SPECTRAL COMPUTED TOMOGRAPHY WITH LINEARIZATION AND PRECONDITIONING

YUNYI HU*, MARTIN S. ANDERSEN†, AND JAMES G. NAGY‡

Abstract. In the area of image sciences, the emergence of spectral computed tomography (CT) detectors highlights the concept of *quantitative imaging*, in which not only reconstructed images are offered, but also weights of different materials that compose the object are provided. If a detector is made up of several energy windows and each energy window is assumed to detect a specific range of energy spectrum, then a nonlinear matrix equation is formulated to represent the discretized process of attenuation of x-ray intensity. In this paper, we present a linearization technique to transform this nonlinear equation into an optimization problem that is based on a weighted least squares term and a nonnegative bound constraint. To solve this optimization problem, we propose a new preconditioner that can significantly reduce the condition number, and with this preconditioner, we implement a highly efficient first order method, Fast Iterative Shrinkage-Thresholding Algorithm (FISTA), to achieve substantial improvements on convergence speed and image quality. We also use a combination of generalized Tikhonov regularization and ℓ_1 regularization to stabilize the solution. With the introduction of new preconditioning, a linear inequality constraint is introduced. In each iteration, we decompose this constraint into small-sized problems that can be solved with fast optimization solvers. Numerical experiments illustrate convergence, effectiveness and significance of the proposed method.

Key words. preconditioning, digital image reconstruction, FISTA, beam-hardening artifacts, spectral computed tomography, bound constraints

AMS Subject Classifications: 65F22, 65F10, 49N45, 65K99

1. Introduction.

The development of new energy-windowed spectral computed tomography (CT) machines have received a great deal of interest in recent years; see, e.g. [1, 24]. These detectors assume that x-rays emitted by the x-ray source are composed of a spectrum of different energies, and in each energy window, the detector can detect a specific range of energy. Moreover, it assumes that the detector can perform photon counting and the data collected by the detector are nonnegative integers. Compared with traditional CT machines, we can avoid introducing beam-hardening artifacts [19] and improve quality of reconstructed images. To reconstruct images of an object, we need to solve a nonlinear equation

$$\boldsymbol{Y} = \exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}, \tag{1.1}$$

where $\boldsymbol{Y}$ is a matrix that gathers the projected data of each energy window in the corresponding column and the exponential operator is applied element-wise (i.e., it is not a matrix function). $\boldsymbol{A}$ is a matrix that is related to the quantitative information of ray trace and $\boldsymbol{C}$ is a matrix that contains linear attenuation coefficients for particular (known) materials at specified energies. $\boldsymbol{S}$ is the matrix that accumulates the spectrum energies for each energy window in the corresponding column. We assume that $\boldsymbol{S}$ is square and invertible. Moreover, $\boldsymbol{\mathcal{E}}$ represents the noise term and we assume that $E_{il} \sim \mathcal{N}(0, y_{il})$ for each component E_{il} in $\boldsymbol{\mathcal{E}}$ and y_{il} in $\boldsymbol{Y}$. We assume that these data are known and the target is to solve the unknown weight matrix $\boldsymbol{W}$. $\boldsymbol{W}$ is of

*Department of Mathematics and Computer Science, Emory University. Email: yhu85@mathcs.emory.edu.

†Department of Applied Mathematics and Computer Science, Technical University of Denmark, Email: mskan@dtu.dk.

‡Department of Mathematics and Computer Science, Emory University. Email: nagy@mathcs.emory.edu.

the size N_v by N_m, where N_v is the number of voxels (pixels if 2D) for each material map and N_m is the number of materials. Since the weight matrix $\boldsymbol{W}$ represents the material maps of different materials, then it must be nonnegative and we need to add a lower bound $\boldsymbol{W} \geqslant \boldsymbol{0}$.

To solve Equation (1.1), we want to vectorize it at first. Then we use the Taylor expansion to remove the point-wise exponential function and obtain an approximate linearized equation. Under the Gaussian assumption, as we show in Section 2, we can transform this equation into a weighted least squares problem under bound constraints:

$$\begin{aligned} &\min_{\boldsymbol{w}} && \frac{1}{2}\|\mathcal{A}\boldsymbol{w}-\boldsymbol{b}\|^2_{\boldsymbol{\Sigma}^{-1}} \\ &\text{subject to} && \boldsymbol{w} \geqslant \boldsymbol{0}, \end{aligned} \tag{1.2}$$

where $\mathcal{A} = \boldsymbol{C} \otimes \boldsymbol{A}$, $\boldsymbol{b} = -\log(\boldsymbol{y})$, $\boldsymbol{y} = \operatorname{vec}(\boldsymbol{Y})$ and $\boldsymbol{w} = \operatorname{vec}(\boldsymbol{W})$. $\boldsymbol{\Sigma}^{-1}$, which combines information from $\boldsymbol{S}$ and $\boldsymbol{y}$, is the inverse covariance matrix generated by Gaussian noise and log transformation. $\|\cdot\|^2_{\boldsymbol{\Sigma}^{-1}}$ represents a weighted 2-norm and $\|\mathcal{A}\boldsymbol{w}-\boldsymbol{b}\|^2_{\boldsymbol{\Sigma}^{-1}} = (\mathcal{A}\boldsymbol{w}-\boldsymbol{b})^T \boldsymbol{\Sigma}^{-1} (\mathcal{A}\boldsymbol{w}-\boldsymbol{b})$. $\boldsymbol{C}$ is of the size N_e by N_m, where N_e is the number of energy and N_m is the number of materials. Since each column of C collects the corresponding linear attenuation coefficients and two materials, such as adipose and glandular, might be similar to each other, the matrix C is likely to be ill-conditioned. On the other hand, Problem (1.2) is similar to a quadratic programming problem under bound constraints. However, direct implementation of an optimization solver does not provide high-quality reconstruction because the ray trace matrix $\boldsymbol{A}$ is large and ill-conditioned, and the columns of the linear attenuation coefficient matrix $\boldsymbol{C}$ might be nearly collinear.

Because of the ill-posedness, Barber et al. [1] proposed a preconditioner based on the eigenvalue decomposition of the matrix product of linear attenuation coefficients, $\boldsymbol{C}^T\boldsymbol{C}$, to orthogonalize columns of $\boldsymbol{C}$. They also suggest using a Poisson noise assumption and construct loss functions that are either based on the maximum likelihood estimator (MLE) or the nonlinear least squares term. Using these types of loss functions and the proposed preconditioner, a *Chambolle-Pock* (CP) primal-dual method [5] is implemented to solve the corresponding optimization problem. However, because the MLE for the Poisson model is nonlinear, it is not obvious to see how this preconditioner can reduce the condition number of the Hessian matrix. Moreover, because each iteration of a second order method for large three-dimensional imaging problems is very costly (in terms of both the computations and storage requirements), in this paper we consider first order methods. With a first order method, it is not necessary to construct either the Hessian or Hessian-vector multiplication in each step.

To mitigate the ill-posedness, we propose a new preconditioner that is based on a rank-1 approximation of the matrix $\boldsymbol{Y}$. With this rank-1 approximation, we can estimate the Hessian of the objective function in (1.2) by a Kronecker product of two parts. The first part of this Kronecker product is of the size $N_m \times N_m$, where N_m denotes the number of materials; usually this is quite small, e.g. $N_m = 2$ or 3. This matrix product is also symmetric and positive definite so we can construct a preconditioner from its inverse Cholesky factorization, and thus transform it into an identity in the preconditioned system. Because the conditioning of the Hessian is closely related to these two matrices and one of them has been transformed into an identity, we have reduced the condition number significantly. Moreover, it is an economical preconditioner since we only need to compute the preconditioner once and can reuse it in the future iterations. The preconditioner proposed in [1] includes only the data of $\boldsymbol{C}$, the matrix of linear attenuation coefficients of material and energy. Compared with this, the preconditioner proposed in this paper includes the information of linear attenuation coefficients, the energy spectrum and photon counting data. It offers a more physically meaningful approximation of the Hessian.

In addition, with the weighted least squares objective function, it is much easier to analyze the condition number before and after preconditioning. Since the performance of a first order method is closely related to the condition number of the Hessian, it is intuitive to implement a first order method if we can reduce the condition number significantly. Based on this idea, Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [2, 21, 20] comes into view. FISTA is a first order method that has an "optimal" function convergence rate, $\mathcal{O}\left(1/k^2\right)$, where k is the number of iterations. Furthermore, this method is suitable for solving problems that have a form of $f\left(\boldsymbol{x}\right) + g\left(\boldsymbol{x}\right)$ where both $f\left(\boldsymbol{x}\right)$ and $g\left(\boldsymbol{x}\right)$ are convex but $g\left(\boldsymbol{x}\right)$ is possibly nonsmooth. This $f\left(\boldsymbol{x}\right)$ can be the weighted least squares term in Problem (1.2) and $g\left(\boldsymbol{x}\right)$ can represent a nonsmooth regularization term such as ℓ_1 regularization or nonnegative constraints. Even if we can achieve fast convergence, the introduction of a preconditioner complicates the bound constraints. The previous bound constraints have become linear inequality constraints because of the preconditioner. However, we can construct a projection problem that can find the closest solutions to satisfy these constraints. Moreover, this projection problem is separable and we can apply highly efficient solvers to compute the solutions to these decomposed small-sized problems. Generally speaking, the implementations of our preconditioner, FISTA and projection problem compliment each other and exhibit high-quality reconstructed images and fast convergence results.

This paper is organized as follows. In Section 2, we review the continuous energy-windowed spectral CT model and the corresponding discretized nonlinear matrix equation. The linearization, vectorization and set-up of the optimization problem are also included in Section 2. The key idea of this paper, preconditioning, is introduced in Section 3. In this section, both the derivation of our preconditioner and an analysis of the reduction of the condition number are presented. The choice of regularization will be exhibited in this section as well. In Section 4, we study FISTA and how we construct and solve the projection problems. Moreover, numerical experiments are presented in Section 5 and concluding remarks are given in Section 6.

2. The Energy-windowed Spectral CT Model.

In this section, we start with an introduction to the basic model. Then we show how to discretize this model to obtain a matrix equation. Since we do not want to solve this matrix equation directly, we therefore vectorize this equation and take the Taylor expansion to the first order term to remove the exponential function. In this case, we can obtain a linear system with transformed noise. With this transformed noise, we can build a weighted least squares optimization problem under bound constraints.

In computed tomography (CT), source x-ray beams are composed of a spectrum of different energies [4]. Recent technological developments have resulted in the design of new photon counting detectors that can discriminate the measured data into specific energy windows. Image reconstruction algorithms that exploit this information can avoid introducing beam-hardening artifacts, obtain material decomposition and improve the quality of reconstructed images. The mathematical model for image reconstruction uses Beer's law [12], which states that the change of x-ray intensity before and after illumination through the object is

$$y_i^{(k)} = \int_E S^{(k)}(e) \exp\left(- \int_{t \in l} \mu\left(\vec{r}\,(t)\,, e\right) \mathrm{d}\, t \right) \mathrm{d}\, e + \eta_i^{(k)}, \quad \begin{cases} i = 1, 2, \cdots, N_d \times N_p, \\ k = 1, 2, \cdots, N_b, \end{cases} \tag{2.1}$$

where

- $y_i^{(k)}$ is x-ray intensity of the i-th pixel in the k-th detector bin.
- E is the photon flux density. Figure 5.2 shows a curve of E versus photon energy.
- N_d is the number of detector pixels. For a material map of the size n by n, we assume $N_d = n$.
- N_p is the number of projections. For cone/fan beam CT, projections are uniformly distributed from 0 to 360 degrees.
- N_b is the number of detector bins. For an energy-windowed CT machine, we usually assume that it has 5 to 6 energy bins.
- $S^{(k)}(e)$ represents photon flux density for the k-th detector bin, which is the number of incident photons at the energy e in the k-th energy window.
- $\mu\left(\vec{r}\,(t)\,, e\right)$ denotes the linear attenuation coefficient that is related to the position function $\vec{r}\,(t)$ and the energy level e.
- $\eta_i^{(k)}$ is the error term for the i-th element in the k-th energy bin and it is assumed to be Gaussian for this model.

In Equation (2.1), the unknown linear attenuation coefficient $\mu\left(\vec{r}\,(t)\,, e\right)$ is dependent on the position function $r\,(t)$ and the energy levels e. If the object is assumed to be composed of several different materials, then a material expansion is introduced to further decompose the function $\mu\left(\vec{r}\,(t)\,, e\right)$ [11]:

$$\mu\left(\vec{r}\,(t)\,, e\right) = \sum_{m=1}^{N_m} u_{m,e} w_m\left(\vec{r}\right), \tag{2.2}$$

where

- N_m is the number of materials that form the object.
- $u_{m,e}$ is the linear attenuation coefficient for the m-th material at the energy level e.
- $w_m\left(\vec{r}\right)$ is the unknown weight of the m-th material at the position $\vec{r}$.

With this decomposition, the unknown variable has been shifted from $\mu\left(\vec{r}\,(t)\,, e\right)$ to the weight fraction $w_m\left(\vec{r}\right)$. If we also assume that $w_m\left(\vec{r}\right)$ can be represented as a sum of product of weights and basis functions $\phi_j\left(\vec{r}\right)$, then another expansion can be expressed by

$$w_m\left(\vec{r}\right) = \sum_{j=1}^{N_v} w_{j,m} \phi_j\left(\vec{r}\right), \tag{2.3}$$

where

- N_v is the number of voxels (pixels if 2D) of images that compose the object.
- $w_{j,m}$ is the weight fraction of the m-th material in the j-th voxel (pixels if 2D).
- $\phi_j\left(\vec{r}\right)$ is the basis function of image representation. The line integral of the basis function, $a_{i,j}$, is the length of the x-ray beam through the j-th voxel (pixel if 2D), incident onto the i-th element of the product of detector pixels

N_d and the number of projections N_p:

$$a_{i,j} = \int_{t\in l} \phi_j\left(\vec{r}\,(t)\right) \mathrm{d}\,t. \tag{2.4}$$

Then the line integral in Equation (2.1) can be simplified by Expansion (2.3) and Integral (2.4):

$$\int_{t\in l} \mu\left(\vec{r}\,(t)\,, e\right) \mathrm{d}\,t = \sum_{m=1}^{N_m}\sum_{j=1}^{N_v} u_{m,e} w_{j,m} \int_{t\in l} \phi_j\left(\vec{r}\,(t)\right) \mathrm{d}\,t = \sum_{j=1}^{N_v}\sum_{m=1}^{N_m} a_{i,j} w_{j,m} u_{m,e}. \tag{2.5}$$

If we also discretize the integral over the energy E and ignore quadrature errors, then the discrete model of Equation (2.1) can be written as:

$$y_i^{(k)} = \sum_{e=1}^{N_e} s_e^{(k)} \exp\left(-\sum_{j=1}^{N_v}\sum_{m=1}^{N_m} a_{i,j} w_{j,m} u_{m,e}\right) + \eta_i^{(k)}, \tag{2.6}$$

where N_e is the number of discrete energies. If we collect $a_{i,j}$, $w_{i,j}$ and $u_{m,e}$ in a matrix form and concatenate $y_i^{(k)}$, $s_e^{(k)}$, $\eta_i^{(k)}$ with respect to their energy windows, then the corresponding matrix equation of (2.6) can be represented as:

$$\boldsymbol{Y} = \exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}, \tag{2.7}$$

where

- $\boldsymbol{Y}$ is a matrix of the size $(N_d \cdot N_p) \times N_b$ that gathers x-ray photons of each energy window in the corresponding column.
- $\boldsymbol{A}$ is a matrix of the size $(N_d \cdot N_p) \times N_v$ that collects the fan-beam geometry and each element corresponds to $a_{i,j}$.
- $\boldsymbol{C}$ is a matrix of the size $N_e \times N_m$ that accumulates linear attenuation coefficients and each entry corresponds to $u_{e,m}$, the linear attenuation coefficient of the energy e and the m-th material.
- $\boldsymbol{S}$ is a matrix of the size $N_e \times N_b$ and each column collects the spectrum energy of a specific range. In the forward problem, we use the full spectrum, but when we solve the inverse problem, the average in each energy window is used to represent the corresponding spectral energy. Therefore, $N_b = N_e$ for the inverse problem and $\boldsymbol{S}$ is an invertible diagonal matrix because the means are placed in the diagonal. A detailed example is shown in Figure 5.2.
- $\boldsymbol{\mathcal{E}}$ is the noise matrix that is of the size $(N_d \cdot N_p) \times N_b$. The assumption for the noise is $E_{il} \sim \mathcal{N}\left(0, y_{il}\right)$ for each element E_{il} in $\boldsymbol{\mathcal{E}}$ and y_{il} in $\boldsymbol{Y}$.

In Equation (2.7), the exponential operator is applied element-wise (i.e., it is not a matrix function). In addition to Equation (2.7), we also require that weight fractions should be nonnegative and this can be illustrated by the constraint $\boldsymbol{W} \geqslant \boldsymbol{0}$.

In several cases, the composition of materials can be similar. For example, glandular and adipose have similar attenuation coefficients at the same energy level and it causes the collinearity. After discretization, the columns of $\boldsymbol{C}$ can be nearly dependent. Moreover, $\boldsymbol{A}$ is large-scale and sparse and it is highly likely to have small singular values. As we will see later, the Hessian system involves the Kronecker product $\boldsymbol{C} \otimes \boldsymbol{A}$ and it can cause the ill-posedness. Since it is challenging to solve this equation directly, it is important to consider approaches to facilitate the process. First, we can introduce a preconditioning matrix $\boldsymbol{M}$ into Equation (2.7):

$$\boldsymbol{Y} = \exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{M}^{-T}\boldsymbol{M}^T\boldsymbol{C}^T\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}. \tag{2.8}$$

If we let $\tilde{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{M}^{-T}$ and $\tilde{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{M}$, then Equation (2.8) is equivalent to

$$\boldsymbol{Y} = \exp\left(-\boldsymbol{A}\tilde{\boldsymbol{W}}\tilde{\boldsymbol{C}}^T\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}. \tag{2.9}$$

So far, we have not introduced how to choose the preconditioner $\boldsymbol{M}$. The choice of $\boldsymbol{M}$ depends on linearization and approximation. In Section (3.1), we will state the process in detail, and in the new coordinate system defined by $\boldsymbol{M}$, the corresponding Hessian will be better conditioned. With the help of the preconditioning matrix $\boldsymbol{M}$, we have transformed the original system of solving $\boldsymbol{W}$ into the new system of solving $\tilde{\boldsymbol{W}}$. Since each entry of $\tilde{\boldsymbol{W}}$ is a linear combination of all entries in the corresponding row of $\boldsymbol{W}$, we can try to find a matrix $\boldsymbol{M}$ such that the new system is better conditioned than the original one.

On the other hand, we do not want to solve the nonlinear matrix equation (2.9) directly because it might introduce a tensor when we compute second order derivatives. In this case, we want to vectorize Equation (2.9) on both sides and linearize it to construst a weighted least squares optimization problem. In the forward problem, we use the full spectrum and the matrix $\boldsymbol{S}$ is then usually rectangular. When we solve the inverse problem, we choose the average in each energy window to represent the corresponding energy spectrum. In this case, $N_b = N_e$ and the matrix $\boldsymbol{S}$ in the inverse problem is a nonsingular diagonal matrix. So we can multiply $\boldsymbol{S}^{-1}$ on both sides of (2.9):

$$\boldsymbol{Y}\boldsymbol{S}^{-1} = \exp\left(-\boldsymbol{A}\tilde{\boldsymbol{W}}\tilde{\boldsymbol{C}}^T\right) + \boldsymbol{\mathcal{E}}\boldsymbol{S}^{-1}. \tag{2.10}$$

Vectorizing both sides of (2.10), and using properties of Kronecker products, we obtain

$$\left(\boldsymbol{S}^{-T} \otimes \boldsymbol{I}\right)\boldsymbol{y} = \exp\left\{-\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right)\tilde{\boldsymbol{w}}\right\} + \left(\boldsymbol{S}^{-T} \otimes \boldsymbol{I}\right)\boldsymbol{e}, \tag{2.11}$$

where $\boldsymbol{y} = \text{vec}(\boldsymbol{Y})$, $\tilde{\boldsymbol{w}} = \text{vec}(\tilde{\boldsymbol{W}})$ and $\boldsymbol{e} = \text{vec}\left(\boldsymbol{E}\right)$. If we let $\tilde{\boldsymbol{y}} = \left(\boldsymbol{S}^{-T} \otimes \boldsymbol{I}\right)\boldsymbol{y}$ and $\tilde{\boldsymbol{e}} = \left(\boldsymbol{S}^{-T} \otimes \boldsymbol{I}\right)\boldsymbol{e}$, then we can subtract $\tilde{\boldsymbol{e}}$ on both sides of (2.11) and obtain

$$\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{e}} = \exp\left\{-\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right)\tilde{\boldsymbol{w}}\right\}. \tag{2.12}$$

By taking the logarithm on both sides of Equation (2.12), we can obtain a linear equation

$$\log\left(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{e}}\right) = -\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right)\tilde{\boldsymbol{w}}. \tag{2.13}$$

However, the left-hand side of Equation (2.13) contains the transformed error term $\tilde{\boldsymbol{e}}$ so we cannot solve this equation directly. In this case, we can separate the error term $\tilde{\boldsymbol{e}}$ from $\tilde{\boldsymbol{y}}$ using a first order Taylor expansion at $\tilde{\boldsymbol{y}}$:

$$\log\left(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{e}}\right) = \log\left(\tilde{\boldsymbol{y}}\right) - \text{diag}\left(\tilde{\boldsymbol{y}}\right)^{-1}\tilde{\boldsymbol{e}} + \mathcal{O}\left(\|\tilde{\boldsymbol{e}}\|_2^2\right). \tag{2.14}$$

If we use the first two terms on the right-hand side of Equation (2.14) to estimate the term $\log\left(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{e}}\right)$, then Equation (2.13) can be expressed by a linear equation with the error term $\text{diag}\left(\tilde{\boldsymbol{y}}\right)^{-1}\tilde{\boldsymbol{e}}$. Let $\boldsymbol{b} = -\log\left(\tilde{\boldsymbol{y}}\right)$, then Equation (2.13) is approximately equal to

$$\boldsymbol{b} \approx \left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right)\tilde{\boldsymbol{w}} - \text{diag}\left(\tilde{\boldsymbol{y}}\right)^{-1}\tilde{\boldsymbol{e}}. \tag{2.15}$$

With this equation and the Gaussian assumption of noise $\boldsymbol{e} \sim \mathcal{N}\left(\mathbf{0},\ \operatorname{diag}\left(\boldsymbol{y}\right)\right)$, we have

$$\boldsymbol{b}|\tilde{\boldsymbol{w}} \sim \mathcal{N}\left(\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right)\tilde{\boldsymbol{w}},\ \boldsymbol{\Sigma}\right), \tag{2.16}$$

where the noise covariance matrix $\boldsymbol{\Sigma}$ is expressed by

$$\boldsymbol{\Sigma} = \operatorname{diag}\left(\tilde{\boldsymbol{y}}\right)^{-1}\left(\boldsymbol{S}^{-T} \otimes \boldsymbol{I}\right)\operatorname{diag}\left(\boldsymbol{y}\right)\left(\boldsymbol{S}^{-1} \otimes \boldsymbol{I}\right)\operatorname{diag}\left(\tilde{\boldsymbol{y}}\right)^{-1}\ , \tag{2.17}$$

and the inverse covariance matrix is given by

$$\boldsymbol{\Sigma}^{-1} = \operatorname{diag}\left(\tilde{\boldsymbol{y}}\right)\left(\boldsymbol{S} \otimes \boldsymbol{I}\right)\operatorname{diag}\left(\boldsymbol{y}\right)^{-1}\left(\boldsymbol{S}^{T} \otimes \boldsymbol{I}\right)\operatorname{diag}\left(\tilde{\boldsymbol{y}}\right). \tag{2.18}$$

Since $\boldsymbol{Y}$ is a matrix that collects the number of photons of each energy window in the corresponding column, each entry of $\boldsymbol{Y}$ is a positive integer whose value can be in the order of hundreds of thousands. As long as the noise does not dominate the projected data, we expect the entries of $\tilde{\boldsymbol{y}}$ will be larger than zero. From Expression (2.18), we can see that the structure of $\boldsymbol{\Sigma}^{-1}$ depends on the structure of the matrix $\boldsymbol{S}$. If $\boldsymbol{S}$ is diagonal, then $\boldsymbol{\Sigma}$ is also diagonal. If we let $\mathcal{A} = \tilde{\boldsymbol{C}} \otimes \boldsymbol{A}$, then (see, e.g., [3]) the best unbiased linear estimator of $\tilde{\boldsymbol{w}}$ for the Gaussian model (2.16) is the solution of

$$\min_{\tilde{\boldsymbol{w}}} \frac{1}{2}\left(\mathcal{A}\tilde{\boldsymbol{w}} - \boldsymbol{b}\right)^{T}\boldsymbol{\Sigma}^{-1}\left(\mathcal{A}\tilde{\boldsymbol{w}} - \boldsymbol{b}\right). \tag{2.19}$$

In addition, we require that $\boldsymbol{W} \geqslant \mathbf{0}$, and with the preconditioner, these constraints are transformed into $\left(\boldsymbol{M} \otimes \boldsymbol{I}\right)\tilde{\boldsymbol{w}} \geqslant \mathbf{0}$. Therefore, we can formulate a weighted least squares problem under bound constraints

$$\begin{array}{ll} \displaystyle\min_{\tilde{\boldsymbol{w}}} & \dfrac{1}{2}\|\mathcal{A}\tilde{\boldsymbol{w}} - \boldsymbol{b}\|^2_{\boldsymbol{\Sigma}^{-1}} \\ \text{subject to} & \left(\boldsymbol{M} \otimes \boldsymbol{I}\right)\tilde{\boldsymbol{w}} \geqslant \mathbf{0}. \end{array} \tag{2.20}$$

In (2.20) the norm $\|\cdot\|^2_{\boldsymbol{\Sigma}^{-1}}$ corresponds to the weighted inner product given in (2.19). From this expression, we know that the objective function is convex. Moreover, the inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$ is diagonal as long as $\boldsymbol{S}$ is diagonal and this optimization problem has linear inequality constraints. Based on these observations, we can identify four challenges involved in solving this optimization problem. At first, we need to choose an appropriate preconditioning matrix to reduce the ill-conditioning of the Hessian. Secondly, we want to select suitable regularizations for the corresponding materials. Thirdly, we have to find an efficient method to solve the constrained weighted least squares problem. These three challenges are related to each other and an appropriate preconditioner with appropriate regularizations will be beneficial for the solver efficiency. Finally, we should handle linear inequality constraints in an efficient way. We will address these four challenges in the following sections.

3. Preconditioning and Regularization.

3.1. Preconditioning. The choice of the preconditiong matrix $\boldsymbol{M}$ is crucial for solving the optimization problem (2.20). If we do not have a preconditioner or we choose the preconditioner $\boldsymbol{M}$ as identity, the original Hessian for the weighted least squares problem (2.20) is expressed by

$$\boldsymbol{H} = \left(\boldsymbol{C}^{T} \otimes \boldsymbol{A}^{T}\right)\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{C} \otimes \boldsymbol{A}\right). \tag{3.1}$$

An appropriate preconditioner can transform the original ill-posed system into a better-conditioned system and thus bring faster convergence speed as well as higher quality of reconstructed images. In general, the preconditioned Hessian $\tilde{\boldsymbol{H}}$ can be represented as

$$\tilde{\boldsymbol{H}} = \mathcal{A}^T \boldsymbol{\Sigma}^{-1} \mathcal{A} = \left(\tilde{\boldsymbol{C}}^T \otimes \boldsymbol{A}^T\right) \boldsymbol{\Sigma}^{-1} \left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right), \tag{3.2}$$

where $\tilde{\boldsymbol{C}} = \boldsymbol{CM}$. From this expression, it is still not obvious how to construct the preconditioner. However, if we can separate the noise covariance matrix $\boldsymbol{\Sigma}^{-1}$ into a Kronecker product of two terms, we can merge several terms using properties of the Kronecker product and transform parts of the Hessian into identity with the help of $\boldsymbol{M}$. To realize this idea, we review the expression of $\boldsymbol{\Sigma}^{-1}$ in Equation (2.18), where we can see that it contains the Kronecker products $\boldsymbol{S} \otimes \boldsymbol{I}$ and $\boldsymbol{S}^T \otimes \boldsymbol{I}$ and it is not necessary to separate these two terms. So we focus on the other terms that include $\operatorname{diag}\{\tilde{\boldsymbol{y}}\}$ and $\operatorname{diag}\{\boldsymbol{y}\}^{-1}$. By definition, these two terms are related to each other by $\tilde{\boldsymbol{y}} = \left(\boldsymbol{S}^{-T} \otimes \boldsymbol{I}\right) \boldsymbol{y}$. In this case, if we can express $\operatorname{diag}\{\boldsymbol{y}\}$ into a Kronecker product of two terms, then we will reach the goal.

Recall that $\boldsymbol{y} = \operatorname{vec}(\boldsymbol{Y})$. Therefore, if we can find two rank-1 matrices, $\boldsymbol{u}$ and $\boldsymbol{v}$, such that $\boldsymbol{Y} \approx \boldsymbol{uv}^T$, then

$$\operatorname{diag}\{\boldsymbol{y}\} \approx \operatorname{diag}\left\{\operatorname{vec}\left(\boldsymbol{uv}^T\right)\right\} = \operatorname{diag}\{\boldsymbol{v}\} \otimes \operatorname{diag}\{\boldsymbol{u}\}. \tag{3.3}$$

These two rank-1 matrices can be obtained by solving a nearest Kronecker product (NKP) problem, which is equivalent to a rank-1 approximation of $\boldsymbol{Y}$ in terms of the Frobenius norm:

$$\min_{\boldsymbol{u},\ \boldsymbol{v}} \|\boldsymbol{Y} - \boldsymbol{uv}^T\|_F. \tag{3.4}$$

The solution to this problem has been studied extensively [23]. Using the singular value decomposition (SVD), one solution to Problem (3.4) can be expressed by $\boldsymbol{u} = \sqrt{\sigma_1}\boldsymbol{u}_1$ and $\boldsymbol{v} = \sqrt{\sigma_1}\boldsymbol{v}_1$, where $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ are the first left and right singular vectors and σ_1 is the corresponding largest singular value of $\boldsymbol{Y}$. Since we only need these terms rather than a full SVD, we can use MATLAB's `svds` function, or other efficient approaches, such as "PROPACK" [14], to calculate only σ_1, $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$.

After we have obtained $\boldsymbol{u}$ and $\boldsymbol{v}$, we can estimate the matrix $\operatorname{diag}\{\boldsymbol{y}\}$ as a Kronecker product of two terms as Equation (3.3). In addition, the term $\operatorname{diag}\{\tilde{\boldsymbol{y}}\}$ can be represented as

$$\begin{aligned}\operatorname{diag}\{\tilde{\boldsymbol{y}}\} &= \operatorname{diag}\left\{\left(\boldsymbol{S}^{-T} \otimes \boldsymbol{I}\right) \operatorname{vec}(\boldsymbol{Y})\right\} \approx \operatorname{diag}\left\{\left(\boldsymbol{S}^{-T} \otimes \boldsymbol{I}\right) \operatorname{vec}\left(\boldsymbol{uv}^T\right)\right\} \\ &= \operatorname{diag}\left\{\operatorname{vec}\left(\boldsymbol{uv}^T \boldsymbol{S}^{-1}\right)\right\} = \operatorname{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\} \otimes \operatorname{diag}\{\boldsymbol{u}\}.\end{aligned} \tag{3.5}$$

If we substitute the terms in (3.3) and (3.5) for the same terms in (2.18), we can obtain that

$$\boldsymbol{\Sigma}^{-1} \approx \left(\operatorname{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\} \boldsymbol{S} \operatorname{diag}\{\boldsymbol{v}\}^{-1} \boldsymbol{S}^T \operatorname{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\}\right) \otimes \operatorname{diag}\{\boldsymbol{u}\}. \tag{3.6}$$

So the preconditioned Hessian matrix is given by

$$\begin{aligned}\tilde{\boldsymbol{H}} &= \left(\tilde{\boldsymbol{C}}^T \otimes \boldsymbol{A}^T\right) \boldsymbol{\Sigma}^{-1} \left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) \\ &\approx \left(\tilde{\boldsymbol{C}}^T \otimes \boldsymbol{A}^T\right) \left[\operatorname{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\} \boldsymbol{S} \operatorname{diag}\{\boldsymbol{v}\}^{-1} \boldsymbol{S}^T \operatorname{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\} \otimes \operatorname{diag}\{\boldsymbol{u}\}\right] \left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) \\ &= \left(\tilde{\boldsymbol{C}}^T \operatorname{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\} \boldsymbol{S} \operatorname{diag}\{\boldsymbol{v}\}^{-1} \boldsymbol{S}^T \operatorname{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\} \tilde{\boldsymbol{C}}\right) \otimes \left(\boldsymbol{A}^T \operatorname{diag}\{\boldsymbol{u}\} \boldsymbol{A}\right).\end{aligned} \tag{3.7}$$

Since the size of $\tilde{\boldsymbol{C}}$ is $N_e \times N_m$, then the first part of the Kronecker product in (3.7) is a square matrix of the size $N_m \times N_m$. In other words, this part only depends on the number of materials that compose the object. Usually, we only consider 2 or 3 materials for the object so that the size of the matrix products for this part is usually either 2×2 or 3×3. Moreover, the matrix $\boldsymbol{Y}$ gathers the number of photons of each energy window in the corresponding column so all of its entries are positive integers. In this case, we can choose $\boldsymbol{u}$ and $\boldsymbol{v}$ to be positive such that $\boldsymbol{C}^T \text{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\} \boldsymbol{S}\text{diag}\left\{\boldsymbol{v}\right\}^{-1} \boldsymbol{S}^T \text{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\} \boldsymbol{C}$ is a symmetric positive definite (SPD) matrix. Therefore, we can calculate $\boldsymbol{M}$ with the Cholesky decomposition:

$$\boldsymbol{C}^T \text{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\} \boldsymbol{S}\text{diag}\left\{\boldsymbol{v}\right\}^{-1} \boldsymbol{S}^T \text{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\} \boldsymbol{C} = \boldsymbol{G}^T\boldsymbol{G}, \tag{3.8}$$

where $\boldsymbol{G}$ is an upper triangular matrix with positive diagonal entries. Since $\tilde{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{M}$, we can choose $\boldsymbol{M} = \boldsymbol{G}^{-1}$ to transform this part into identity. From Expression (3.7), we see that the preconditioned Hessian, $\tilde{\boldsymbol{H}}$, is dependent on a Kronecker product of two parts and the first part has been transformed into an identity. In particular, since the condition number of this part is typically significantly greater than 1, the condition number of the preconditioned Hessian $\tilde{\boldsymbol{H}}$ is significantly smaller than the original Hessian $\boldsymbol{H}$.

After we have obtained the matrix $\boldsymbol{M}$, we can analyze the effect of preconditioning using the SVD. Without preconditioning, the Hessian matrix $\boldsymbol{H}$ depends on two parts, $\boldsymbol{C}^T \text{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\} \boldsymbol{S}\text{diag}\left\{\boldsymbol{v}\right\}^{-1} \boldsymbol{S}^T \text{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\} \boldsymbol{C}$ and $\boldsymbol{A}^T \text{diag}\left\{\boldsymbol{u}\right\} \boldsymbol{A}$. If we assume the singular value decomposition for these two matrices are $\boldsymbol{U}_1\boldsymbol{\Sigma}_1\boldsymbol{V}_1^T$ and $\boldsymbol{U}_2\boldsymbol{\Sigma}_2\boldsymbol{V}_2^T$, then the condition number of the original Hessian $\boldsymbol{H}$ is closely related to $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$. Let the largest and smallest singular values of $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ be σ_{1max}, σ_{1min}, σ_{2max} and σ_{2min}, respectively, then the condition number of the original Hessian, $\kappa\left(\boldsymbol{H}\right)$, can be estimated as

$$\kappa\left(\boldsymbol{H}\right) = \frac{\sigma_{1max}\sigma_{2max}}{\sigma_{1min}\sigma_{2min}}. \tag{3.9}$$

On the other hand, the condition number of the preconditioned Hessian can be approximated by

$$\kappa(\tilde{\boldsymbol{H}}) = \frac{\sigma_{2max}}{\sigma_{2min}}. \tag{3.10}$$

Since the fraction $\sigma_{1max}/\sigma_{1min}$ is most likely to be significantly greater than 1, the condition number of $\tilde{\boldsymbol{H}}$ is likely to be much smaller than $\boldsymbol{H}$. To validate this phenomenon, we can build a numerical example to compare the condition numbers. For an object that is composed of two materials and each material map is of the size 16×16, we can construct the original Hessian $\boldsymbol{H}$ and the preconditioned Hessian $\tilde{\boldsymbol{H}}$ explicitly and compute the estimations of condition numbers for these two Hessian matrices. The result is presented in Table 3.1. From Table 3.1, we can see that the

Matrix Types	Condition Numbers
Original Hessian	2.00 e+06
Preconditioned Hessian	2.59 e+04

Table 3.1
Comparison of Condition Numbers

difference between $\kappa(\boldsymbol{H})$ and $\kappa(\tilde{\boldsymbol{H}})$ is around two orders of magnitude, which indicates the significance of this preconditioner. For a linear system that involves the preconditioned Hessian $\tilde{\boldsymbol{H}}$, the convergence rate is highly dependent on the condition number. With a better-conditioned system, we can compute the solution in a more efficient way. Moreover, we will validate the strength of this preconditioner by solving the preconditioned system versus the original system. More details are presented in Section 5.

3.2. Regularization. With the help of our preconditioner, we can speed up an optimization algorithm and achieve higher accuracy. To further alleviate the noise amplification, it is important to add regularization terms to the objective function. In total, we have m materials and the weights of these m materials are not equal. Rather than adding a single regularization to all weights, we should add a specific regularization to each material. In addition, for different materials, we can choose distinct regularizations to match their properties. For the dominant material, we select the generalized Tikhonov regularization to smooth the edges. For other materials, we choose the ℓ_1 regularization to penalize the sum of weights. Based on this idea, we can represent the regularization term as a sum of m parts:

$$R\left(\boldsymbol{w}\right) = \sum_{i=1}^{m} \frac{\alpha_i}{2} R_i\left(\boldsymbol{w}_i\right), \tag{3.11}$$

where $\boldsymbol{w}_i$ is the vectorization form of the i-th weight matrix, $R_i\left(\boldsymbol{w}_i\right)$ is the corresponding regularization term and α_i is the regularization parameter.

The choice of what type of regularization to use is problem-specific, and *a priori* knowledge of the object being imaged could inform this decision. For example, if it is known that the object contains two material maps with relatively equal distributions, we might select two generalized Tikhonov regularizations. In breast imaging, if the object is dominated by glandular and adipose tissue, it might make sense to use a generalized Tikhonov regularization for each of them. On the other hand, it could be the case that the object is dominated by one material (or one set of materials), with a relatively sparse distribution of another material. In the breast imaging situation, the object may contain small micro-calcifications or areas highlighted by an iodine tracer. In this case, one can use generalized Tikhonov regularizations for the dominating materials (e.g., glandular and adipose tissue) and an ℓ_1 regularization for the sparse material. We illustrate this with two materials, one that dominates, and one that is sparse:

$$R\left(\boldsymbol{w}\right) = \frac{\alpha_1}{2}\|\boldsymbol{L}\boldsymbol{w}_1\|_2^2 + \frac{\alpha_2}{2}\|\boldsymbol{w}_2\|_1. \tag{3.12}$$

If we add these regularization terms to the objective function in Equation (2.20), we can rewrite it as an augumented system:

$$\begin{aligned} \min_{\tilde{\boldsymbol{w}}} \quad & \left\| \begin{bmatrix} \frac{\sqrt{2}}{2}\boldsymbol{\Sigma}^{-\frac{1}{2}}\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) \\ \sqrt{\frac{\alpha_1}{2}}\tilde{\boldsymbol{L}} \end{bmatrix} \tilde{\boldsymbol{w}} - \begin{bmatrix} \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2 + \frac{\alpha_2}{2}\begin{bmatrix}\boldsymbol{0} & \boldsymbol{1}\end{bmatrix}\left(\boldsymbol{M}\otimes\boldsymbol{I}\right)\tilde{\boldsymbol{w}} \\ \text{subject to} \quad & \left(\boldsymbol{M}\otimes\boldsymbol{I}\right)\tilde{\boldsymbol{w}} \geqslant \boldsymbol{0}, \end{aligned} \tag{3.13}$$

where $\tilde{\boldsymbol{L}} = \begin{bmatrix}\boldsymbol{L} & \boldsymbol{0}\end{bmatrix}\left(\boldsymbol{M}\otimes\boldsymbol{I}\right)$. As we can see, the objective function in this problem consists of two parts: one is smooth and convex and the other one is possibly non-smooth. Because of these properties, we can think about using FISTA [2] to solve this problem. It not only fits the features of the objective function but also provides an optimal convergence rate. In addition, we are concerned about the linear inequality constraints, and in each step, we can maintain these constraints by solving a projection problem that is based on the 2-norm.

4. FISTA and Projections.

In this section, we first briefly present the main algorithm FISTA. To implement FISTA to solve the target optimization problem, we need to determine the step size and handle the nonnegative constraints. For the step size, we introduce how to compute the Lipschitz constant numerically and then choose a constant step size based on the calculated Lipschitz constant. For the nonnegative constraints, we build another quadratic programming problem and solve it with a delicate decomposition and efficient algorithms.

4.1. FISTA.

Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) is a first order method that belongs to the family of Iterative Shrinkage-Thresholding Algorithm (ISTA). This method is proposed by Beck et al., and compared with the $\mathcal{O}\left(1/k\right)$ rate of convergence of ISTA, it has a best function value convergence rate $\mathcal{O}\left(1/k^2\right)$, where k is the number of iterations. Moreover, it is very appropriate for problems in imaging science because it is usually used to solve the nonsmooth convex problem

$$\min_{\boldsymbol{x}} \quad f\left(\boldsymbol{x}\right) + g\left(\boldsymbol{x}\right), \tag{4.1}$$

where $f\left(\boldsymbol{x}\right)$ and $g\left(\boldsymbol{x}\right)$ are both convex functions and $g\left(\boldsymbol{x}\right)$ might not be smooth. In imaging sciences, $f\left(\boldsymbol{x}\right)$ is likely to be a least squares loss function to test the goodness of fit and $g\left(\boldsymbol{x}\right)$ can be a regularization term such as a ℓ_1 penalty or a total variation regularization. For Problem (3.13), we construct an augumented loss function that merges the generalized Tikhonov regularization term, which corresponds to $f\left(\boldsymbol{x}\right)$ in (4.1). For the regularization term, the ℓ_1 regularization is nonsmooth but convex and this matches $g\left(\boldsymbol{x}\right)$ in (4.1).

The details of this algorithm are shown in Algorithm (4.1). For the main algorithm, we need to compute the smallest Lipschitz constant K at first. Then we can update the current step using FISTA. Because of the linear inequality constraints, we need to project the new step onto these constraints to keep the solution feasible. We would like to implement FISTA with a constant step size to solve the optimization problem (3.13). To implement this method, we need several preparations, which we will discuss in the following sections.

Algorithm 4.1 FISTA and Projections [2]

```
Initialization:
Calculate the smallest Lipschitz constant $K$ in (4.3) by Power Method.
Set up the initial guess $\tilde{\boldsymbol{W}}_0$; Let $\boldsymbol{y}_0 = \mathrm{vec}\left(\tilde{\boldsymbol{W}}_0\right)$, $\boldsymbol{x}_{old} = \boldsymbol{y}_0$ and $t_1 = 1$;
for $k = 1,\ 2,\ \cdots$ do
    Calculate the gradients, $\nabla f\left(\boldsymbol{y}_k\right)$ and $\nabla g\left(\boldsymbol{y}_k\right)$, of $f\left(\boldsymbol{y}_k\right)$ and $g\left(\boldsymbol{y}_k\right)$ in (4.2);
    $\boldsymbol{x}_k = \boldsymbol{y}_k - \frac{1}{L(f)}\left[\nabla f\left(\boldsymbol{y}_k\right) + \nabla g\left(\boldsymbol{y}_k\right)\right]$;
    Reshape $\boldsymbol{x}_k$ into a matrix and use CVXGEN to solve the projection problems to obtain $\boldsymbol{x}_{new}$ as (4.6);
    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$;
    $\boldsymbol{y}_{k+1} = \boldsymbol{x}_{new} + \left(\frac{t_k - 1}{t_{k+1}}\right)\left(\boldsymbol{x}_{new} - \boldsymbol{x}_{old}\right)$;
    $\boldsymbol{x}_{old} = \boldsymbol{x}_{new}$.
```

4.2. Lipschitz Constant. The first step is to calculate the smallest Lipschitz constant. If we let

$$\begin{aligned} f\left(\tilde{\boldsymbol{w}}\right) &= \left\| \begin{bmatrix} \frac{\sqrt{2}}{2}\boldsymbol{\Sigma}^{-\frac{1}{2}}\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) \\ \sqrt{\frac{\alpha_1}{2}}\tilde{\boldsymbol{L}} \end{bmatrix} \tilde{\boldsymbol{w}} - \begin{bmatrix} \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2, \\ g\left(\tilde{\boldsymbol{w}}\right) &= \frac{\alpha_2}{2}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{1} \end{bmatrix}\left(\boldsymbol{M} \otimes \boldsymbol{I}\right)\tilde{\boldsymbol{w}}, \end{aligned} \tag{4.2}$$

then we need the smallest Lipschitz constant K for $\nabla f\left(\tilde{\boldsymbol{w}}\right)$, which is the largest eigenvalue for $\nabla^2 f\left(\tilde{\boldsymbol{w}}\right)$. That is to say,

$$K = \lambda_{max}\left[\left(\tilde{\boldsymbol{C}}^T \otimes \boldsymbol{A}^T\right)\boldsymbol{\Sigma}^{-1}\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) + \alpha_1 \tilde{\boldsymbol{L}}^T\tilde{\boldsymbol{L}}\right]. \tag{4.3}$$

Since we only need the largest eigenvalue, it is not necessary for us to construct these matrices explicitly; instead we can use an iterative approach, such as the power method [6]. Note that we only need to calculate K once for all FISTA iterations. The details are shown in Algorithm (4.2).

Algorithm 4.2 Power Method [6]

```
*Initialization*:
Generate a random vector $\boldsymbol{q}_0$ and normalize $\boldsymbol{q}_0$;
**for** $i = 1,\ 2,\ \cdots$ **do**
    $\boldsymbol{z}_i = \left[\left(\tilde{\boldsymbol{C}}^T \otimes \boldsymbol{A}^T\right)\boldsymbol{\Sigma}^{-1}\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) + \alpha_1 \tilde{\boldsymbol{L}}^T\tilde{\boldsymbol{L}}\right]\boldsymbol{q}_{i-1}$;
    $\boldsymbol{q}_i = \boldsymbol{z}_i / \left\|\boldsymbol{z}_i\right\|_2$;
    $\lambda_i = \boldsymbol{q}_i^T\left[\left(\tilde{\boldsymbol{C}}^T \otimes \boldsymbol{A}^T\right)\boldsymbol{\Sigma}^{-1}\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) + \alpha_1 \tilde{\boldsymbol{L}}^T\tilde{\boldsymbol{L}}\right]\boldsymbol{q}_i$;
```

4.3. Projections. In addition to the largest eigenvalue, we also need to handle the linear inequality constraints $\left(\boldsymbol{M} \otimes \boldsymbol{I}\right)\tilde{\boldsymbol{w}} \geqslant \boldsymbol{0}$. Generally speaking, we can regard Problem (3.13) as a quadratic programming problem under these specific constraints. To impose the linear inequality constraints, we can construct another quadratic programming problem that offers a nearest solution to satisfy these constraints. If we assume that we have obtained $\tilde{\boldsymbol{w}}_k$ in the k-th step, then we build a projection problem of the form:

$$\begin{aligned} &\min_{\tilde{\boldsymbol{w}}_{new}} && \left\|\tilde{\boldsymbol{w}}_{new} - \tilde{\boldsymbol{w}}_k\right\|_2^2 \\ &\text{subject to} && \left(\boldsymbol{M} \otimes \boldsymbol{I}\right)\tilde{\boldsymbol{w}}_{new} \geqslant \boldsymbol{0}. \end{aligned} \tag{4.4}$$

For small and medium size problems, we can solve it efficiently by direct implementation of standard optimization algorithms. For example, we can use CVX [7, 8] to solve Problem (4.4), which turns to be low-cost both in storage and calculation consumptions. However, there are challenges for large-scale problems. For example, saving long vectors or constructing sparse matrices might require large storage space. Therefore, we should find a method to decompose Problem (4.4) into small pieces and try to solve each small problem accurately and efficiently.

Suppose we reshape vectors into matrices, for example using MATLAB's "reshape" function, $\tilde{\boldsymbol{W}}_{new} = \text{reshape}\left(\tilde{\boldsymbol{w}}_{new}, N_v, N_m\right)$ and $\tilde{\boldsymbol{W}}_k = \text{reshape}\left(\tilde{\boldsymbol{w}}_k, N_v, N_m\right)$, then by Kronecker product properties and the connection between the 2-norm and the Frobenius norm, Problem (4.4) is equivalent to

$$\begin{aligned} &\min_{\tilde{\boldsymbol{W}}_{new}} && \left\| \tilde{\boldsymbol{W}}_{new} - \tilde{\boldsymbol{W}}_k \right\|_F^2 \\ &\text{subject to} && \tilde{\boldsymbol{W}}_{new} \boldsymbol{M}^T \geqslant \boldsymbol{0}. \end{aligned} \tag{4.5}$$

If we focus on each row of $\tilde{\boldsymbol{W}}_k$, $\tilde{\boldsymbol{W}}_k\left(i,\ :\right)$, then Problem (4.5) can be rewritten as

$$\begin{aligned} &\min_{\tilde{\boldsymbol{W}}_{new}} && \sum_{k=1}^{N_v} \left\| \tilde{\boldsymbol{W}}_{new}\left(i,\ :\right) - \tilde{\boldsymbol{W}}_k\left(i,\ :\right) \right\|_2^2 \\ &\text{subject to} && \tilde{\boldsymbol{W}}_{new}\left(i,\ :\right) \boldsymbol{M}^T \geqslant \boldsymbol{0}, \end{aligned} \tag{4.6}$$

where $\tilde{\boldsymbol{W}}_{new}\left(i,\ :\right)$ is the corresponding i-th row in $\tilde{\boldsymbol{W}}_{new}$. It is obvious that this problem is separable, and the original problem (4.5) can be separated into small-sized problems that only involve each row of $\tilde{\boldsymbol{W}}_{new}$ and $\tilde{\boldsymbol{W}}_k$. Since each row only depends on the number of materials N_m, then the size of each problem is usually 2×1 or 3×1. In this case, we can solve each small-sized problem efficiently and concatenate the solutions into a large matrix. To realize this idea, we can find a highly efficient solver for small-sized problems and loop around the number of voxels (pixels if 2D) N_v. In this paper, we choose CVXGEN [15, 16, 17, 18] to generate a customized solver for small quadratic programming problems. It is a problem-specific, fast and accurate code generator which can achieve advance performance in particular for small-sized quadratic programming problems. In addition, if computer clusters are available, we can write parallel programming codes, such as MPI or OpenMP, and compute the solution to this projection problem in parallel. The speedup in this case relies on the number of available compute nodes, but clearly there is potential for significant speedup with such an approach.

In conclusion, we can see that this algorithm incorporates the advantages of the power method, FISTA and the fast solver, CVXGEN, for small-sized problems. With the power method, we only need to save the Hessian-vector multiplication rather than the full Hessian, and it is very cheap to compute. Moreover, we can achieve a rapid convergence by FISTA in the main loop. Finally, the projection problem is decomposed into many small pieces and each can be solved by CVXGEN efficiently.

5. Numerical Experiments.

To test the performance of our preconditioner and the main algorithm, we set up a test problem that is composed of two materials, plexiglass and polyvinyl chloride (PVC). The size of each material map is 128×128. The first material map is a circular mask that dominates the object, while the second material map consists of small "spikes" that are scattered randomly inside the circle. The number of "spikes" is chosen to be 50. Outside of the circle, we assume that there exist no weights of the object. These two images are shown in Figure 5.1.

Inside the mask, the darker blue areas for the first material map are mainly located in the upper left and lower right corners, which corresponds to blank points. Other areas inside the circle are represented by heavily weighted yellow and green color. In the second material map, the weights are scattered around the image and only occupy a small part of the area in total. This test problem can be regarded as a simplification of a real life application. For example, in medical imaging for cancer detection, the first material map is similar to a small area of human body or tissue, while the second material map can represent the calcium located inside this area.

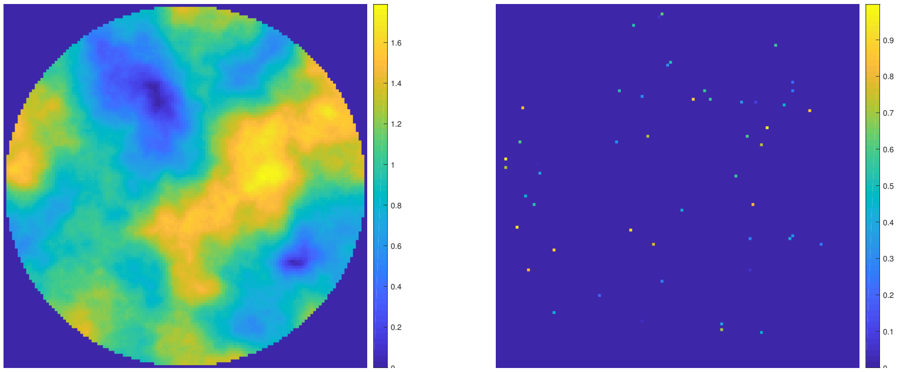

Fig. 5.1. *The original material maps for Plexiglass (Left) and PVC (Right).*

In addition to the test images, we also need other parameters in Equation (1.1). To generate the ray trace matrix $\boldsymbol{A}$, we use the MATLAB function `fanbeamtomolinear` from AIR Tools [13, 10, 9] to simulate a fan-beam geometry with a flat detector. Other parameters that we need to choose in this function are presented in Table 5.1. In addition, we use 180 projections in total which are equally distributed from 0 to 360 degrees. The spectral energy of the x-ray source is generated by the MATLAB function `spektrSpectrum` [22] with 120 keV voltage as input. The detector is assumed to be photon-counting with 5 energy windows. From the first energy window to the fifth energy window, we assume that they can detect the range of photon energies 10 to 34 keV, 35 to 49 keV, 50 to 64 keV, 65 to 79 keV and 80 to 120 keV, respectively.

Items	Parameters (cm)
Width of Domain	2.0
Distance from Source to Rotation Center	3.0
Distance from Source to Detector	5.0
Detector Width	4.0

Table 5.1
Geometry Parameters of CT Machine

The plot of photon flux density versus photon energy is presented in Figure 5.2. In Figure 5.2, the red curve represents photon intensity of x-ray source and the gray boxes indicate energy windows of the detector. Moreover, the black dots are the values of mean photon energy in each energy window. When we build the test problem, the full energy spectrum and all the corresponding linear attenuation coefficients are used, while only the mean photon energies and the corresponding linear attenuation coefficients are applied for reconstruction. As it is well-known, this strategy of generating data on a finer grid and solve it on a coarser grid is a standard approach to avoiding what is called the inverse crime.

We also plot the curves of linear attenuation coefficients with respect to photon energy in Figure 5.3. From Figure 5.3, we can see that the slopes of these two curves are close to each other, which are likely to introduce the collinearity between coefficients. Moreover, we assume that the entries of the matrix $\boldsymbol{Y}$ follow a Poisson distribution, and for large scale problems, from the Central Limit Theorem, the Poisson distribution is approximated well by a Gaussian distribution. So the assumption of Gaussian model is valid.

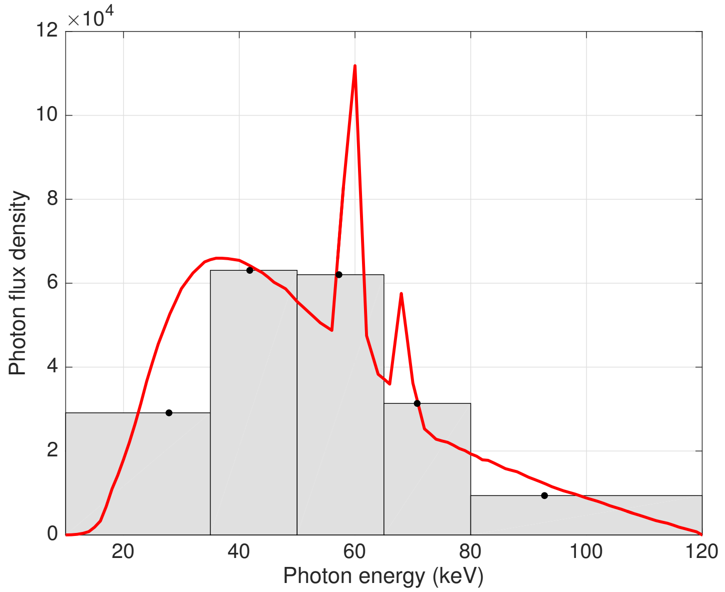

Fig. 5.2. *Detector bins and photon flux density.*

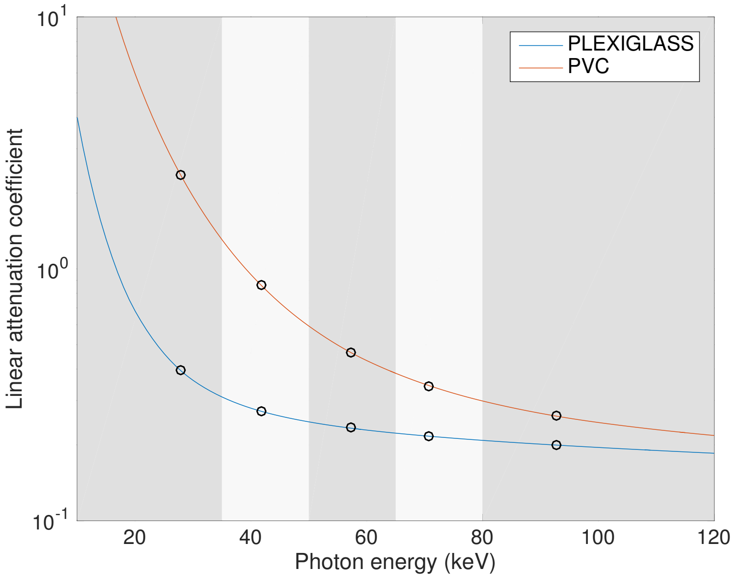

Fig. 5.3. *Linear attenuation coefficients and photon flux density.*

The reconstructed images are shown in Figure 5.4. From Figure 5.4, we can see that we achieve almost perfect separation for these two materials. Moreover, the reconstructed images have excellent quality in terms of visuality. Both two material maps are relatively close to the true images. In the first material map, the distribution of weights is clear to identify. The low intensity pixels are located in the upper left

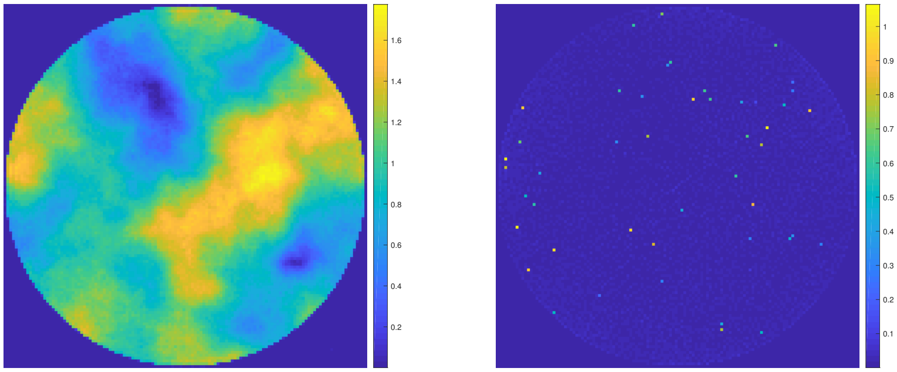

Fig. 5.4. *The reconstructed images for plexiglass (Left) and PVC (Right).*

and lower right areas of the circle, while other places are occupied by the yellow and green colors. Moreover, we can easily recognize the edges of the circle that indicate the boundary of the object, which is a plus. As we can see, the reconstruction of small "spikes" are of great difficulty because of the randomness of weights and spots. However, we can see that the small "spikes" are scattered in the same positions as the true image, while they are masked by the shade of a circle. These results present the significance of methods proposed in this paper.

To further validate the results, we plot the relative errors of these two materials versus the number of FISTA iterations. The decrease of relative errors of corresponding materials is shown in Figure 5.5. From this figure, we can see that the relative

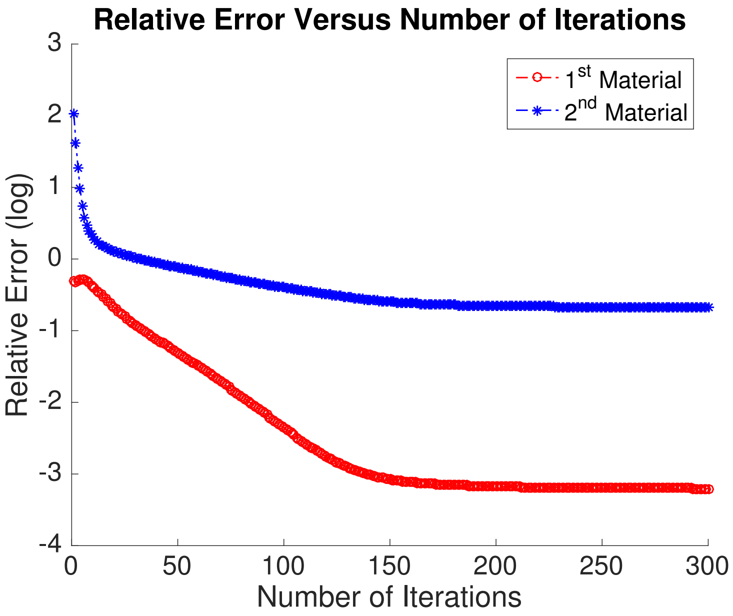

Fig. 5.5. *The related errors for each iteration (with preconditioner) for plexiglass and PVC.*

error of the first material drops sharply as the number of iterations increase. It then stagnates after around 150 iterations. However, the relative error of the second material only decreases fast in the beginning, and after several iterations, the rate of change slows down and the relative error cannot reduce further. We can also identify the same phenomenon by comparing the true and reconstructed images of the second material map. Even if the spots of these "spikes" are approximately correct, the numerical weights of these dots might not be the same. Moreover, there are a large number of small values in the background of the reconstructed image, causing somewhat large relative errors, even though visually the result looks quite good.

Other accuracy measures illustrate this phenomenon. In Figure 5.6, we plot the mean squared error (MSE) at each iteration. In Figure 5.7, the structural similarity index (SSIM) is presented. Not surprisingly MSE produces information very similar to the relative errors, but it also shows a clear diminution for the second material from Figure 5.6. The SSIM is a metric for image quality and large values correspond to better solutions. From Figure 5.7, it can be found that the quality of the reconstructed first material map improves slowly in the early iterations but it achieves a higher quality measure in the end compared with the second material map. In summary, all of these errors and quality measures illustrate fast convergence to high quality reconstructions.

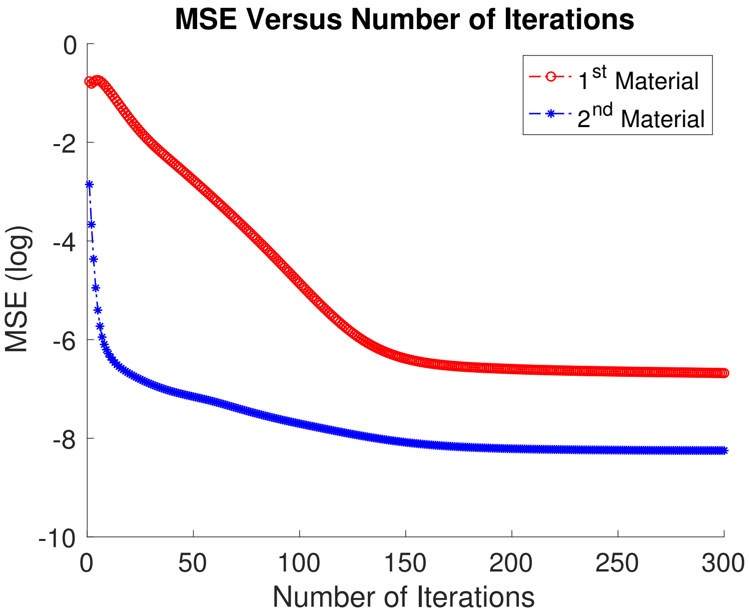

Fig. 5.6. *MSE for each iteration (with preconditioner) for plexiglass and PVC.*

It may also be of interest to observe the decay of norm of the gradient at each iteration, which is shown in Figure 5.8. From this figure, we can see that the norm of the gradient decreases significantly in the beginning and levels off after a sufficient number of iterations, indicating the convergence to a minimizer.

To further validate the strength of our proposed preconditioner, we compare the performance with a preconditioner proposed by Barber [1], and the performance without using any preconditioners. As previously mentioned, the approach proposed in [1] is based on the eigenvalue decomposition of $\boldsymbol{C}^T\boldsymbol{C}$. The results are shown in Figure 5.9, where we plot the decay of relative errors for these three cases. To reduce clutter in this plot, we only show results for the first material; the behavior for the

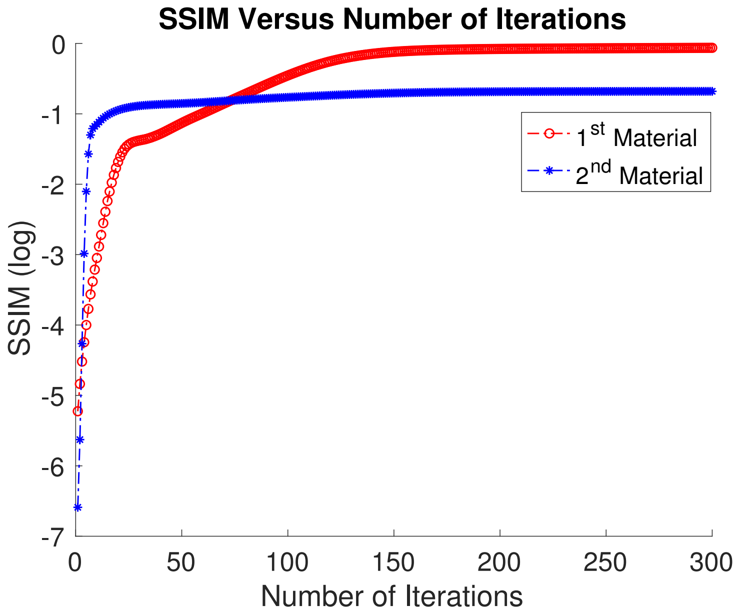

FIG. 5.7. *SSIM for each iteration (with preconditioner) for plexiglass and PVC.*

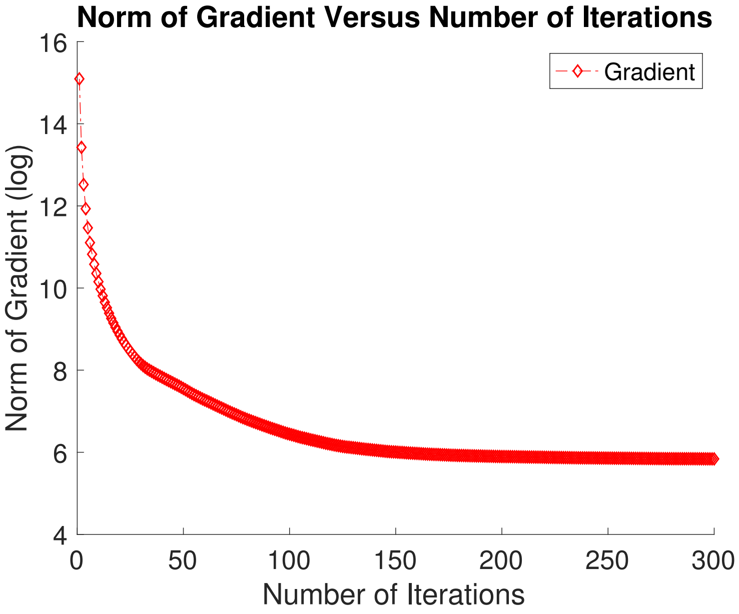

FIG. 5.8. *The norm of the gradient for overall materials, normalized by the 2-norm of the image.*

second material is the same. From this figure, we can easily observe that both preconditioners are effective at accelerating convergence, with our approach producing the fastest convergence and the lowest relative errors.

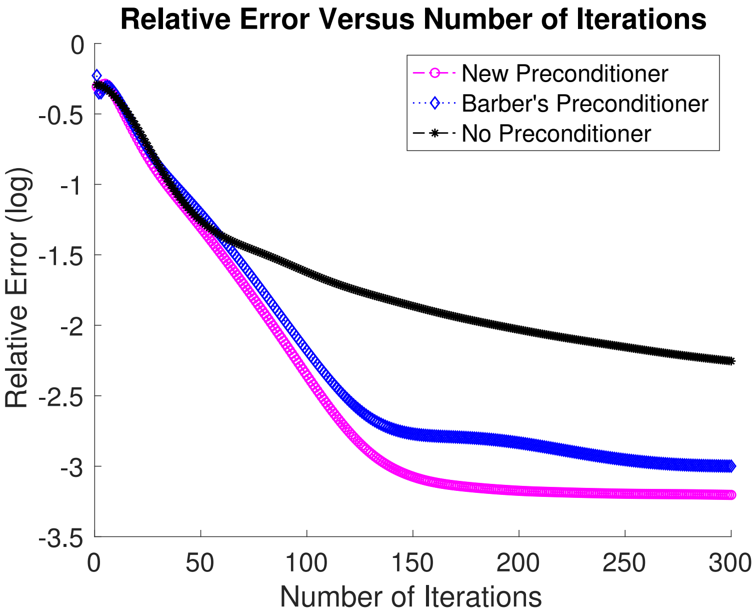

FIG. 5.9. *The decay of related errors with new preconditioner, Barber's [1] preconditioner, and with no preconditioner.*

6. Conclusions and Remarks.

In this paper, we use the Gaussian assumption of noise to construct a weighted least squares problem under bound constraints for energy discriminating x-ray detectors in computed tomography. Based on this problem, we propose a new preconditioner that includes not only the information of the linear attenuation coefficient matrix $\boldsymbol{C}$ but also the projected data matrix $\boldsymbol{Y}$ and the energy spectrum matrix $\boldsymbol{S}$. With this new preconditioner, the condition number of the Hessian can be reduced significantly. To implement this new preconditioner within an optimization framework, we suggest to use a first order method, FISTA, that can generate fast convergence speed. Because of the introduction of the new preconditioner, we recommend to construct a projection problem and compute the nearest step that will satisfy the linear inequality constraints for each iteration. Finally, numerical experiments also specify the advantages of the method mentioned in this paper. For future work, it would be interesting to consider other regularization schemes to emphasize the edges of the object, such as the total variation.

Acknowledgments. This work is supported by U.S. National Science Foundation under grant no. DMS-1819042. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NSF.

REFERENCES

[1] R. F. Barber, E. Y. Sidky, T. G. Schmidt, and X. Pan, *An algorithm for constrained one-step inversion of spectral CT data*, Physics in Medicine and Biology, 61 (2016), p. 3784.
[2] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 183–202.
[3] A. Bjorck, *Numerical Methods for Least Squares Problems*, vol. 51, SIAM, Philadelphia, PA, 1996.
[4] V. M. Bustamante, J. G. Nagy, S. S. Feng, and I. Sechopoulos, *Iterative breast tomosyn-*

thesis image reconstruction, SIAM Journal on Scientific Computing, 35 (2013), pp. S192–S208.

[5] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, Journal of Mathematical Imaging and Vision, 40 (2011), pp. 120–145.

[6] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 3, JHU Press, MD, 2012.

[7] M. Grant, S. Boyd, and Y. Ye, *CVX: Matlab software for disciplined convex programming*, 2008.

[8] M. C. Grant and S. P. Boyd, *Graph implementations for nonsmooth convex programs*, in Recent Advances in Learning and Control, Springer, 2008, pp. 95–110.

[9] P. C. Hansen and J. S. Jørgensen, *AIR Tools II: algebraic iterative reconstruction methods, improved implementation*, Numerical Algorithms, (2017), pp. 1–31.

[10] P. C. Hansen and M. Saxild-Hansen, *AIR Tools: A MATLAB package of algebraic iterative reconstruction methods*, Journal of Computational and Applied Mathematics, 236 (2012), pp. 2167–2178.

[11] B. J. Heismann, B. T. Schmidt, and T. Flohr, *Spectral Computed Tomography*, SPIE Bellingham, WA, 2012.

[12] J. D. Ingle Jr and S. R. Crouch, *Spectrochemical Analysis*, Prentice Hall College Book Division, NJ, 1988.

[13] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, SIAM, Philadelphia, PA, 2001.

[14] R. M. Larsen, *Lanczos bidiagonalization with partial reorthogonalization*, DAIMI Report Series, 27 (1998).

[15] J. Mattingley and S. Boyd, *Automatic code generation for real-time convex optimization*, Convex Optimization in Signal Processing and Communications, (2009), pp. 1–41.

[16] J. Mattingley and S. Boyd, *Real-time convex optimization in signal processing*, IEEE Signal Processing Magazine, 27 (2010), pp. 50–61.

[17] J. Mattingley and S. Boyd, *CVXGEN: A code generator for embedded convex optimization*, Optimization and Engineering, 13 (2012), pp. 1–27.

[18] J. Mattingley, Y. Wang, and S. Boyd, *Code generation for receding horizon control*, in Computer-Aided Control System Design (CACSD), 2010 IEEE International Symposium on, IEEE, 2010, pp. 985–992.

[19] J. L. Mueller and S. Siltanen, *Linear and Nonlinear Inverse Problems with Practical Applications*, SIAM, Philadelphia, PA, 2012.

[20] A. S. Nemirovsky and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*, Chichester: Wiley, 1983.

[21] Y. E. Nesterov, *A method for solving the convex programming problem with convergence rate o (1/kˆ 2)*, in Doklady Akademii Nauk SSSR, vol. 269, 1983, pp. 543–547.

[22] J. H. Siewerdsen, A. M. Waese, D. J. Moseley, S. Richard, and D. A. Jaffray, *Spektr: A computational tool for x-ray spectral analysis and imaging system optimization*, Medical Physics, 31 (2004), pp. 3057–3067.

[23] C. F. Van Loan, *The ubiquitous kronecker product*, Journal of Computational and Applied Mathematics, 123 (2000), pp. 85–100.

[24] V. S. K. Yokhana, B. D. Arhatari, T. E. Gureyev, and B. Abbey, *Soft-tissue differentiation and bone densitometry via energy-discriminating x-ray microct*, Optics Express, 25 (2017), pp. 29328–29341.