Published in final edited form as:

IEEE Trans Radiat Plasma Med Sci. 2017 November ; 1(6): 511–516. doi:10.1109/TRPMS.

Fourier-Domain Analysis of the Iterative Landweber Algorithm

Gengsheng L. Zeng [Fellow, IEEE] and Ya Li

Abstract

This paper proposes a novel method of using the frequency-domain transfer function to investigate the property of an iterative algorithm for minimizing a quadratic objective function. This paper focuses on a two-dimensional (2D) tomography problem, which can be X-ray computed tomography (CT), positron emission tomography (PET), and single photon emission computed tomography (SPECT). Two questions regarding to the linear iterative Landweber algorithm are considered. The first question is whether stopping early is equivalent to getting a minimum-norm solution. The second question is whether the low frequency components always converge first. Our answers to these two questions are No.

Index Terms

Iterative algorithm; image reconstruction; tomography; numeric analysis; Fourier transform; linear filter

I. Introduction

In medical imaging such as X-ray computed tomography (CT), positron emission tomography (PET), and single photon emission computed tomography (SPECT), iterative image reconstruction algorithms are more and more popular in research and clinical applications [1–4]. An iterative algorithm is used to minimize an objective function, in which an imaging matrix A is used to describe the imaging physics and geometry. In tomography, this matrix A , representing the Radon transform, has some special properties [4]. One important property is that the matrix $A^T A$ acts as a convolution operator with a Fourier-domain transfer function of $1/\|\omega\|$, where ω is the frequency vector in the Fourier domain. It is this important property that makes it possible to represent the solution of an iterative algorithm in the Fourier domain [5–8]. The solution is fully characterized by the Fourier-domain transfer function.

In practice, the iterative algorithms stop after a finite number of iterations. If a constant is used as the initial image, a common experience is that the image starts out being very smooth without any contrast and sharp edges; as the iteration number increases, the contrast improves and the edges appear sharper. If the iteration number is too large, the resultant image becomes too noisy to be useful.

Once an iterative algorithm is stopped, many questions can be asked [9, 10], such as: What is this solution? Is it essentially the minimum-norm solution? Does the iteration number associate with frequency components? Do lower frequency components converge sooner than the higher frequency components? Some people believe that L^2 regularization (i.e.,

minimum-norm constraint) and early stopping can be equivalent [11]. According to our experiences, iterative reconstruction gives blurred images for low iteration numbers; images get sharper when iteration numbers are higher [4].

The iterative Landweber algorithm in this paper is restricted to two-dimensional (2D) image reconstruction from Radon transform of an object. Section II of this paper provides the mathematical tools which are essential in solving the proposed problems. The main procedure is as follows. The solution at the k th iteration of an iterative linear Landweber algorithm is expressed in a closed-form. Then this closed-form solution is expressed in the Fourier domain. The solution is the filtered-backprojection (FBP) solution followed by a post-filter. The transfer function of the post filter is associated with the iteration number k . An iterative FBP algorithm is also considered in Section II. Iterative FBP algorithms apply a filter to the data before backprojection [12–14]. In Section III, the frequency-domain transfer function is used to investigate the behavior of the iterative algorithm and to answer the raised questions. In the literature, there is little discussion on Fourier domain study of an iterative algorithm [15]. To our knowledge, our proposed Fourier-domain method to investigate the properties of an iterative algorithm is novel. Some numerical results are presented to support our answers.

II. METHODS

This section presents the essential tools, and these tools are based on our previous work [5–8]. Many details have been omitted; only the main statements are stated.

A. Closed-form representation of the iterative Landweber algorithm

Let us consider an iterative Landweber algorithm that minimizes the following L^2 -norm quadratic objective function:

$$f = (AX - P)^T (AX - P) + \beta (BX)^T (BX), \quad (1)$$

where X is the image array expressed as a vector, P is the projection array expressed as a vector, A is the imaging matrix, B is a regularization matrix, and β is a parameter controlling influence of the regularization term on the objective function f . The Landweber algorithm is a gradient descent algorithm, and the gradient of the objection (1) is given as

$$\nabla f = 2A^T AX - 2A^T P + 2\beta B^T BX. \quad (2)$$

The Landweber algorithm that minimizes (1) is given as

$$X^{(k)} = X^{(k-1)} + \alpha (A^T P - A^T AX^{(k-1)} - \beta B^T BX^{(k-1)}), \quad (3)$$

where $X^{(k)}$ is the result of X at the k th iteration and the relaxation parameter α controls the step size. For a quadratic objective function (1), the Landweber algorithm will converge if the relaxation parameter $\alpha > 0$ is small enough. The recursive expression (3) can be transformed to a non-recursive form as follows.

$$\begin{aligned}
 X^{(k)} &= X^{(k-1)} + \alpha A^T (P - AX^{(k-1)}) - \alpha \beta B^T B X^{(k-1)} \\
 &= \alpha A^T P + (I - \alpha A^T A - \alpha \beta B^T B) X^{(k-1)} \\
 &= \alpha A^T P + (I - \alpha A^T A - \alpha \beta B^T B) [\alpha A^T P + (I - \alpha A^T A - \alpha \beta B^T B) X^{(k-2)}] \\
 &= \left[\sum_{n=0}^{k-1} (I - \alpha A^T A - \alpha \beta B^T B)^n \right] \alpha A^T P + (I - \alpha A^T A - \alpha \beta B^T B)^k X^{(0)}.
 \end{aligned} \tag{4}$$

If the square matrix $(I - M)$ is non-singular, we have the identity:

$$\sum_{n=0}^{k-1} M^n = (I - M)^{-1} (I - M^k), \tag{5}$$

which can be readily verified by pre-multiplying $(I - M)$ on both sides. Thus the non-recursive form (4) can be further written as a closed form without the Σ sign:

$$\begin{aligned}
 X^{(k)} &= (\alpha A^T A + \alpha \beta B^T B)^{-1} \times [I - (I - \alpha A^T A - \alpha \beta B^T B)^k] \alpha A^T P + (I - \alpha A^T A - \alpha \beta B^T B)^k X^{(0)} \\
 &= (A^T A + \beta B^T B)^{-1} [I - (I - \alpha A^T A - \alpha \beta B^T B)^k] \alpha A^T P + (I - \alpha A^T A - \alpha \beta B^T B)^k X^{(0)}.
 \end{aligned} \tag{6}$$

Expression (6) can be treated as a linear system with two separate inputs: $A^T P$ and $X^{(0)}$. We will consider two special cases of matrix B below.

Case (i) $B = A$, minimum L^2 -norm solution with the norm in the projection domain:

Algorithm (6) becomes

$$X^{(k)} = \frac{1}{1+\beta} (A^T A)^{-1} [I - (I - \alpha(1+\beta) A^T A)^k] A^T P + (I - \alpha(1+\beta) A^T A)^k X^{(0)} \tag{7}$$

and as $k \rightarrow \infty$,

$$X^{(\infty)} = \frac{1}{1+\beta} (A^T A)^{-1} A^T P. \tag{8}$$

Case (ii) $B = I$, minimum L^2 -norm solution with the norm in the image domain:

Algorithm (6) becomes

$$X^{(k)} = (A^T A + \beta I)^{-1} [I - ((1 - \alpha\beta)I - \alpha A^T A)^k] A^T P + ((1 - \alpha\beta)I - \alpha A^T A)^k X^{(0)} \quad (9)$$

and as $k \rightarrow \infty$,

$$X^{(\infty)} = (A^T A + \beta I)^{-1} A^T P. \quad (10)$$

In (8), $(A^T A)^{-1} A^T$ is the generalized inverse of matrix A if $(A^T A)^{-1}$ exists. In (10), $(A^T A + \beta I)^{-1} A^T$ is another generalized inverse of matrix A if $(A^T A)^{-1}$ does not exist.

B. Fourier-domain representation

In tomography, the matrix $A^T A$ is a projection-backprojection operator. When it operates upon an image X , it convolves the image X by a two-dimensional (2D) $1/r$ kernel, where r is the distance to the origin [4]. In the Fourier domain, this $1/r$ convolution kernel corresponds to the $1/\|\omega\|$ transfer function [4].

This matrix domain and Fourier domain equivalence is true only for the ideal situation, where the projector A represents the exact Radon transform. A discrete projector A is an approximation of the Radon transform. A better approximation can be achieved by using a smaller pixel size and a larger image array. This paper assumes that this approximation is accurate enough for our discussion on the properties of the iterative Landweber algorithm.

Realizing that $A^T P$ is the pure backprojection of the sinogram P without ramp filtering, (8) and (10) are simply “backproject first, then filter” algorithms [4]. In (8) the post-filter is $\|\omega\|/(1+\beta)$, and in (10) the post-filter is $1/(1/\|\omega\| + \beta) = \|\omega\|/(1 + \beta\|\omega\|)$.

We can also write (7) and (9) in terms of filtering relationship in the Fourier domain. If the initial image $X^{(0)}$ is zero, the k th iteration of the iterative Landweber algorithm can be obtained by applying a post-filter to the pure backprojection of the sinogram by using the equivalence between the matrix $A^T A$ and the Fourier-domain transfer function $1/\|\omega\|$. We also notice that the identity matrix I corresponds to the transfer function 1 in the Fourier domain.

For (7), the corresponding post-filter's transfer function is

$$\frac{1}{1+\beta} \|\omega\| \left[1 - \left(1 - \frac{\alpha(1+\beta)}{\|\omega\|} \right)^k \right]. \quad (11)$$

For (9), the corresponding post-filter's transfer function is

$$\frac{1}{1/\|\vec{\omega}\| + \beta} \left[1 - \left((1 - \alpha\beta) - \frac{\alpha}{\|\vec{\omega}\|} \right)^k \right] = \frac{\|\vec{\omega}\|}{1 + \beta\|\vec{\omega}\|} \left[1 - \left((1 - \alpha\beta) - \frac{\alpha}{\|\vec{\omega}\|} \right)^k \right]. \quad (12)$$

C. Iterative Filtered Backprojection (FBP) Algorithm

Now we modify the objective function (1) by introducing two square matrices M and $\hat{M}$ as

$$f_M = (AX - P)^T M (AX - P) + \beta (BX)^T \hat{M} (BX). \quad (13)$$

The gradient of the objection (13) is

$$\nabla f_M = 2A^T M A X - 2A^T M P + 2\beta B^T \hat{M} B X. \quad (14)$$

The Landweber algorithm that minimize (13) is given as

$$X^{(k)} = X^{(k-1)} + \alpha (A^T M P - A^T M A X^{(k-1)} - \beta B^T \hat{M} B X^{(k-1)}) \quad (15)$$

Following the same steps as in (4) and (6), we obtain a closed form:

$$X^{(k)} = (A^T M A + \beta B^T \hat{M} B)^{-1} \times [I - (I - \alpha A^T M A - \alpha \beta B^T \hat{M} B)^k] A^T M P + (I - \alpha A^T M A - \alpha \beta B^T \hat{M} B)^k X^{(0)}. \quad (16)$$

If M is a diagonal matrix, it weighs the projections P and the forward projections at each iteration. The resultant iterative algorithm is a weighted Landweber algorithm. As a common practice, the weights can be assigned as the reciprocal of the projection noise variance.

It is well-known that one can represent linear convolution as multiplication by a Toeplitz matrix [16]. If M is a Toeplitz or block-Toeplitz matrix, it filters the projections. In the Fourier domain, this matrix M can be designed so that the resultant filter is a one-dimensional (1D) ramp filter $|\omega|$ or a Laplacian filter ω^2 and so on by using the Central Slice Theorem [4]. In tomography, post-filtering a backprojected image by a 2D ramp filter $\|\omega\|$ is equivalent to pre-filtering the projections by a 1D ramp filter $|\omega|$. Similarly, if the 2D filter is Laplacian $\|\omega\|^2$, its equivalent 1D counterpart is the 1D Laplacian ω^2 . This relationship can be extended to any radially symmetric 2D filters. This point will be made clearer in the following two examples.

1) *M is a ramp filter*: Let us consider the situation of M being a ramp filter and then (13) is an iterative filtered backprojection (iterative FBP) algorithm.

If we further assume that $B = A$, $M = \hat{M}$ and the initial image is zero, (16) becomes

$$X^{(k)} = (A^T M A + \beta A^T M A)^{-1} \times [I - (I - \alpha A^T M A - \alpha \beta A^T M A)^k] \times A^T M P, \quad (17)$$

where $A^T M P$ represents the conventional FBP algorithm, and $A^T M A = I$. Thus (17) is further reduced to

$$X^{(k)} = \frac{1 - (1 - \alpha - \alpha \beta)^k}{1 + \beta} A^T M P. \quad (18)$$

Algorithm (18) is not interesting, because the result is simply a scaled version of the FBP image with different scaling factors.

If we consider a different case that $B = I$, $\hat{M}$ is also a ramp filter, and the initial image is zero, (16) becomes

$$X^{(k)} = (I + \beta \hat{M})^{-1} [I - (I - \alpha I - \alpha \beta \hat{M})^k] A^T M P. \quad (19)$$

The Fourier-domain transfer function of the post-filter in (19) is given as

$$\frac{1 - (1 - \alpha - \alpha \beta \|\vec{\omega}\|)^k}{1 + \beta \|\vec{\omega}\|}. \quad (20)$$

This is a low-pass filter and its bandwidth is controlled by parameter k . When $k \rightarrow \infty$, the low-pass filter approaches $1/(1 + \beta \|\vec{\omega}\|)$.

We must point out that the parameter α in the matrix-domain expressions and in the Fourier-domain expressions are different. In (20), the convergence requirement is $|1 - \alpha - \alpha \beta \|\vec{\omega}\|| < 1$, that is, $0 < \alpha < 2/(1 + \beta \|\vec{\omega}\|)$. In discrete implementation, the highest frequency is 0.5. Therefore, parameter α can be chosen anywhere in $0 < \alpha < 2/(1 + 0.5\beta)$.

2) *M is a Laplacian filter*: In the Fourier domain, a Laplacian filter is the square of the ramp filter. In the spatial domain, it is the second order derivative.

We further assume that $B = A$ and the initial image is zero, (17) still applies. However, the Fourier-domain transfer function of the post-filter is given as

$$\frac{1 - (1 - (\alpha + \alpha\beta) \|\vec{\omega}\|)^k}{(1 + \beta) \|\vec{\omega}\|}. \quad (21)$$

This filter applies to the image $A^T MP$. Notice that $A^T MP$ is no longer the conventional FBP image, but is a ramp filtered version of the conventional FBP image. If we want to express the filter that applies to the conventional FBP image, (21) becomes

$$\frac{1 - (1 - (\alpha + \alpha\beta) \|\vec{\omega}\|)^k}{1 + \beta}. \quad (22)$$

We now consider a different case that $B = I$ and the initial image is zero, (19) is still valid. However, the Fourier-domain transfer function of the post-filter (applying to image $A^T MP$) is given as

$$\frac{1 - (1 - \alpha \|\vec{\omega}\| - \alpha\beta \|\vec{\omega}\|^2)^k}{(1 + \beta \|\vec{\omega}\|) \|\vec{\omega}\|}. \quad (23)$$

If want to express the filter applied to the conventional FBP image, (23) is modified as

$$\frac{1 - (1 - \alpha \|\vec{\omega}\| - \alpha\beta \|\vec{\omega}\|^2)^k}{1 + \beta \|\vec{\omega}\|}. \quad (24)$$

This is because when a “backprojection first, then filter” algorithm can be transformed into an FBP-type algorithm by replacing the post-backprojection filter $1/\|\vec{\omega}\|$ as the pre-backprojection filter $1/|\omega|$. In the Laplacian FBP, the conventional ramp filter $|\omega|$ is replaced by a parabola filter ω^2 . If we combine $1/|\omega|$ with the parabola filter ω^2 , we obtain the ramp filter $|\omega|$ again. Therefore, when M is a Laplacian, the Laplacian FBP can be implemented as the conventional FBP in the sense that a ramp filter is used as the pre-backprojection filter and using (24) instead of (23).

In (22), (23) and (24), the convergence requirement is $|1 - \alpha \|\vec{\omega}\| - \alpha\beta \|\vec{\omega}\|^2| < 1$, that is, $0 < \alpha < 2/(\|\vec{\omega}\| + \beta \|\vec{\omega}\|^2)$. In discrete implementation, the highest frequency is 0.5. Therefore, parameter α can be chosen anywhere in $0 < \alpha < 2/(0.5 + 0.25\beta)$. In general, if the relaxation parameter $\alpha > 0$ is small enough, the algorithm will converge.

Let $\beta = 0$; both (22) and (24) become

$$1 - (1 - \alpha \|\vec{\omega}\|)^k. \quad (25)$$

This is a high-pass filter and its bandwidth is controlled by parameter k . When $k \rightarrow \infty$, this high-pass filter's transfer function approaches 1. This is a very interesting situation and will be further discussed in the Result section.

3) M is a ramp filter and $\hat{M}$ is a Laplacian filter: Here we assume that the initial image is zero, B is the identity matrix, M is the ramp filter, and $\hat{M}$ is the Laplacian filter. The Fourier-domain transfer function of the post-filter is given as

$$\frac{1 - (1 - \alpha - \alpha\beta \|\vec{\omega}\|^2)^k}{1 + \beta \|\vec{\omega}\|^2}. \quad (26)$$

In (26), $1/(1 + \beta \|\vec{\omega}\|^2)$ is a low-pass filter and it looks like a Gaussian function. On the other hand, $1 - (1 - \alpha - \alpha\beta \|\vec{\omega}\|^2)^k$ is a high-pass filter and approaches the all-pass filter with $(1 - \alpha - \alpha\beta \|\vec{\omega}\|^2)^k \rightarrow 0$ as k increases.

III. Results: Answers to the two questions

A. Question #1: Is stopping early equivalent to minimum-norm regularization?

In this part we investigate a simple iterative algorithm with $\beta = 0$ and $M = I$. We want to compare its result at the k th iteration with the true minimum-norm solution ($\beta = 0$) with $k \rightarrow \infty$ and $M = I$. If these two results are related, we can find their relationship in the Fourier domain, using their post-filter's transfer functions.

We assume that their equivalent results can be obtained by first performing a conventional FBP reconstruction algorithm and then by applying a post filter. The post filters can be readily obtained by modifying the first term of the right-hand-side of (9), $[1 - (1 - \alpha\beta - \alpha\|\vec{\omega}\|)^k]/[1 + \beta\|\vec{\omega}\|]$.

For the case of the simple iterative algorithm result at the k th iteration, we let $\beta = 0$; the post filter is given as

$$1 - \left(1 - \frac{\alpha}{\|\vec{\omega}\|}\right)^k. \quad (27)$$

On the other hand, for the case of the true minimum-norm solution (8), the post filter is given as

$$\frac{1}{1+\beta\|\vec{\omega}\|}. \quad (28)$$

If these two results were the same, we would have

$$1-\left(1-\frac{\alpha}{\|\vec{\omega}\|}\right)^k \equiv \frac{1}{1+\beta\|\vec{\omega}\|}. \quad (29)$$

This implies that there exists a constant β such that

$$\beta = \frac{1}{\|\vec{\omega}\|} \frac{\left(1-\frac{\alpha}{\|\vec{\omega}\|}\right)^k}{1-\left(1-\frac{\alpha}{\|\vec{\omega}\|}\right)^k}, \forall \vec{\omega}. \quad (30)$$

This constant β does not exist, because the right-hand-side of (30) is a function of $\vec{\omega}$ and not a constant. Some plots of the right-hand-side of (30) are illustrated in Figure 1, which shows that the curves are not constants. This figure implies that equations (29) and (30) are not valid. Stopping early in a simple iterative Landweber algorithm is not equivalent to minimum-norm regulated solution.

B. Question #2: Do the low frequency components always converge first?

If the initial image is a constant, it is common to observe the following phenomenon: At the beginning of an iterative algorithm the images are smooth. As the iteration number gets larger, higher frequency components appear and the images become sharper. Is this always the case? The answer is No. Here we give a counter example of iterative FBP algorithm with $\beta=0$ and M as the Laplacian filter. The k th iteration result of this particular algorithm can be equivalently obtained by applying the post filter (25) to the result of the conventional FBP reconstruction. The transfer function (25) re-stated here:

$$1-(1-\alpha\|\vec{\omega}\|)^k, \quad (31)$$

which is a high-pass filter for each positive number k . Some curves of transfer function (25) are shown in Figure 2. It is observed that the transfer functions represented by (25) are indeed high-pass filters and the bandwidth gets wider as k increases.

As shown in Figure 2, when the iteration number k is low, (25) can be approximated by a ramp filter, which significantly enhances the high frequency components. When the iteration number k is large, (25) approaches an all-pass filter, which gradually brings in the low frequency components.

C. Computer simulations

Computer simulations were performed to show the effects of the filter used in an iterative FBP algorithm. The parallel-beam imaging geometry was assumed. Analytic calculation was used to evaluate the line integrals and 180 views over 180° were used in projection data generation. Transmission projection noise model was assumed. The number of detector bins on the detection was 256, and the size of the reconstructed images was 256×256 .

Figure 3 shows the results from an iterative FBP algorithm with a Laplacian filter $[-1, 2, -1]$ and $\alpha = 0.1$. The iteration number is noted by k . This example shows that the high-frequency components converge sooner than the low-frequency components.

As comparison, the conventional iterative Landweber reconstructions using the same noisy projection data set as in Figure 3 are shown in Figure 4, where the low-frequency components converge sooner than the high-frequency components.

The easiest way to see why the Laplacian filter can make the high frequency component converge sooner is to compare Figures 1 and 2. These two figures are frequency domain illustrations of how an iterative algorithm works. Figure 1 is for the normal iterative algorithm. When the iteration number k is low, the low frequency components are rich. When the iteration number k gets larger, the low frequency components are diminished. The opposite effects are illustrated in Figure 2, which is for the case of the Laplacian filter. When the iteration number k is low, the high frequency components are rich. When the iteration number k gets larger, the low frequency components are flourished.

These computer simulation results support our claim that there exists an iterative reconstruction algorithm that has an opposite convergence property that we commonly see. The higher frequency components can converge sooner than the lower frequency components.

In fact, we can design an iterative algorithm to make some desired frequency components to converge sooner. How to design such an algorithm is not in the scope of this paper. However, this flexibility can be illustrated by Figure 5, where the iterative FBP algorithm uses a filter $[-1, 2.1, -1]$. Its results are different from those in Figure 3 and also different from those in Figure 4.

The differences between Figures 3, 4, and 5 are the matrix M in the iterative algorithm (15). The results shown in Figure 4 use the identity matrix as M . This is a common application of the iterative algorithm. In the actual implementation, this 2D matrix is implemented as a 1D convolution with a kernel $[0, 1, 0]$. The images start out blurry and become sharper as the iteration number gets higher.

The results shown in Figure 3 use the Laplacian matrix as M . In the actual implementation, this 2D matrix M is implemented as a 1D convolution with a kernel $[-1, 2, -1]$. The images start out over-sharp and the low-frequency components gradually join in as the iteration number gets higher. We notice that as some high-frequency noise present in the image at the first iteration, the high-frequency noise remains in the image as the iteration number

increases. The noise does not get filtered out. This is a disadvantage of using an iterative algorithm in which the high-frequency components converge sooner.

The results shown in Figure 5 use a matrix M , whose equivalent 1D convolution kernel is $[-1, 2.1, -1]$. One can readily see that by slightly changing the convolution kernel, the iterative algorithm's behavior can be altered significantly. This might open up a new way in designing new iterative algorithms.

IV. Conclusions

Fourier-domain transfer function is a powerful tool. It is a novel approach to study linear iterative algorithms in the Fourier domain. It reveals that early stopping of the iteration is not equivalent to a minimum-norm solution. It also discovers that one can create an iterative FBP algorithm that makes the high-frequency components converge sooner. This second observation suggests that one can develop an iterative algorithm to make any desired frequency components converge sooner than other frequency components. It is not known yet whether these new iterative algorithms with frequency components preferences have any clinical applications.

Projection domain noise weighting for iterative reconstruction, in which the matrix M is a diagonal matrix, is shift variant and cannot be expressed as convolution. Thus, it does not have a simple Fourier-domain transfer function. In this paper, we do not consider noise weighting, which leads to a shift-variant system, which is not in the scope of this work. In our other papers [7, 8], the ray-by-ray projection noise weighting is considered by using the quantization method. For example, the noise weights are quantized into 10 values and 10 different Fourier-domain transfer functions are formed accordingly. If the weighting factor is w , the parameter α is scaled and becomes αw . In the backprojection procedure, only one of the 10 filtered values is backprojected in a noise-weighted FBP algorithm. Thus, the Fourier-domain components in the image consist of the mixture of the frequency components obtained from those 10 filters. Noise weighting does not change the frequency contents very much. Noise weighting only changes convergence rate. Increasing parameter α has a similar effect of increasing iteration number k .

References

1. Landweber L. An iteration formula for Fredholm integral equations of the first kind. *Amer J Math.* 1951; 73:615–624.
2. Combettes, PL., Pesquet, J-C. Proximal splitting methods in signal processing. In: Bauschke, HH, Burachik, RS, Combettes, PL, Elser, V, Luke, DR., Wolkowicz, H., editors. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer; New York: 2011. p. 185-212.
3. Herman, GT. *Fundamentals of computerized tomography: Image reconstruction from projections*. 2. Springer; 2009.
4. Zeng, GL. *Medical Imaging Reconstruction, A Tutorial*. Higher Education Press, Springer; Beijing: 2010.
5. Zeng GL. A filtered backprojection algorithm with characteristics of the iterative Landweber algorithm. *Med Phys.* 2012; 39:603–607. [PubMed: 22320769]
6. Zeng GL. A filtered backprojection MAP algorithm with nonuniform sampling and noise modeling. *Med Phys.* 2012; 39:2170–2178. [PubMed: 22482638]

7. Zeng GL, Zamyatin A. A filtered backprojection algorithm with ray-by-ray noise weighting. *Med Phys.* 40:031113. <http://dx.doi.org/10.1118/1.4790696> (7 pages), Online Publication Date: 28 February 2013, PMID: 23464293, PMCID: PMC5148087.
8. Zeng GL. Model-based filtered backprojection algorithm: A tutorial. *Biomedical Engineering Letters.* Mar; 2014 4(1):3–18. (<http://link.springer.com/article/10.1007/s13534-014-0121-7>). [PubMed: 25574421]
9. Bishop CM. Regularization and complexity control in feed-forward networks. *Proceedings International Conference on Artificial Neural Networks ICANN'95.* 1995; 1:141–148.
10. Sjöberg J, Ljung L. Overtraining, regularization and searching for a minimum, with application to neural networks. *International Journal of Control.* 1995; 62(6):1391–1407.
11. Goodfellow, I., Bengio, Y., Courville, A. *Deep Learning.* Cambridge, MA, USA: MIT Press; 2016.
12. Chang LT. A method for attenuation correction in radionuclide computed tomography. *IEEE Trans Nucl Sci.* 1978; 25:638–643.
13. Lalush DS, Tsui BMW. Improving the convergence of iterative filtered backprojection algorithms. *Med Phys.* 1994; 21:1283–1286. [PubMed: 7799872]
14. Sunnegårdh J, Danielsson P-E. Regularized iterative weighted filtered backprojection for helical cone-beam CT. *Med Phys.* 2008; 35:4173–85. [PubMed: 18841870]
15. Bertero, M., Boccacci, P. *Introduction to inverse problems in imaging.* Institute of Physics Publishing; Dirac House, Temple Back, Bristol BS1 6BE, UK: 1998. (see chapter 6)
16. Gray RM. Toeplitz and circulant matrices: A review. *Foundations and Trends in Communications and Information Theory.* 2006; 2(3):155–239.

Fig. 1.
The right-hand-side of (29) with $\alpha = 0.0001$ and 3 different k values.

Fig. 2.
The right-hand-side of (29) with $\alpha = 0.01$ and 4 different k values.

Fig. 3.

Results for an iterative FBP algorithm with $\alpha = 0.1$ and a Laplacian filter whose convolution kernel is $[-1, 2, -1]$. This example shows that the high-frequency components converge sooner than the low-frequency components.

Fig. 4.

Results for a conventional iterative Landweber algorithm with $\alpha = 0.001$. This example shows that the low-frequency components converge sooner than the high-frequency components.

Fig. 5.

Results for an iterative FBP algorithm with a filter kernel $[-1, 2.1, -1]$ and $\alpha = 0.01$.