϵ is an arbitrarily small but fixed constant. In Section 11 we demonstrate experimentally that it is worthwhile to use the larger upper bound $2/\rho(S^{-1}A^TMA)$ for λ_k, instead of the easily computed bound $2/\max\{w_i\}$.

3.5. Simultaneous Algebraic Reconstruction Technique (SART)

This method was originally developed in the ART setting [24], but it can also be written and implemented in the SIRT form (3) and we therefore categorize it as an SIRT method. It is written in the following matrix form:

$$x^{k+1} = x^k + \lambda_k D_{\text{r}}^{-1} A^T D_{\text{c}}^{-1} (b - Ax^k), \tag{11}$$

where the diagonal matrices D_{r} and D_{c} are defined in terms of the row and the column sums:

$$D_{\text{r}} = \text{diag}(\|a^i\|_1), \qquad D_{\text{c}} = \text{diag}(\|a_j\|_1).$$

We do not include weights in this method. The convergence for SART was independently established in [17,20], where it was shown that $\rho(D_{\text{r}}^{-1}A^TD_{\text{c}}^{-1}A) = 1$ and that convergence therefore is guaranteed for $0 < \lambda_k < 2$.

4. Considerations towards the package

To establish the computational work in the different methods in this package we introduce the concept of a work unit WU, defined as the work involved in one matrix-vector multiplication, and WU $= 2mn$ flops if A is a dense matrix. All methods use 2 WU per iteration except symmetric Kaczmarz, which uses 4 WU since it applies twice as many steps per iteration. If a stopping rule is used in the ART methods then one additional WU is needed to compute the residual vector.

The user should notice that in this package, the iterations of the SIRT methods are much faster than those of the ART methods, because MATLAB loops are slow — using other programming languages there would not be this difference in the execution times.

When doing operations with the diagonal matrices M and T we have chosen the fastest implementation; in case of memory exhaustion most of the SIRT methods also have an alternative implementation which requires less memory but with a larger running time. If alternative code exists it can be found in the comments in the code.

All methods can be restarted, continuing the iterations from the last iteration of a previous call. For the ART methods this is achieved simply by calling the ART function again using the previous last iterate as starting vector. For the SIRT methods, this requires that M and T (when needed), as well as the estimate of the spectral radius, are returned from the SIRT function and passed as input in the next call, along with the previous last iterate.

In case of a fixed relaxation parameter λ, all methods check if λ is in the interval for which convergence is guaranteed, and a warning is given if this is not the case. For the SIRT methods this requires estimation of the spectral radius needed in the upper bound, which is done by means of MATLAB's `svds` and `eigs` functions — the convergence is fast because the largest

eigenvalue or singular value is well separated from the rest for matrices with nonnegative entries, and we feel that this slight overhead is acceptable for making the software user-friendly. Note that if the same matrix A is involved in repeated calls to the same iterative method, then one avoids re-computation of the spectral radius by specifying three output variables in the first call and supplying restart parameters for the subsequent calls, e.g.,

```
options.lambda = lambda;                          % Fixed lambda
[X,info,restart] = landweber(A,b1,k,[],options); % First call
options.restart = restart;
[X,info] = landweber(A,b2,k,[],options);          % Second call
```

The same technique should be used when using a relaxation-parameter rule that involves the use of the spectral radius; see Section 6.

If no fixed λ is specified and no method is specified for choosing λ_k then we use an ad-hoc default fixed λ which was found by numerical experiments; see Table 1 in Section 10.

5. SVD analysis and semi-convergence

The SIRT iterates x^k from (3) with $T = I$, unit weights $w_i = 1$, and a fixed relaxation parameter $\lambda_k = \lambda$ can be expressed as filtered SVD solutions [25,26]. If we let the SVD for the matrix $M^{1/2}A$ take the form

$$M^{1/2}A = U\Sigma V^T = \sum_{i=1}^{n} u_i \sigma_i v_i^T,$$

then the kth iterate can be written as

$$x^k = V\Phi^{[k]}\Sigma^\dagger U^T b, \quad \Phi^{[k]} = \mathrm{diag}(\varphi_1^{[k]}, \ldots, \varphi_n^{[k]}), \tag{12}$$

where $\Sigma^\dagger$ is the pseudoinverse of Σ, and the diagonal elements of $\Phi^{[k]}$ are so-called filter factors given by

$$\varphi_i^{[k]} = 1 - (1 - \lambda\sigma_i^2)^k, \quad i = 1, \ldots, n. \tag{13}$$

The filter factors for small singular values satisfy

$$\varphi_i^{[k]} \approx k\lambda\sigma_i^2 \quad \text{for } \lambda\sigma_i^2 \ll 1. \tag{14}$$

This shows that the filter factors decay fast enough that we can achieve a regularized solution [3].

We now consider the propagation of noise from the right-hand side b to the solution x^k, and we therefore write b as

$$b = b^* + e, \quad b^* = Ax^*, \ e = \text{noise}, \tag{15}$$

where x^* is the solution defined in Theorem 2 with a noise-free right-hand side. Given the expression in (12) it follows that the error in x^k is given by

$$x^* - x^k = V(I - \Phi^{[k]})\Sigma^\dagger U^T b^* - V\Phi^{[k]}U^T e.$$

Hence we can write the ith component of the error, in the SVD basis, as

$$v_i^T(x^* - x^k) = (1 - \varphi_i^{[k]})\langle v_i, x^*\rangle - \varphi_i^{[k]}\frac{\langle u_i, e\rangle}{\sigma_i}. \tag{16}$$

The first component is the regularization error while the second component is the noise error.

We can assume that the discrete Picard condition is satisfied, i.e., that $|\langle u_i, b^*\rangle|$ decay faster than the singular values σ_i. In addition we assume that the noise is white, i.e., that all $\langle u_i, e\rangle$ have the same probability. From the behavior of the filter factors $\varphi_i^{[k]}$ it then follows that the regularization error decreases with k while the noise error increases with k. For more details, see the analysis in [25–27].

The typical situation is therefore that the error norm $\|x^* - x^k\|_2$ decreases in the initial iterations where the regularization error dominates, while the error norm starts to increase after a certain stage when the noise error starts to dominate. This particular behavior of iterative methods applied to discretizations of inverse problems is called semi-convergence [10]. Fig. 1 illustrates this behavior.

For the SIRT methods and the symmetric Kaczmarz method, the expression (14) shows that the parameters k and λ play the same role in the filter factors (and thus in the suppression of the noise). In particular, if $\varphi_i^{[k]}$ and $\bar{\varphi}_i^{[k]}$ denote the filter factors for two different choices λ and $\bar{\lambda}$, then $\varphi_i^{[k]} \approx \bar{\varphi}_i^{[\bar{k}]}$ if $\bar{k} \approx k\lambda/\bar{\lambda}$. The error histories for Cimmino's method in Fig. 1 confirm that the number of iterations to reach the minimum error is inversely proportional to λ.

For consistent problems we observe experimentally the same behavior for the classical and randomized Kaczmarz methods, as illustrated in Fig. 1. However, the situation is different in the case of inconsistent problems: Here the approximate solution in the sense of semi-convergence, i.e., the iterate $x^{[k]}$ with smallest error, is known to depend on the choice of λ.

For efficiency of the methods, it is important to choose the relaxation parameter λ_k in such a way that we achieve the smallest error in the smallest number of iterations. Moreover, we need a reliable stopping criterion that can stop the iterations at this point. Such methods are discussed in the next two sections.

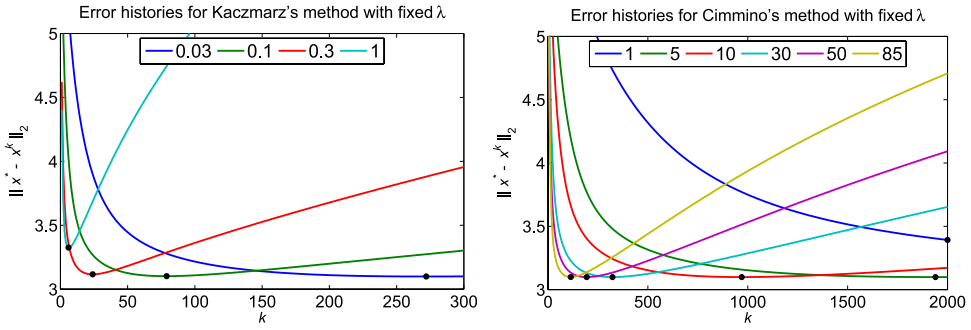

Fig. 1. Error histories for the Kaczmarz and Cimmino methods with different values of a fixed relaxation parameter λ, clearly showing the semi-convergence of these methods. The black dots indicate the minimum errors, and we see that these minimum errors are independent of λ (except for extreme values).

6. Methods for choosing the relaxation parameter

We have implemented three strategies for choosing the relaxation parameter in accordance with the semi-convergence discussed above. These methods are based on the observation that for both the ART and the SIRT methods with a fixed relaxation parameter $\lambda_k = \lambda$, the error reaches a smallest value which is practically independent of λ. This is illustrated in Fig. 1, where the black dots indicate the minimum errors.

6.1. Training for an optimal fixed parameter

The optimal fixed relaxation parameter λ is the one that achieves the fastest semi-convergence to the minimum error, and finding this λ requires knowledge of the exact solution. In order to estimate the optimal λ we developed and implemented a *training* method which finds the λ for which we achieve the fastest semi-convergence for a noisy test problem with known exact solution. If this test problem resembles the given problem then, hopefully, the estimated λ from the test problem is almost optimal for the given problem.

Our training algorithm has two parts: first we must determine the overall minimum error for all feasible λ, and then we must determine the λ for which we reach this minimum error with the smallest number of iterations. If η_λ denotes the minimum error for at given λ, then we note that η_λ changes slightly with λ due to the discrete nature of the iterates — but the deviation of η_λ for different λ is small; cf. Fig. 1. From experiments we found that $\lambda_{\text{SIRT}} = 1/\rho(A^T MA)$ is a safe choice of relaxation parameter to determine the overall minimum error. We define the upper bound for the overall minimum error as $\hat{\eta} = 1.01\eta_{\lambda_{\text{SIRT}}}$.

Now let k_λ denote the number of iterations needed to reach the error level $\hat{\eta}$ for a given λ. Our goal in the second part of the training algorithm is to compute the optimal λ that minimizes k_λ. Extensive tests indicate that k_λ is a unimodal function of λ and we can therefore use the following modified version of the golden section search.

This search method uses an initial search interval (α, β) for λ which is the convergence interval for the given SIRT method. The method then computes two interior points $\alpha' = \alpha + r(\beta - \alpha)$ and $\beta' = \alpha + (1-r)(\beta - \alpha)$, where $r = (3-\sqrt{5})/2$, and the associated iteration numbers $k_{\alpha'}$ and $k_{\beta'}$. We also compute the minimum errors $\eta_{\alpha'}$ and $\eta_{\beta'}$ for these points. We then reduce the interval according to the following procedure, in the given order:

$\eta_{\alpha'} > \hat{\eta}$: the minimum error for $\lambda = \alpha'$ has not reached the overall minimum value, and we reduce the interval to (α', β).
$\eta_{\beta'} > \hat{\eta}$: the minimum error for $\lambda = \beta'$ has not reached the overall minimum value, and we reduce the interval to (α, β').
$k_{\alpha'} \geq k_{\beta'}$: both α' and β' are feasible values of λ and, according to the unimodality, we reduce the interval to (α', β).
$k_{\alpha'} < k_{\beta'}$: both α' and β' are feasible values of λ, and we reduce the interval to (α, β').

The interval reduction continues until the interval width is sufficiently small, and the optimal value of λ is then chosen as the interval's midpoint.

The training algorithm that we implemented for the ART methods is very similar; except that our experiments show that we should use $\lambda_{\text{ART}} = 0.25$ to determine the overall minimum error. Note, however, that for inconsistent problems the iteration number k and the relaxation parameter λ do not play interchangeable roles in the classical and randomized Kaczmarz methods, and the user may want to manually experiment with other values of λ than the one found by our strategy.

In the implementation of the training strategy a default maximum number of iterations $k_{\max}$ is used. For some problems we may not reach the overall minimum error within this number of iterations, and it is therefore possible for the user to increase the maximum number of iterations via an input parameter. This input parameter can also be decreased, if the user will only allow a small value of iterations (for example, to limit computing time). A possible consequence could be that, within the allowed number of iterations, the error does not reach its overall minimum and we are then faced with a different problem, namely, to determine the relaxation parameter that gives the smallest error for $k = k_{\max}$.

6.2. Line search

Instead of using training to determine a fixed λ – which is time consuming and requires a good test problem with realistic noise – we can try to compute λ_k in each iteration, in such a way that we reach the minimum error almost as fast as possible. The line search strategy [28] tries to do this by minimizing the error $\|x^* - x^k\|_2$ in each iteration, which is possible for SIRT methods with $T = I$ and consistent problems, i.e., $Ax^* = b$. Similar techniques are described in [29,30].

We can write the Landweber, Cimmino and CAV methods as $x^{k+1} = x^k + \lambda_k p^k$, where $p^k = A^T M(b - Ax^k)$. Line search minimizes the error norm $\|x^{k+1} - x^*\|_2$ for the next iterate, which leads to the choice $\lambda_k = \langle p^k, x^* - x^k\rangle / \|p^k\|_2^2$. If we use that $Ax^* = b$ and define $r^k = b - Ax^k$ then the numerator becomes $\langle M(b - Ax^k), b - Ax^k\rangle = \langle Mr^k, r^k\rangle$ while the denominator becomes $\|p^k\|_2^2 = \|A^T M(b - Ax^k)\|_2^2 = \|A^T Mr^k\|_2^2$. This gives us the following expression:

$$\lambda_k = \frac{\langle Mr^k, r^k\rangle}{\|A^T Mr^k\|_2^2}, \quad r^k = b - Ax^k. \tag{17}$$

For the DROP and SART methods, where $T \neq I$, the alternative expression

$$\lambda_k = \frac{\langle Mr^k, r^k\rangle}{\|A^T Mr^k\|_T^2}, \quad r^k = b - Ax^k \tag{18}$$

was derived in [31]; this choice minimizes the error norm $\|x^{k+1} - x^*\|_{T^{-1}}$ in each step.

6.3. Relaxation to control noise propagation = Diminishing step size

As an alternative to the line search strategy, which assumes a consistent problem, two other strategies were recently introduced in [25]. Both methods arise from the analysis of the semi-convergence behavior and the goal is to control and limit the noise component of the error. They are derived for SIRT methods with $T = I$, but in our package they can also be used for all SIRT methods (although the theory may not be not valid for $T \neq I$). Skipping the derivation of the strategies (which can be found in [25,31]), the relaxation parameters are given by $\lambda_0 = \lambda_1 = \sqrt{2}/\rho$ and

$$\lambda_k^{(1)} = \nu^{(1)} \frac{2}{\rho}(1 - \zeta_k), \qquad \lambda_k^{(2)} = \nu^{(2)} \frac{2}{\rho} \frac{1 - \zeta_k}{(1 - \zeta_k^k)^2}, \quad k = 2, 3, \ldots, \tag{19}$$

where $\rho = \rho(A^T MA)$ and ζ_k is the unique root in $(0, 1)$ of the polynomial

$$g_{k-1}(y) = (2k - 1)y^{k-1} - (y^{k-2} + \cdots + y + 1).$$

The semi-convergence is achieved by diminishing the step size, thus "slowing down" the iterations.

If we choose the constants $\nu^{(1)} = \nu^{(2)} = 1$ then we refer to (19) as the Ψ_1 and Ψ_2 strategies. Modified strategies are obtained by using $\nu^{(1)} = 2$ and $\nu^{(2)} = 1.5$ which are chosen heuristically to accelerate the convergence (see [25]).

7. Stopping rules

We implemented three strategies for determining the optimal number of iterations k_{opt}. The first two strategies require knowledge of the noise level $\delta = \|e\|_2$ as well as a user-chosen parameter τ, and we give training strategies to choose a reasonable value of τ. Throughout this section we define $r_M^k = M^{1/2}(b - Ax^k)$.

7.1. Discrepancy principle and monotone error rule

The well-known discrepancy principle (DP) amounts to choosing k_{opt} as the smallest k satisfying:

$$\begin{cases} \|r_M^k\|_2 \leq \tau\delta\|M^{1/2}\|_2, & \text{SIRT methods with } T = I, \\ \|r^k\|_2 \leq \tau\delta, & \text{all other methods.} \end{cases} \tag{20}$$

The monotone error rule (ME) [32] chooses the stopping index k_{opt} as the smallest k for which

$$\frac{\langle r_M^k, r_M^k + r_M^{k+1}\rangle}{\|r_M^k\|_2} \leq \tau\delta\|M^{1/2}\|_2. \tag{21}$$

While the theory underlying this rule only holds for $T = I$, we found experimentally that it also works for DROP and SART and therefore it is also included for these methods. A comparison of the rules (20) and (21) can be found in [33].

7.2. Training the DP and ME parameter

To generate reliable versions of the DP and ME stopping rules we use training [33] to determine a reasonable value of the parameter τ. For the SIRT methods with $T = I$ we define the ratio

$$R_k = \begin{cases} \|r_M^k\|_2/(\delta\|M^{1/2}\|_2) & \text{for DP} \\ \langle r_M^k, r_M^k + r_M^{k+1}\rangle/(\delta\|M^{1/2}\|_2\|r_M^k\|_2) & \text{for ME} \end{cases}$$

and for the remaining methods we use $R_k = \|r^k\|_2/\delta$. Given a test problem with known $\delta = \|e\|_2$ we compute the iteration number k_δ that minimizes the error $\|x^* - x^k\|_2$ and set

$$\tau = (R_{k_\delta} + R_{k_\delta - 1})/2.$$

A more robust approach is to repeat this for several noise realizations, and use the average of the found τ values; this is the version implemented in AIR Tools.

7.3. Normalized cumulative periodogram (NCP)

In the NCP approach [34,35] (see also [3]) we consider the residual vector $r^k = b - Ax^k$ as a time series, and the exact right-hand side as a smooth signal which appears clearly different from the noise vector e in (15). We then need to find the iteration number k for which the residual changes from being signal-like (dominated by components from the exact right-hand side) to being noise-like (dominated by components from the noise e).

Let $\hat{r}^k \in \mathbb{C}^m$ denote the discrete Fourier transform of the residual vector r^k, and let q denote the largest integer such that $q \le m/2$. Then we define the normalized cumulative periodogram (NCP) for the residual vector r^k as the vector $c^k \in \mathbb{R}^q$ with elements

$$c_i^k = \|\hat{r}^k(2\colon i+1)\|_2^2/\|\hat{r}^k(2\colon q+1)\|_2^2, \quad i = 1, \ldots, q.$$

If the residual vector consists of white noise, then by definition the expected power spectrum is flat, i.e., $\mathcal{E}(|(\hat{r}^k)_i|^2)$ is independent of i, and the points $(i, \mathcal{E}(c_i^k))$ on the expected NCP lie on the straight line from (0, 0) to $(q, 1)$. Actual noise does not have an ideal flat spectrum, but we can still expect the NCP to be close to a straight line. We thus choose the iteration number k for which the residual r^k represents white noise the most, in the sense that its NCP is closest to a straight line.

8. Nonnegativity constraints and projected methods

In many imaging problems it is known a priori that the reconstruction should be nonnegative, and it is therefore convenient to add such a constraint to the reconstruction algorithms, see Chapter 9 in [36]. We included this option in all the methods implemented in AIR Tools.

Let $\mathcal{P}$ denote the projection onto the positive orthant, i.e.,

$$[\mathcal{P}(x)]_i = \begin{cases} x_i, & x_i \ge 0 \\ 0, & \text{else.} \end{cases}$$

Then the projections are incorporate in each step of the iterations, i.e.,

$$x \leftarrow \mathcal{P}\left(x + \lambda_k \frac{b_i - \langle a^i, x\rangle}{\|a^i\|_2^2} a^i\right) \quad \text{for the ART methods,}$$

and

$$x^{k+1} = \mathcal{P}(x^k + \lambda_k T A^T M(b - Ax^k)) \quad \text{for the SIRT methods.}$$

The extension of the above parameter-choice rules to this case was studied in [31].

9. Test problems

The package includes three simplified 2D tomography test problems: parallel- and fan-beam medical X-ray tomography and seismic travel-time tomography. The former arise from transmission tomography [10] where one studies an object with non-diffractive radiation. The loss of intensity of the X-rays is recorded by a detector and used to produce an image of the irradiated object. Let I_o denote the source intensity, let I denote the intensity of the ray after having passed through the object, and let $f(t)$ denote the linear attenuation coefficient at $t \in \mathbb{R}^2$. Then the line integral of $f(t)$ along the ray satisfies

$$\int_{\text{ray}} f(x)d\ell = \log(I_0/I).$$

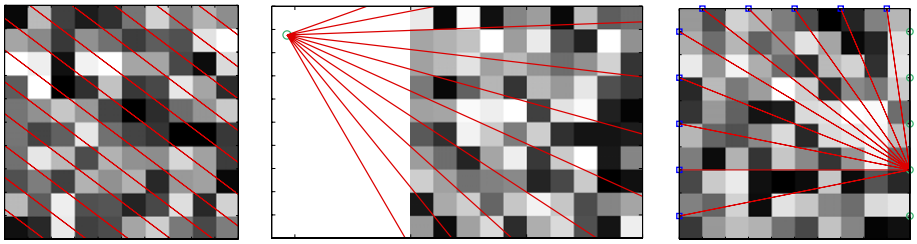

Fig. 2. Examples of the parallel- and fan-beam tomography problems (left and middle) and the seismic tomography problem (right) for $N = 10$. Only a few of the rays are shown.

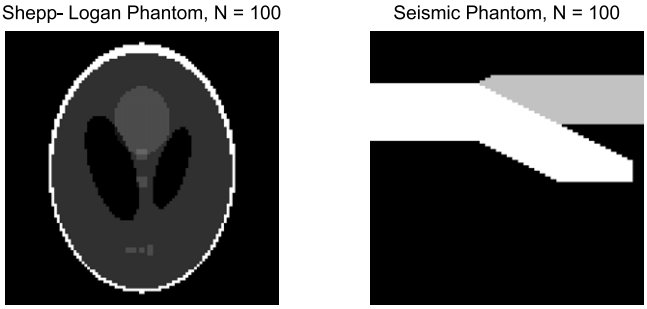

Fig. 3. The exact solutions for the parallel/fan-beam and seismic tomography test problems.

For parallel-beam tomography the rays arise from sources arranged in parallel and with equal spacing, and the sources are rotated around the domain using different angles in such a way that the rays are still parallel. For fan-beam tomography we only have a single source, from which a number of rays are arranged like a fan (we use an equiangular distribution of the rays), and again the source is rotated keeping the distance to the center of the domain constant. See Fig. 2 for an illustration of the geometries.

In seismic tomography the travel time of a seismic wave through a domain of the subsurface is measured. The travel time τ of a seismic wave along a ray is given by the line integral

$$\tau = \int_{\text{ray}} f(t)d\ell,$$

where $f(t)$ is the slowness at $t \in \mathbb{R}^2$, i.e., the reciprocal of the wave velocity. We consider a 2D subsurface slice; on the right border s equispaced sources are located, and on the left border and on the surface a total of p equispaced seismographs (or receivers) are located. For each of the s sources, p rays are transmitted such that all receivers are "hit".

To discretize the tomography problems, we divide the square domain into a grid of $N \times N$ pixels numbered from 1 to $n = N^2$ (starting with the cell in the upper left corner and running along the columns). Each cell j is assigned a constant value x_j of the attenuation or slowness, such that the vector x is a discretized version of the sought function. Moreover, we define the matrix element a_{ij} as the length of the ith ray through cell j, and $a_{ij} = 0$ if ray i does not pass through cell j. The measurement b_i of the attenuation or travel time for ray i is then:

$$b_i = \sum_{j=1}^{n} a_{ij}x_j, \quad i = 1, \ldots, m,$$

where m is the number of rays. Thus we obtain a linear system $Ax = b$ with a sparse $m \times n$ coefficient matrix determined solely by the geometry of the problem.

For the parallel- and fan-beam test problems the exact solution is the modified Shepp–Logan phantom head from [37]. For the seismic test problem we created a new phantom which illustrates a 2D subsurface of simple convergent boundaries of two tectonic plates with different slowness (we use a case where the plates create a subduction zone where one plate moves underneath the other). Fig. 3 shows the two exact solutions for $N = 100$.

We emphasize that our model problems are very simplistic — realistic forward modeling of tomographic data requires advanced software such as SNARK09 [12] for medical tomography and FAST [38] for travel-time seismic tomography.

10. Overview of the package

The AIR Tools package was developed for MATLAB version 7.8. Table 1 summarizes the reconstruction methods and their parameters and options, and below we list all the functions and scripts included in the package.

Table 1
The reconstruction methods in AIR Tools and their parameters and options.

	`kaczmarz`	`symkaczmarz`	`randkaczmarz`	`landweber`	`cimmino`	`cav`	`drop`	`sart`
λ_k upper bound	2	2	2	$2/\rho$	$2/\rho$	$2/\rho$	$2/\rho$	2
Default λ	0.25	0.25	1	$1/\rho$	$1/\rho$	$1/\rho$	$1/\rho$	1
Weights	−	−	−	−	+	+	+	−
Line search	−	−	−	+	+	+	+	+
Ψ_1 and Ψ_2	−	+	−	+	+	+	+	+
Modified Ψ_1 and Ψ_2	−	−	−	+	+	+	+	+
Discrep. principle	+	+	+	+	+	+	+	+
Monotone error rule	−	−	−	+	+	+	+	+
NCP	+	+	+	+	+	+	+	+

Iterative ART methods	
`kaczmarz`	Kaczmarz's method
`randkaczmarz`	The randomized Kaczmarz method
`symkaczmarz`	The symmetric Kaczmarz method
Iterative SIRT methods	
`cav`	Component Averaging (CAV) method
`cimmino`	Cimmino's method
`drop`	Diagonally Relaxed Orthogonal Projections (DROP) method
`landweber`	Landweber's method
`sart`	Simultaneous Algebraic Reconstruction Technique (SART)
Training routines	
`trainDPME`	Training strategy to find the best parameter τ when discrepancy principle or monotone error rule is used as stopping rule
`trainLambdaART`	Training strategy to find the best constant relaxation parameter λ for a given ART method
`trainLambdaSIRT`	Training strategy to find the best constant relaxation parameter λ for a given SIRT method
Test problems	
`fanbeamtomo`	Creates a 2D fan-beam tomography problem
`paralleltomo`	Creates a 2D parallel-beam tomography problem
`seismictomo`	Creates a 2D seismic tomography problem
Demo scripts	
`ARTdemo`	Illustrates the simple use of the ART methods
`nonnegdemo`	Illustrates the use of nonnegativity constraints
`SIRTdemo`	Illustrates the simple use of the SIRT methods
`trainingdemo`	Illustrates the use of the training routines as pre-processors for the SIRT and the ART methods
Auxiliary routines	
`calczeta`	Calculates the roots of the polynomial $g_k(y)$ of degree k
`rzr`	Removes zero rows from A and corresponding elements of b

11. Numerical examples

Our first example shows how to use AIR Tools to set up a small fan-beam tomography test problem, and compute reconstructions by means of Cimmino's method and three strategies for computing the relaxation parameter (training for a fixed λ, line search, and control of noise propagation):

```
N = 24;      % Problem size is N-by-N.
rnl = 0.05;  % Relative noise level.
kmax = 20;   % Number of of iterations.
```

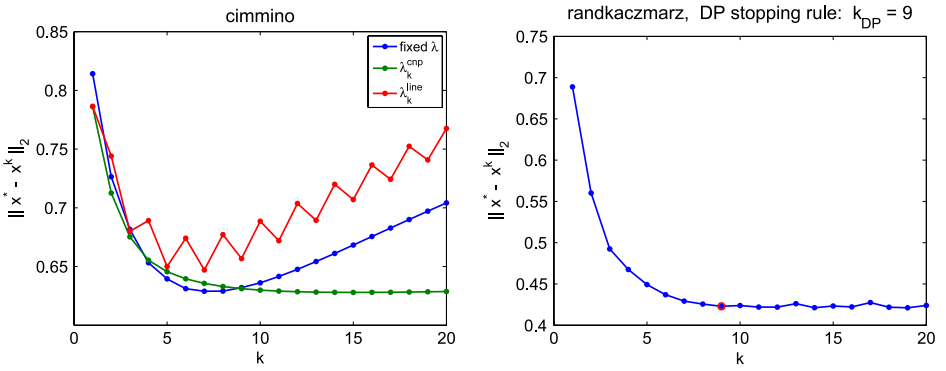

Fig. 4. Error histories $\|x^* - x^k\|_2$ for a small fan-beam tomography problem. Left: results for Cimmino's method using three different strategies for computing the relaxation parameter. Right: results for the randomized Kaczmarz method, using the discrepancy principle stopping with parameter τ found by training.

```
[A,bex,xex] = fanbeamtomo(N,10:10:180,32);  % Test problem
nx = norm(xex); e = randn(size(bex));       %  with noise.
e = rnl*norm(bex)*e/norm(e); b = bex + e;
lambda = trainLambdaSIRT(A,b,xex,@cimmino); % Train lambda.
options.lambda = lambda;                    % Iterate with
  X1 = cimmino(A,b,1:kmax,[],options);      %  fixed lambda.
options.lambda = 'psi2';                    % Iterate with
  X2 = cimmino(A,b,1:kmax,[],options);      %  cnp strategy.
options.lambda = 'line';                    % Iterate with
  X3 = cimmino(A,b,1:kmax,[],options);      %  line search.
```

The error histories for this example are shown in Fig. 4, and we see that all three strategies give fast semi-convergence to a minimum error about 0.63. The fixed λ is found by training as described in Section 6.1; this strategy can be cumbersome and only works when a realistic training problem is available. The line search strategy from Section 6.2 gives an undesired 'zig-zag' behavior of the error which is often observed for noisy problems. Finally, the noise-controlling strategy from Section 6.3 clearly controls the noise propagation, such that the error stays near the minimum value.

Our second example shows how to use the randomized Kaczmarz method with the discrepancy principle from Section 7.1 to solve the same test problem, using the training strategy from Section 7.2 to determine a good value of the parameter τ. The code is given below, and the corresponding error history is shown in Fig. 4.

```
% Find tau parameter for Discrepancy Principle by training.
delta = norm(e);
options.lambda = 1.5;
tau = trainDPME(A,bex,xex,@randkaczmarz,'DP',delta,5,options);
% Use randomized Kaczmarz with DP stopping criterion.
options.stoprule.type = 'DP';
options.stoprule.taudelta = tau*delta;
[x,info] = randkaczmarz(A,b,kmax,[],options);
k = info(2);              % Number of iterations used.
```

We conclude with an example which investigates the choice of the upper λ-limit for the DROP method of Section 3.4. Fig. 5 shows the number of iterations k_λ needed to reach the minimum error (see Section 6.1), for 50 values of λ between 0 and $2/\rho(S^{-1}A^TMA)$. All weights are 1, and the dotted vertical line shows the maximum λ-value $2/\max_i w_i = 2$ suggested in [18]. Clearly, the optimal λ that gives the smallest k_λ is larger than 2. This shows that we should use the full λ-interval, even though the upper bound requires more computational effort than $2/\max_i w_i$. More test and comparisons among the methods can be found in [11].

Acknowledgments

We are grateful to Jakob Heide Jørgensen for providing efficient MATLAB code to compute the sparse matrix in the test problems, and to Klaus Mosegaard for suggesting the seismic travel-time tomography test problem. We also thank Tommy Elfving for encouragement and advice during the development of the package, Jim Nagy for pointing out the need for nonnegativity constraints, and the referee for constructive comments and suggestions.

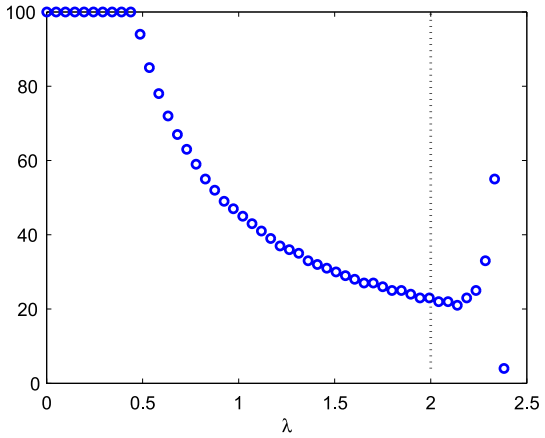

Fig. 5. The optimal number of iterations k_λ as a function of λ for the DROP method applied to the seismic tomography test problem. The optimal λ is greater than $2/\max_i w_i = 2$.

References

[1] M. Bertero, P. Boccacci, Introduction to Inverse Problems in Imaging, IOP Publishing Ltd., London, 1998.
[2] H.W. Engl, M. Hanke, A. Neubauer, Regularization of Inverse Problems, Kluwer, Dordrecht, The Netherlands, 1996.
[3] P.C. Hansen, Discrete Inverse Problems: Insight and Algorithms, SIAM, Philadelphia, 2010.
[4] S. Kaczmarz, Angenäherte Auflösung von Systemen linearer Gleichungen, Bull. Acad. Pol. Sci. Lett. A35 (1937) 355–357.
[5] G. Cimmino, Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari, La Ric. Sci., XVI, Ser. II, Anno IX 1 (1938) 326–333.
[6] R. Gordon, R. Bender, G.T. Herman, Algebraic reconstruction techniques for 3 dimensional electron microscopy and X-ray photograph, J. Theoret. Biol. 29 (1970) 471–481.
[7] G.N. Hounsfield, Computericed transverse axial scanning tomography: part I, discription of the system, Br. J. Radiol 46 (1973) 1016–1022.
[8] G.T. Herman, Fundamentals of Computerized Tomography: Image Reconstruction from Projections, 2nd ed., Springer, New York, 2009.
[9] A.C. Kak, M. Slaney, Principles of Computerized Tomographic Imaging, SIAM, Philadelphia, 2001.
[10] F. Natterer, The Mathematics of Computerized Tomography, SIAM, Philadelphia, 2001.
[11] M. Saxild-Hansen, AIR Tools — A MATLAB Package for Algebraic Iterative Reconstruction Techniques, M.Sc. Thesis, DTU Informatics, Technical University of Denmark, 2010.
[12] R. Davidi, G.T. Herman, J. Klukowska, SNARK09: A Programming System for the Reconstruction of 2D Images from 1D Projections, The CUNY Institute for Software Design and Development, http://www.snark09.com/SNARK09.pdf.
[13] P.P.B. Eggermont, G.T. Herman, A. Lent, Iterative algorithms for large partitioned linear systems, with applications to image reconstruction, Linear Algebra Appl. 40 (1981) 37–67.
[14] C.D. Meyer, Matrix Analysis and Applied Linear Algebra, SIAM, Philadelphia, 2000.
[15] Å. Björck, T. Elfving, Accelerated projection methods for computiong pseudoinverse solutions of systems of linear equations, BIT 19 (1979) 145–163.
[16] T. Strohmer, R. Vershynin, A randomized Kaczmarz algorithm for linear systems with exponential convergence, J. Fourier Anal. Appl. 15 (2009) 262–278.
[17] Y. Censor, T. Elfving, Block-iterative algorithms with diagonally scaled oblique projections for the linear feasibility problem, SIAM J. Matrix Anal. Appl. 24 (2002) 40–58.
[18] Y. Censor, T. Elfving, G.T. Herman, T. Nikazad, On diagonally relaxed orthogonal projection methods, SIAM J. Sci. Comput. 30 (2007) 473–504.
[19] M. Jiang, G. Wang, Convergence studies on iterative algorithms for image reconstruction, IEEE Trans. Med. Imaging 22 (2003) 569–579.
[20] M. Jiang, G. Wang, Convergence of the Simultaneous Algebraic Reconstruction Technique (SART), IEEE Trans. Image Process. 12 (2003) 957–961.
[21] G. Qu, C. Wang, M. Jiang, Necessary and sufficient convergence conditions for algebraic image reconstruction algorithms, IEEE Trans. Image Process. 18 (2009) 435–440.
[22] L. Landweber, An iteration formula for Fredholm integral of the first kind, Amer. J. Math. 73 (1951) 615–624.
[23] Y. Censor, D. Gordan, R. Gordan, Component averaging: an efficient iterative parallel algorithm for large sparse unstructured problems, Parallel Comput. 27 (2001) 777–808.
[24] A.H. Andersen, A.C. Kak, Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm, Ultrason. Imaging 6 (1984) 81–94.
[25] T. Elfving, T. Nikazad, P.C. Hansen, Semi-convergence and relaxation parameters for a class of SIRT algorithms, Electron. Trans. Numer. Anal. 37 (2010) 321–336.
[26] A. van der Sluis, H.A. van der Vorst, SIRT- and CG-type methods for the iterative solution of sparse linear least-squares problems, Linear Algebra Appl. 130 (1990) 257–303.
[27] T. Elfving, T. Nikazad, C. Popa, A class of iterative methods: semi-convergence, stopping rules, inconsistency, and constraining, in: Y. Censor, M. Jiang, G. Wang (Eds.), Biomedical Mathematics: Promising Directions in Imaging, Therapy Planning, and Inverse Problems, Medical Physics Publishing, Madison, WI, 2010.
[28] L.T. Dos Santos, A parallel subgradient projections method for the convex feasibility problem, J. Comput. Appl. Math. 18 (1987) 307–320.
[29] G. Appleby, D.C. Smolarski, A linear acceleration row action method for projecting onto subspaces, Electron. Trans. Numer. Anal. 20 (2005) 253–275.
[30] A. Dax, Line search acceleration of iterative methods, Linear Algebra Appl. 130 (1990) 43–63.
[31] T. Elfving, P.C. Hansen, T. Nikazad, Semi-convergence and relaxation parameters for projected SIRT algorithms, SIAM J. Sci. Comput. (submitted for publication).
[32] U. Hämarik, U. Tautenhahn, On the monotone error rule for aprameter choice in iterative and continuous regularization methods, BIT 41 (2001) 1029–1038.
[33] T. Elfving, T. Nikazad, Stopping rules for Landweber-type iteration, Inverse Problems 23 (2007) 1417–1432.
[34] P.C. Hansen, M.E. Kilmer, R.H. Kjeldsen, Exploiting residual information in the parameter choice for discrete ill-posed problems, BIT 46 (2006) 41–59.
[35] B.W. Rust, D.P. O'Leary, Residual periodograms for choosing regularization parameters for ill-posed problems, Inverse Problems 24 (2008) doi:10.1088/0266-5611/24/3/034005.
[36] C. Vogel, Computational Methods for Inverse Problems, SIAM, Philadelphia, 2002.
[37] P. Toft, The Radon Transform, Theory and Implementation, Ph.D. Thesis, DTU Informatics, Technical University of Denmark, 1996, pp. 199–201.
[38] C. Zelt, FAST: 3-D First Arrival Seismic Tomography programs, Dept. Earth Science, Rice University, Houston, TX, www.geophysics.rice.edu/department/faculty/zelt/fast.html.