Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Yunyi Hu Date

Matrix Computations and Optimization for Spectral Computed Tomography

By

Yunyi Hu

Doctor of Philosophy

Mathematics

James G. Nagy

Advisor

Lars Ruthotto

Committee Member

Yuanzhe Xi

Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.

Dean of the James T. Laney School of Graduate Studies

Date

Matrix Computations and Optimization for Spectral Computed Tomography

By

Yunyi Hu

B.S., Northeastern University (P. R. China), 2014

Advisor: James G. Nagy, Ph.D.

An abstract of

A dissertation submitted to the Faculty of the

James T. Laney School of Graduate Studies of Emory University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in Mathematics

2019

Abstract

Matrix Computations and Optimization for Spectral Computed Tomography

By Yunyi Hu

In the area of image science, 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. In this thesis, we focus on optimization, preconditioning and model development of spectral CT. For simple energy discriminating detectors, a nonlinear optimization framework is built on a Poisson likelihood estimator and bound constraints. A nonlinear interior-point trust region method is implemented to compute the solution. For energy-windowed spectral CT, a nonlinear least squares approach is proposed to describe the problem and under bound constraints, a two-step method using the projected line search and the trust region approach, incorporated with an adaptive preconditioner, is used to solve the problem. In addition, a weighted least squares formulation is derived from the Gaussian noise assumption and another preconditioner that is based on rank-1 approximation is introduced to obtain robust reconstruction. The Fast Iterative Shrinkage-Thresholding Algorithm (FISTA), along with a projection step, is used to calculate the solution iteratively. Compared with a direct solver, a two-step model is developed using an auxiliary variable. With this two-step model, a row-wise computational method is proposed, which further reduces memory requirements and improves solution accuracy. Numerous numerical experiments are conducted to indicate the strength of methods and real-life examples are presented to show possible applications.

Matrix Computations and Optimization for Spectral Computed Tomography

By

Yunyi Hu

B.S., Northeastern University (P. R. China), 2014

Advisor: James G. Nagy, Ph.D.

A dissertation submitted to the Faculty of the

James T. Laney School of Graduate Studies of Emory University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in Mathematics

2019

Acknowledgments

I would like to express my deep gratitude to my advisor, Professor James G. Nagy. In academia, Jim helped me set up my current projects, taught me the necessary skills, introduced me to other great researchers, inspired me with new ideas and encouraged me when I met obstructions. Without him, I really doubt if I could move my research to the current point. He is not only a great coach in academia, but also a great educator in my life. He supported me when I was distracted by daily trivia and gave me priceless advices on my career development. All of these experiences build my maturity and profession, which are significant preparations for my future of life. I feel exceptionally lucky of being a student of him and words cannot fully reveal my gratefulness.

I also want to show my great thankfulness to Professor Martin S. Andersen who offered me the chance to visit Technical University of Denmark. I really enjoyed the life in Copenhagen where I immersed myself in work. Under his supervision, I gained a deeper insight into my research projects and we made great process on proposing new methods. This experience is an invaluable treasure in my life.

In addition, I would like to express appreciation to Professor Lars Ruthotto, Professor Michele Benzi and Professor Yuanzhe Xi, who guided me to explore the charm of mathematics and trained me with their great efforts. Moreover, I am grateful to other faculty and staff in the Department of Mathematics. They are always friendly and approachable when I need any helps.

Finally, I am indebted to my friends and family. I am fortunate to meet my friends who are from all around the world. I always remember the times when we shared our happiness and sadness with each other. I also want to express special thanks to my parents. Thank you for offering me the entire freedom and allowing me to choose whatever I like.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

An active area of interest in tomographic imaging is *quantitative imaging*, where in addition to producing an image, information about the material composition of the object is recovered. In order to obtain the information of material composition, it is necessary to better model of the image formation (i.e., forward) problem and/or to collect additional independent measurements. In x-ray computed tomography (CT), better modeling of the physics can be done by using the more accurate polyenergetic representation of source x-ray beams. In addition, recent advances in engineering have produced detectors that are made up of several energy windows and each energy window is assumed to detect a specific range of energy spectra. With this technique, a nonlinear matrix equation is formulated to represent the discretized process of attenuation of x-ray intensity

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

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 par-

ticular (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 these data are known and the target is to solve the unknown weight matrix $\boldsymbol{W}$. $\boldsymbol{W}$ real and is of 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. The derivation of this equation is not straightforward and we will go into details later.

1.1 Contributions of Work

For different noise distributions, model (1.1) can be transformed into various formats and the methods used to compute solutions are diversified. If we assume that we only have one energy window, then the matrix equation (1.1) is reduced to a nonlinear system of the form

$$\boldsymbol{y} = \exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{s} + \boldsymbol{\eta}, \tag{1.2}$$

where $\boldsymbol{y}$ is the vector of projected data, $\boldsymbol{s}$ collects the information of energy spectrum and $\boldsymbol{\eta}$ represents the noise vector. For this problem, we transform it into a nonlinear optimization problem that involves both equality and inequality constraints. In order to keep the second order derivative positive semi-definite, we propose a modified Hessian. In addition to the modified Hessian, a problem-specific nonlinear interior-point trust region method is implemented to solve this problem. Moreover, total variation regularization is applied to stabilize the solutions. Both for full CT and limited angle cases, we can obtain images of high quality with this method.

If we take the assumption of multiple energy windows and each energy window can only detect a specific range of energy, then the basic model is the same as (1.1). Under the constraint of nonnegative weights, we can build a nonlinear optimization

problem as

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2}\left\|\boldsymbol{y}-\left(\boldsymbol{S}^{T}\otimes\boldsymbol{I}\right)\exp\left\{-\left(\boldsymbol{C}\otimes\boldsymbol{A}\right)\boldsymbol{w}\right\}\right\|_{2}^{2} \\ \text{subject to} \quad & \boldsymbol{w}\geqslant\boldsymbol{0}. \end{aligned} \tag{1.3}$$

In (1.3), $\otimes$ represents the Kronecker product and the exponential operator is applied point-wise. $\boldsymbol{w}\geqslant\boldsymbol{0}$ indicates that each entry in $\boldsymbol{w}$ should be bounded blow by zero. Moreover, $\boldsymbol{y}$ and $\boldsymbol{w}$ are vectorization of $\boldsymbol{Y}$ and $\boldsymbol{W}$, respectively. For problem (1.3), we formulate it as a nonlinear least squares problem, which is solved by a Gauss-Newton scheme. We show that if the object contains a mixture of materials with one known to be sparsely represented, then a combination of generalized Tikhonov and ℓ_1 regularization can be very effective in producing high quality quantitative reconstructions. Because the approximate Hessian system in the Gauss-Newton scheme is very ill-conditioned, we propose a preconditioner that effectively clusters eigenvalues and, therefore, accelerates convergence when the conjugate gradient method is used to solve the linear subsystems. To implement the preconditioner, a two-step method is used to guarantee faster convergence and to provide further stabilization. In particular, in the first step we compute the approximate Cauchy point, and then in the second step we set up and solve a quadratic programming problem. Numerical experiments illustrate the convergence, effectiveness, and significance of the proposed method.

To avoid solving a nonlinear optimization problem, we might consider taking advantage of the Gaussian noise assumption and transform it into a weighted least squares problem. In this case, it is necessary to assume that $\boldsymbol{S}$ is square and invertible. Moreover, for the noise term $\boldsymbol{\mathcal{E}}$, we assume that $E_{il}\sim\mathcal{N}\left(0,y_{il}\right)$ for each component E_{il} in $\boldsymbol{\mathcal{E}}$ and y_{il} in $\boldsymbol{Y}$. A linearization technique is used to transform the nonlinear equation (1.1) into an optimization problem that is based on a weighted

least squares term and a nonnegative bound constraint of the form

$$\begin{aligned} \min_{\tilde{\boldsymbol{w}}} \quad & \frac{1}{2}\|\mathcal{A}\tilde{\boldsymbol{w}} - \boldsymbol{b}\|^2_{\boldsymbol{\Sigma}^{-1}} \\ \text{subject to} \quad & (\boldsymbol{M} \otimes \boldsymbol{I})\,\tilde{\boldsymbol{w}} \geqslant \boldsymbol{0}, \end{aligned} \tag{1.4}$$

where $\tilde{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{M}^{-T}$, $\tilde{\boldsymbol{w}} = \mathrm{vec}(\tilde{\boldsymbol{W}})$, $\tilde{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{M}$, $\mathcal{A} = \tilde{\boldsymbol{C}} \otimes \boldsymbol{A}$, $\tilde{\boldsymbol{y}} = \left(\boldsymbol{S}^{-T} \otimes \boldsymbol{I}\right)\boldsymbol{y}$ and $\boldsymbol{b} = -\log(\tilde{\boldsymbol{y}})$. The matrix $\boldsymbol{M}$ represents the corresponding preconditioner and $\|\cdot\|^2_{\boldsymbol{\Sigma}^{-1}}$ is a weighted least squares term such that $\|\mathcal{A}\tilde{\boldsymbol{w}} - \boldsymbol{b}\|^2_{\boldsymbol{\Sigma}^{-1}} = (\mathcal{A}\tilde{\boldsymbol{w}} - \boldsymbol{b})^T \boldsymbol{\Sigma}^{-1} (\mathcal{A}\tilde{\boldsymbol{w}} - \boldsymbol{b})$. To solve this optimization problem, we want to use a first order method to obtain fast and robust reconstruction. However, because of the ill-posedness, direct implementation of optimization methods does not offer us satisfactory results. Therefore, we propose a new Hessian preconditioner in order to 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 preconditioner, a linear inequality constraint is also added. In each iteration, we decompose this constraint into small-sized problems that can be solved with fast optimization solvers. Furthermore, we conduct numerical experiments to indicate strengths of the proposed method and potential improvements.

Rather than solving $\boldsymbol{W}$ directly in one step, we can use an auxiliary variable, $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{W}$, and construct a two-step framework as

$$\begin{aligned} \boldsymbol{Y} &= \exp\left(-\boldsymbol{X}\boldsymbol{C}^T\right)\boldsymbol{S} + \boldsymbol{E}, \\ \boldsymbol{X} &= \boldsymbol{A}\boldsymbol{W}. \end{aligned} \tag{1.5}$$

For the first step, we can reuse the Gaussian assumption of noise and build a row-wise

moldel as

$$(\boldsymbol{x}_i)_{\mathrm{ml}} = \underset{\boldsymbol{x}_i}{\operatorname{argmin}} \left\{ \left\| \boldsymbol{C}\boldsymbol{x}_i - \hat{\boldsymbol{b}}_i \right\|^2_{\boldsymbol{\Sigma}^{-1}_{\hat{\boldsymbol{b}}_i}} \right\}, \tag{1.6}$$

where $\boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}_i}$ is the noise covariance matrix related to $\hat{\boldsymbol{b}}_i$ and $(\boldsymbol{x}_i)_{\mathrm{ml}}$ is the maximum likelihood estimator to the i-th row of $\boldsymbol{X}$. Even though this model is based on the maximum likelihood estimator, we can compute the solution corresponding to each row slice one at a time, and sum when they all are computed. Each small-sized problem does not depend on others, so these can also be solved in parallel. For the second step, we need to quantify the noise propagation and solve a least squares optimization problem under bound constraints. To solve the corresponding optimization problem, we use FISTA with projections onto the boundary to obtain faster convergence and higher accuracy. Instead of using the two-step method, we can build a coupled optimization problem based on (1.5) and solve $\boldsymbol{W}$ directly. Numerical experiments show that the expense is lower and reconstructed images are of higher quality using the two-step framework compared with the coupled method.

1.2 Outline of Thesis

To reconstruct spectral CT images, we are required to solve a complicated nonlinear equation. Distinct assumptions might provide us with different models and various solutions. This thesis mainly focuses on how to build efficient models and find superior solutions.

The thesis is organized as follows. In Chapter 2, we first review the physical background and basics of computed tomography (CT) and present three classical methods to compute solutions. Since the traditional CT does not take energy spectra into consideration, we introduce the spectral CT model in Chapter 3. We also build a nonlinear optimization problem and implement a nonlinear interior-point trust region method to solve it. In Chapter 4, we switch to the energy-windowed spectral CT

model and build a nonlinear least squares problem based on it. An adaptive preconditioner based on the Gauss-Newton Hessian approximation is also included in Chapter 5. In Chapter 6, we use the assumption that noise is normally distributed to transform the energy-windowed spectral CT model into a weighted least squares problem. Another efficient preconditioner inspired by the interlacing of Kronecker products and diagonal matrices is also presented in Chapter 5. In Chapter 6, we move further to consider a two-step model as well as introduce a solution that is based on row slicing and weighted least squares optimization. As a comparison, the coupled method is also included in Chapter 6. Numerical experiments are conducted to show the strengths of proposed methods and conclusions and comparisons are drawn in the end of each chapter.

Chapter 2

X-ray Computed Tomography

In 1971 Godfrey Hounsfield [4] opened a new window for the world with his invention of the computed tomography (CT) machine. CT machines project x-ray beams from a known source through an object, which are then received by detectors. The energy of the source x-ray beams are known, and the detectors measure the energy after they pass through the object. The energy difference between the source and the detector depends on the *attenuation* properties of the object, and is modeled by the Beer's law [34]. If enough measurements are obtained, then by solving an inverse problem arising from the Beer's law, the inner structure of an object can be reconstructed. Because the Beer's law results in a very challenging nonlinear inverse problem, most image reconstruction algorithms are based on linear approximations. The linear approximations allow for very fast algorithms, and images they produce are often quite good. Thus CT immediately gained popularity for medical diagnostics, but the technique is also used widely in industry to, for example, inspect inanimate objects for defects, or in security to look for weapons and other dangerous materials.

Current CT machines mostly use single-energy tubes to conduct scanning. For single-energy CT, we assume that the x-ray tube only emits a uniform energy and this energy is used to estimate the attenuation properties of the object. Using the

Radon transform [14], a linear system is constructed to represent an approximation to this physical process. Under certain conditions, an image obtained from solving the corresponding linear system is clear enough to identify necessary information. However, the assumption of single uniform energy is only a simplification of the underlying physics. In actuality, the x-ray beams consist of a spectrum of energies, rather than a single energy. Because of this simplification, precise estimation of the attenuation properties of the object is very challenging, and often leads to appearance of so-called beam-hardening artifacts [45].

In 2006, the invention of dual-source CT machines refreshed this field [19]. Using two x-ray tubes with different voltages, two sets of projected data corresponding to two energy spectra are obtained to conduct image reconstruction. Basically, the attenuation properties of the object can be represented as a function of two variables, position and energy. With an extra energy, we can obtain more information about the object. On the one hand, the beam-hardening artifacts generated by single-energy CT can be dramatically reduced using dual-source CT, which offers images of higher quality. On the other hand, not only the inner structure but also the material separation of an object can be acquired, which provides the fundamental basics of quantitative spectral CT methods. Even if the idea of basis-material decomposition (BMD) was proposed by Alvarez and Macovski in 1976 [1], it is suddenly in the spotlight after the emergence of dual-source CT machines. If the contrast of different materials is small, it is hard to differentiate them by investigating the gray images generated by single-energy CT. For example, in breast imaging the object consists mainly of glandular and adipose tissues, which have similar densities and therefore are difficult to separate by traditional techniques. With spectral CT methods, it is likely that we can reconstruct the image of these two materials using different linear attenuation coefficients corresponding to distinct energies.

Nowadays, spectral CT still evolves both in theory and practice. If we assume

that the detector of CT machines has multiple energy windows and further assume each energy window can detect only a specific range of energy, then it is an energy-windowed spectral CT model. This model is more complicated, but on the other side, it can provide us with more flexibility on mathematical theories as well as more quantitative information of images. In practice, one step forward from dual-source CT machines, there are multi-energy CT machines that explore improvements over more than two sources. However, the real-life application of multi-energy CT is limited by the overlap of various energy spectra and this limitation might be overcame by photon counting detectors. On the one hand, we can obtain the material decomposition of an object and higher quality images with spectral CT. On the other hand, we want to lower the radiation dose to reduce risks of causing other health issues. To reach these goals, spectral CT is still an active research area and more promising results might show up in the near future.

2.1 Physical Background

In computed tomography, x-ray beams are emitted (usually in a cone) from a source at known energies and are directed to pass through an object under investigation, after which the remaining energy of the x-ray beams are measured at a detector. See Figure 2.1 for a 2D illustration. The amount of energy lost as the x-ray beams pass through the object is referred to as attenuation. The amount of attenuation depends on the energy of the x-ray beams, and on the material through which it penetrates; low dose energies are more easily attenuated, and denser materials have higher attenuation properties. The detector is typically partitioned into a grid of bins; in a full 3D model, these are often loosely referred to as detector pixels. If the 3D object is discretized into a grid of small volume elements (called voxels), then each voxel can be associated with a particular attenuation value, referred to as an

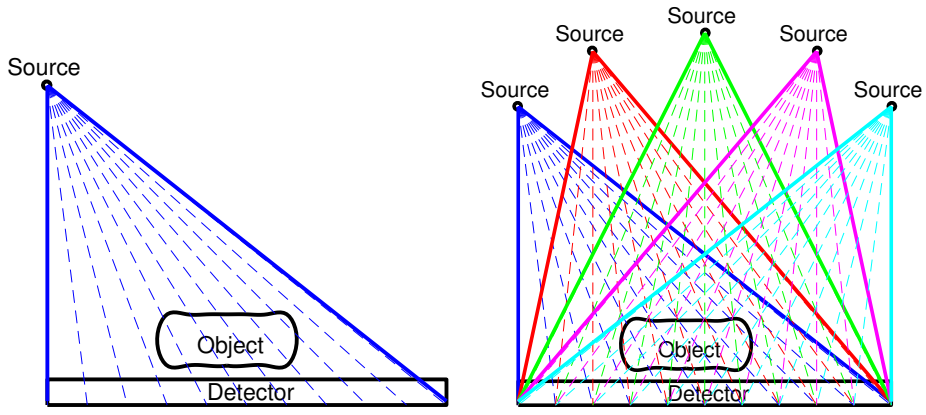

Figure 2.1: Illustration of a 2D CT imaging setup. The left illustrates how x-ray beams are emitted from the source in a cone, pass through an object of interest, and are then measured at the detector. The right illustrates how the source might be rotated around the object to collect additional data. In this illustration we assume the detector remains stationary, which is often the case in limited angle tomography applications such as tomosynthesis, but it should be obvious how the illustration would be modified if the detector rotates with the source.

attenuation coefficient. The problem of CT image reconstruction is to determine these attenuation coefficients from a sequence of measured projection data, which is obtained by rotating the source (at least partially) around the object; again, we refer to Figure 2.1 for a 2D illustration of the data collection process.

2.2 The Beer-Lambert's Law

As shown in Figure (2.1), the change of intensity before and after illumination can be described by the Beer's law or the Beer-Lambert's law [5, 38]:

$$\frac{\mathrm{d}\, I}{\mathrm{d}\, x} = -\mu(x) I, \tag{2.1}$$

where I represents the intensity and $\mu(x)$ is the linear attenuation coefficient at x. It shows that the change of intensity at a specific location equals to the product of the intensity at that location and the corresponding linear attenuation coefficient. The

physical meaning explained by Equation (2.1) is shown in Figure (2.2). In Figure (2.2), an object is located in the center and a x-ray beam with intensity I_0 penetrates it and the detector captures this x-ray beam with intensity I_1. $I_1 < I_0$ and the attenuation characteristic of the object causes the reduction of intensity and this change is also captured by Equation (2.1). For non-homogeneous material $\mu(x)$, the

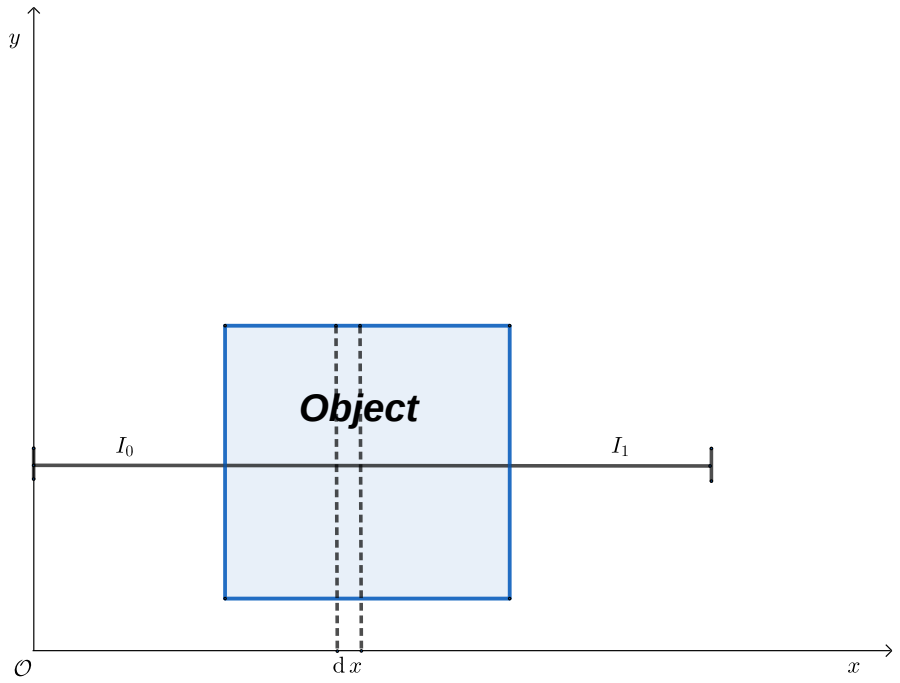

Figure 2.2: Explanation of the Beer-Lambert's law. The blue brick represents the object. There is a x-ray beam illuminating the object with the itensity I_0 and it is received by the detector with itensity I_1.

ordinary differential equation (2.1) has an analytical solution

$$I = I_0 \exp\left\{-\int_l \mu(x)\,\mathrm{d}\,x\right\}. \tag{2.2}$$

As long as we know I_0 and I_1, we can try to compute $\mu(x)$ using mathematical techniques. Basically, we have three mainstream methods to solve it: the analytical reconstruction using the Radon transform and the filtered back-projection, Algebraic Reconstruction Technique (ART) and statistical reconstruction methods.

Basically, Equation (2.1) has two underlying assumptions:

- x-ray beams pass through an object and they are not bent or diffracted by this object. They travel along a straight line.
- x-ray beams are monochromatic, which means they are of only one single energy.

The first assumption is realistic because x-rays are of high energy and thus of short wavelength. In most cases, the diffraction effect can be ignored and it is a reasonable approximation. However, the second assumption is not realistic, because in reality x-rays are made up of a spectrum of energies rather than a single energy. A more comprehensive equation might be written as

$$I = \int_E S(e) \exp\left\{ - \int_l \mu(x) \, \mathrm{d}\, x \right\} \mathrm{d}\, e, \tag{2.3}$$

where $S(e)$ is the energy intensity at energy level e and E is the energy spectrum. Compared with Equation (2.2), Equation (2.3) has two integrals and the nonlinearity might cause more difficulties to compute the solution. In this chapter, we will focus on Equation (2.2) in most cases and discuss the classical methods to find solutions.

2.3 Classical Reconstruction Methods

2.3.1 The Radon Transform and the Inverse Radon Transform

The Radon transform was introduced by Johann Radon [50] in 1971. If we assume θ is expressed as $\theta = (\cos\theta, \sin\theta)^T$, then the Radon transform is an integral transform such that

$$\mathcal{R}\mu(s, \theta) = \int_{l_{s,\theta}} \mu(x) \, \mathrm{d}\, x, \tag{2.4}$$

where $l_{s,\theta}$ is a signed line represented by the signed orthogonal distance s and the angle θ. $l_{s,\theta}$ can be represented as an equation

$$l_{s,\theta} = \{(x, y) | x \cos\theta + y \sin\theta = s\} . \tag{2.5}$$

The Radon transform is illustrated in Figure 2.3, where $l_{s,\theta}$ is drawn perpendicular

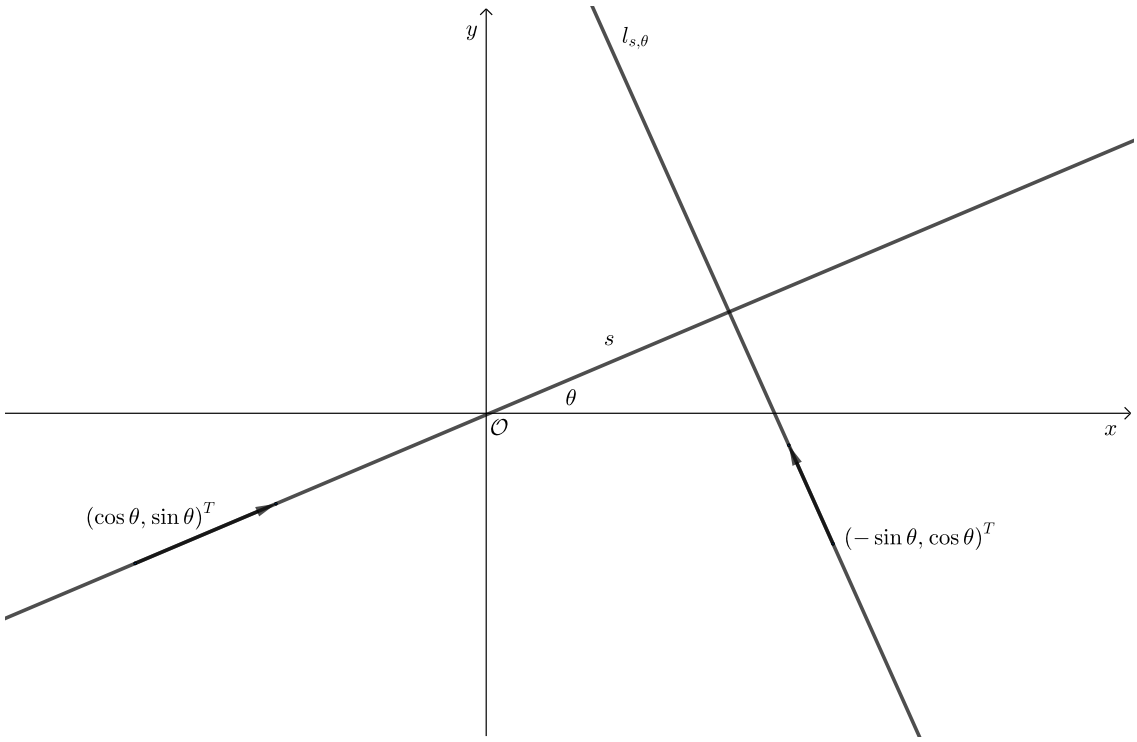

Figure 2.3: Explanation of the Radon transform. The line $l_{s,\theta}$ is perpendicular to the ray passing through the origin. The distance between these two lines is s.

to the original line.

Recall that from the Beer's law, the intensity I can be represented on behalf of s and θ, $I_{s,\theta}$. Then Equation (2.2) is equivalent to

$$I_{s,\theta} = I_0 \exp\left\{ - \int_{l_{s,\theta}} \mu(x) \, \mathrm{d}\, x \right\} . \tag{2.6}$$

If we move all intensities in Equation (2.6) into one side and take a logarithm, then we can obtain the Radon transform. So the connection between the Beer-Lambert's

law and the Radon transform is

$$\mathcal{R}\mu(s,\theta) = \int_{l_{s,\theta}} \mu(x)\,\mathrm{d}\,l = -\log\left(\frac{I_{s,\theta}}{I_0}\right). \tag{2.7}$$

Moreover, the Radon and Fourier transforms have a strong connection as

$$\widetilde{\mathcal{R}\mu}(r,\theta) = \int_{-\infty}^{\infty} \mathcal{R}\mu(s,\theta)e^{-irs}ds = \hat{\mu}(r\theta). \tag{2.8}$$

If we represent x in a polar coordinate form as $x = \rho n_\phi$, where $|\rho|$ is the distance between the current point and the origin, then the inverse Radon transform is expressed as

$$\mu(x) = \frac{1}{(2\pi)^2}\int_0^{\pi}\int_{-\infty}^{\infty} \widetilde{\mathcal{R}\mu}(r,\theta)|r|e^{ir<x,\theta>}\,\mathrm{d}\,r\,\mathrm{d}\,\theta, \tag{2.9}$$

if $\mu(x)$ and $\widetilde{Ru}(r,\theta)$ are both absolutely integrable. The variable $|r|$ in Equation (2.9) can amplify the noise and the reconstructed images might not be satisfactory. This phenomenon also matches a slow decay of singular values of the Radon transform operator. Small singular values can exaggerate the influence of high-frequency noise so as to corrupt the reconstructed images. Compared with the inverse Radon transform, a more useful technique is called filtered back-projection (FBP) [12]. As the name indicates, it adds a filter to the inverse transform formula (2.9). In FBP, the radial integral is regarded as a filter used in the Radon transform

$$\mathcal{G}\mathcal{R}(t,\theta) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \widetilde{\mathcal{R}\mu}(r,\theta)|r|e^{irt}\,\mathrm{d}\,r. \tag{2.10}$$

On the other hand, the angular integral is explained as a back-projection of the previous transform

$$\mu(x) = \frac{1}{2}\int_0^{\pi} \mathcal{G}\mathcal{R}(<x,\theta>,\theta)\,\mathrm{d}\,\theta. \tag{2.11}$$

In contrast to the inverse Radon Transform, the filter of FBP can amplify high-

frequency components while suppressing low-frequency components so the process is more robust and reconstructed images are of higher quality and less noisy. Basically, it takes four steps to reconstruct images using FBP:

- For each angle θ, we collect measured data as (2.7).
- Compute the filter as a radial integral in (2.10).
- Calculate the back-projection as (2.11).
- Collect the reconstructions of all directions and synthesize the results.

2.3.2 Algebraic Reconstruction Technique

Rather than computing the linear attenuation coefficient μ analytically, we can also solve μ using iterative methods. If we use polar coordinates and discretize Equation (2.2) with respect to the j-th projection and the i-th detector pixel, we can obtain

$$I_i = I_0 \exp\left\{ -\int_{t \in l_{i,j}} \mu\left(\vec{r}\,(t)\right) \mathrm{d}\, t \right\}. \tag{2.12}$$

To be convenient, we let $b_i = -\log \frac{I_i}{I_0}$ and can obtain that

$$b_i = \int_{t \in l_{i,j}} \mu\left(\vec{r}\,(t)\right) \mathrm{d}\, t. \tag{2.13}$$

If we further decompose $\mu\left(\vec{r}\,(t)\right)$ into an expansion

$$\mu\left(\vec{r}\,(t)\right) = \sum_j \mu_j \phi_j\left(\vec{r}\,(t)\right), \tag{2.14}$$

where $\phi_j\left(\vec{r}\,(t)\right)$ is a basis function of the image representation. The line integral of the basis function, $a_{i,j}$, is the length of the x-ray beam through the j-th pixel of the target image, incident onto the i-th element of detector pixels. So $a_{i,j}$ can be

expressed as

$$a_{i,j} = \int_{t \in l_{i,j}} \phi_j \left(\vec{r}(t) \right) \mathrm{d}t. \tag{2.15}$$

The geometric meaning of $a_{i,j}$ is shown in Figure 2.4. In Figure 2.4, a x-ray intersects

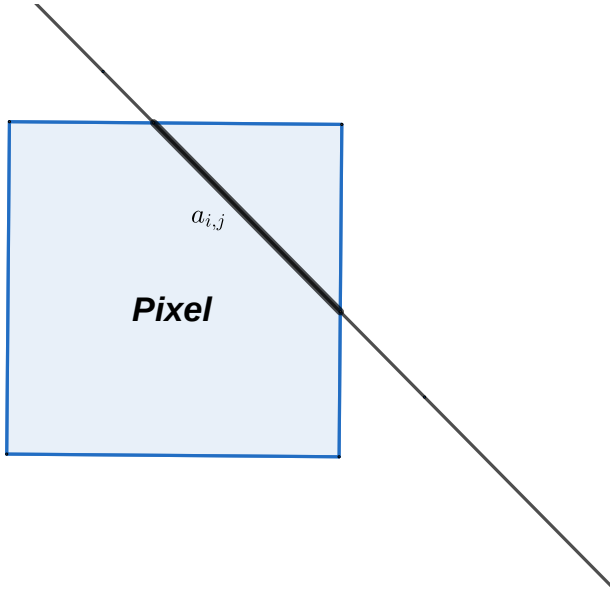

Figure 2.4: Explanation of the line integral over the j-th image basis function. The thick line inside the j-th pixel represents the distance $a_{i,j}$.

the j-th pixel of the target image and the line integral over the j-th basis function, $a_{i,j}$, corresponds to the distance of the x-ray inside the j-th pixel (the thick line). With the previous expansion, Equation (2.13) can be expressed as

$$b_i = \sum_j \mu_j \int_{t \in l_{i,j}} \phi_j \left(\vec{r}(t) \right) \mathrm{d}\, t = \sum_j a_{i,j} \mu_j. \tag{2.16}$$

If we collect $a_{i,j}$, $b_{i,j}$ and μ_j with respect to their sub-indexes, we can obtain a linear system

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{u}. \tag{2.17}$$

Therefore, given the vector $\boldsymbol{b}$ and the matrix $\boldsymbol{A}$, our goal is to compute the unknown variable $\boldsymbol{u}$ in a robust and efficient way. Since the matrix $\boldsymbol{A}$ captures the geometry of CT machines, it is not square most of time. Moreover, it can be rank-deficient if

the data we obtained are based on limited angle projections. Rather than calculate it directly, we need to find alternative methods to handle these issues.

Even if the matrix $\boldsymbol{A}$ might be rectangular, we can still build a least squares problem of the form

$$\min_{\boldsymbol{u}} \quad \|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{b}\|_2^2. \tag{2.18}$$

Many methods have been proposed to solve Equation (2.18) under different scenarios. For CT problems, one of the most famous methods is called Algebraic Reconstruction Technique (ART) [25], which is also a rediscovery of the Kaczmarz method [36]. We let $\boldsymbol{a}_i$ represent the i-th row of $\boldsymbol{A}$, then the ART method is an iteration framework given by

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \lambda_k \frac{b_i - \boldsymbol{a}_i^T \boldsymbol{u}_k}{\|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i^T, \tag{2.19}$$

where λ_k is a step size parameter in the k-th iteration. Compared with other methods, ART does not require to save the full entries of the matrix $\boldsymbol{A}$ and it only needs each row slice in the current iteration. Moreover, it has been observed for tomography problems that the initial convergence speed is often fast.

Mathematically speaking, it converges to the pseudoinverse solution, $\boldsymbol{A}^\dagger \boldsymbol{b}$. If $\boldsymbol{A}$ has full column rank, then $\boldsymbol{A}^\dagger$ can be expressed as

$$\boldsymbol{A}^\dagger = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T. \tag{2.20}$$

If $\boldsymbol{A}$ is rank-deficient and has full column rank, Problem (2.18) has infinitely many solutions. Among these infinitely many solutions, $\boldsymbol{A}\boldsymbol{A}^\dagger \boldsymbol{b}$ is the orthogonal projection of $\boldsymbol{b}$ onto the column space of $\boldsymbol{A}$. The orthogonal projection offers the "shortest" distance so the solution obtained by ART might be the "best" among all possible solutions. This is also a reason why ART is popular for limited angle CT reconstruction.

In addition to the previous iterative method, we can also build a regularized least

squares problem

$$\min_{\boldsymbol{u}} \quad \|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{b}\|_2^2 + \alpha R(\boldsymbol{u}), \tag{2.21}$$

where α is the regularization parameter and $R(\boldsymbol{u})$ is the selected regularization term. To smooth the edges, we can take ℓ_2 regularization, where $R(\boldsymbol{u}) = \|\boldsymbol{u}\|_2^2$. To strengthen the edges, we can take ℓ_1 regularization, where $R(\boldsymbol{u}) = \|\boldsymbol{u}\|_1$. In the case of ℓ_2 regularization we can build an augmented system,

$$\left\| \begin{bmatrix} \boldsymbol{A} \\ \sqrt{\alpha} \end{bmatrix} \boldsymbol{u} - \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2, \tag{2.22}$$

which can then be solved by ART. The regularization parameter can be chosen using methods such as L-curve [31], the discrepancy principle [17] and generalized cross-validation [23].

2.3.3 Statistical Reconstruction Methods

We know that x-ray beams are made of x-ray photons, and rather than computing the deterministic solution directly from an equation, we can explore the randomness of x-ray photons. If we assume that the detector is a photon-counting detector, then we can use statistical tools to build the model.

In Equation (2.3), if we discretize the equation with respect to the j-th projection and the i-th detector pixel, then we can obtain

$$y_i = \int_E S(e) \exp\left\{ - \int_{t \in l_{i,j}} \mu\left(\vec{r}\,(t)\right) \mathrm{d}\, t \right\} \mathrm{d}\, e, \tag{2.23}$$

where y_i represents the number of photons measured at the i-th pixel of detector. If we further discretize the equation over the energy spectrum, then it can be expressed

as

$$y_i = \sum_k s_k \exp\left\{ - \int_{t \in l_{i,j}} \mu\left(\vec{r}(t)\right) \mathrm{d}\, t \right\}. \tag{2.24}$$

Repeating the same steps as (2.12), (2.13) and (2.14), we can obtain

$$y_i = \bar{s}_i \exp\left\{ - \left[\boldsymbol{A}\boldsymbol{u} \right]_i \right\}, \tag{2.25}$$

where $\bar{s}_i$ is the average photon density of the j-th projection. If we both consider the errors introduced in the photon diffusion and counting, we can add a noise term

$$y_i = \bar{s}_i \exp\left\{ - \left[\boldsymbol{A}\boldsymbol{u} \right]_i \right\} + \eta_i. \tag{2.26}$$

One realistic assumption is that the projected data are Poisson distributed as

$$y_i \sim \text{Poisson}(\bar{s}_i \exp\left\{ - \left[\boldsymbol{A}\boldsymbol{u} \right]_i \right\}). \tag{2.27}$$

With this assumption, we can build a maximum likelihood function

$$f_{\boldsymbol{y}}\left(\boldsymbol{y}; \boldsymbol{u}\right) = \prod_i \frac{\bar{s}_i^{y_i} \exp\left\{ -y_i \left[\boldsymbol{A}\boldsymbol{u} \right]_i \right\} \exp\left(-\bar{s}_i \exp\left\{ - \left[\boldsymbol{A}\boldsymbol{u} \right]_i \right\} \right)}{y_i!}. \tag{2.28}$$

If we ignore the constant terms, the corresponding likelihood function is

$$L\left(\boldsymbol{u}; \boldsymbol{y}\right) = \prod_i \exp\left\{ -y_i \left[\boldsymbol{A}\boldsymbol{u} \right]_i \right\} \exp\left(-\bar{s}_i \exp\left\{ - \left[\boldsymbol{A}\boldsymbol{u} \right]_i \right\} \right). \tag{2.29}$$

The log-likelihood function can be expressed as

$$l\left(\boldsymbol{u}; \boldsymbol{y}\right) = \sum_i \left(-y_i \left[\boldsymbol{A}\boldsymbol{u} \right]_i - \bar{s}_i \exp\left\{ - \left[\boldsymbol{A}\boldsymbol{u} \right]_i \right\} \right). \tag{2.30}$$

To maximize the log-likelihood function, it is equivalent to minimizing the negative

log-likelihood function. So the objective function is

$$l(\boldsymbol{u};\boldsymbol{y}) = \sum_i \left(y_i \left[\boldsymbol{A}\boldsymbol{u}\right]_i + \bar{s}_i \exp\left\{-\left[\boldsymbol{A}\boldsymbol{u}\right]_i\right\}\right). \tag{2.31}$$

With this objective function, we can build an optimization problem as

$$\begin{aligned} &\underset{\boldsymbol{u}}{\text{argmin}} \quad l(\boldsymbol{u};\boldsymbol{y}) \\ &\text{subject to} \quad \boldsymbol{u} \geqslant 0. \end{aligned} \tag{2.32}$$

This optimization problem consists of a nonlinear objective function and bound constraints. Therefore, we cannot regard it as a linear system and compute the solution directly. Even if the nonlinearity provides more challenges for us, it also indicates numerous possibilities. To obtain the maximum likelihood estimators, nonlinear optimization skills are used to find the optimal solutions.

2.4 Ill-posed Inverse Problems

Using Hadamard's [28] definition, an inverse problem is well-posed as long as it satisfies three conditions:

- Existence: there should be at least one solution.
- Uniqueness: the solution should be unique.
- Stability: with initial conditions, the solution should depend continuously on the change.

Problems that fail to meet any of these three criteria are classified as ill-posed problems. For numerical analysis, it has corresponding explanations:

- The forward operator is compact, so it does not have a continuous inverse in the infinite-dimensional space, and the discrete problem inherits this property of

ill-posedness. Therefore, computing an accurate approximation is a challenging problem.

- The forward operator or linear system has small singular values. When we solve inverse problems, the inversion of small singular values will amplify the noise so the solution might be far away from the truth.

In our cases, the computed tomography problem is an ill-posed inverse problem since the forward operator has a decay of small singular values and the rank-deficient system has multiple solutions. To mitigate the effects, three classical methods use different techniques:

- The filtered back-projection uses a filter to emphasize the high-frequency components and reduce the influence of noise.

- Algebraic Reconstruction Technique takes orthogonal projections and regularizations to obtain a better solution. If the regularization operator $R(\boldsymbol{u})$ is omitted, the early termination of the iteration can be used to avoid noise applification in the solution.

- Statistical reconstruction methods apply randomness of photons and compute the maximum likelihood estimator as evidence.

The solutions obtained using these classical methods are of higher quality. However, these results are based on the assumption of either a monochromatic x-ray source or an approximation of average of photon flux density. Furthermore, we cannot tell the composition of the target object we obtain as well as the percentages of different materials. In the following sections, we will focus on the spectral CT models, the material decomposition and optimization frameworks used to compute robust and efficient solutions.

Chapter 3

Nonlinear Optimization for the Energy Integrating Detector Model

In this chapter we focus on spectral computed tomography with a single energy discriminating detector. To reconstruct material maps of each composition, we need to find solutions to an inverse problem of the form

$$\boldsymbol{y} = \exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{s} + \boldsymbol{\eta}, \tag{3.1}$$

where $\boldsymbol{y}$ is a vector of projection data, $\boldsymbol{s}$ is a vector containing spectral information of the corresponding energies of the source x-ray, and $\boldsymbol{\eta}$ is the noise term. The exponentiation is done element-wise. $\boldsymbol{A}$ is a matrix that is related to the quantitative information of ray trace, $\boldsymbol{C}$ is a matrix of (known) material specific attenuation coefficients, and $\boldsymbol{W}$ is a matrix of unknowns, whose columns correspond to the weights of each of known materials of the object being imaged.

To solve this problem, Elbakri and Fessler [15] and Chung et al. [11] suggest that we could use a 2-material model with a polyenergetic assumption on the source x-ray. Moreover, Mejia-Bustamante et al. [7] extended this idea, and provided a GPU implementation [6]. We remark that the problem can be reformulated in terms of mass

attenuation coefficients (see, e.g., [15]), which encodes density information into the mathematical model. This can be important in cases when materials have the same chemical composition but different densities. However, it does not fundamentally change our proposed optimization approach, so in the remainder of the chapter we use the model based on linear attenuation coefficients. To solve the nonlinear inverse problem in [6, 7], a maximum likelihood function was used to represent the evidence with respect to parameters $\boldsymbol{W}$. The gradient descent method is used to solve the resulting optimization problem, with an implicit enforcement of the constraint that in each voxel, the weights across all materials should sum to one.

In this chapter, we build an optimization problem with Poisson maximum likelihood function and use a problem-specific nonlinear interior point trust region method to solve it. Moreover, we provide a modified Hessian that is close to the true Hessian and is also positive semi-definite. The interior point method has proven to be efficient and stable for solving nonlinear optimization problems, and with the implementation of our modified Hessian, we can simultaneously obtain an accurate approximation of the true Hessian and avoid negative curvature. Numerical experiments have shown very promising results for this method.

In Section 3.1, we present the general polyenergetic model for spectral computed tomography and derive the specific mixed attenuation model from the general model. The discretization of this model is included in Section 3.1 as well. In Section 3.2, we revise this model to one that is more amenable to numerical implementation and formulate an optimization problem that is based on Poisson likelihood function. The standard form of this optimization problem and the method to solve it are discussed in Section 3.3. Moreover, numerical experiments for both full CT and limited angle reconstruction are presented in Section 3.4. In Section 3.5, we conclude with merits and limitations about the model and the optimization method.

3.1 The Energy Integrating Detector Model

The computed tomography (CT) process mainly consists of three parts: x-ray beams are emitted from a source with specific energies, an object is illuminated by x-ray beams and attenuated x-ray beams are received by a detector. In this process, the intensities of x-ray beams are reduced and using Beer's law [18], the energy integrating detector model can be written as

$$y_i = \int_E S(e) \exp\left(- \int_{t \in \ell} \mu\left(\vec{r}\,(t)\,, e\right) \mathrm{d}\, t \right) \mathrm{d}\, e + \eta_i, \quad i = 1, 2, \cdots, N_d \times N_p, \tag{3.2}$$

where

- y_i is the x-ray intensity of the i-th pixel in the detector.
- E is the photon flux density. Figure 3.1 shows a curve of E versus photon energy with relative low potential (26 keV).
- N_d is the number of detector pixels. For a material map of size n by n, we assume $N_d = n$ and the number of projection rays for each angle is equivalent to $\lfloor \sqrt{2} N_d \rfloor$.
- N_p is the number of projections. For cone/fan beam CT, projections are distributed equally from 0 to 360 degrees.
- $S(e)$ represents the system spectral response, which is a product of x-ray energy with the number of incident photons at that energy.
- The outer integral is over all x-ray energies emitted from the source, and the inner integral is along lines that follow the x-ray beam paths through the object.
- $\mu\left(\vec{r}\,(t)\,, e\right)$ denotes the attenuation coefficient, which is related to the position function $\vec{r}\,(t)$ and the energy level e.

- η_i represents unknown errors in the measurements, which can include x-ray scatter and electronic noise.

The traditional methods to solve this inverse problem are mostly based on filtered back-projection (FBP) [49]. If we assume the source to be monoenergetic and the source energy is s_e, then we can build a linear inverse problem by dividing s_e on both sides and by applying the natural logarithm function to the data and to the model. Other approaches can be used to solve this linear inverse problem, such as incorporating different regularization schemes. These usually involve applying appropriate iterative optimization methods, such as the conjugate gradient method [45]. However, the images obtained from traditional methods on the simplified linear model might lead to significant beam hardening artifacts when the object is made up of several very distinct materials, such as bone and soft tissue. In addition, the linear models cannot be used to recognize the the actual types of materials from the results, nor can they separate different materials when they are mixed [33]. Moreover, the traditional methods are unstable when it comes to the limited angle cases. For these reasons, we consider the full nonlinear polyenergetic model.

In Model (3.2), the unknown linear attenuation coefficient $\mu\left(\vec{r}\left(t\right), e\right)$ is dependent on the position function $r\left(t\right)$ and 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 $\mu\left(\vec{r}\left(t\right), e\right)$ [33]:

$$\mu\left(\vec{r}\left(t\right), e\right) = \sum_{m=1}^{N_m} u_{m,e} w_m\left(\vec{r}\right), \tag{3.3}$$

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.

- $w_m(\vec{r})$ is the unknown weight of the m-th material at the position $\vec{r}$.

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

$$w_m(\vec{r}) = \sum_{j=1}^{N_v} w_{j,m}\phi_j(\vec{r}), \tag{3.4}$$

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).
- $\phi_j(\vec{r})$ 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 the detector pixels N_d and the number of projections N_p:

$$a_{i,j} = \int_{t\in l} \phi_j(\vec{r}(t))\,\mathrm{d}\,t. \tag{3.5}$$

By assumption, the attenuation coefficient for the m-th material and the energy level e, $u_{m,e}$, is already known and the only unknown variable is the weight $w_{j,m}$. Using this expression, we have transformed the goal of solving $\mu_{j,e}$ to the target of solving for $w_{j,m}$, which is dependent on the number of voxels and the number of materials. Usually, the number of materials is 2 or 3 and it is significantly fewer than the number of energies. To simplify the problem, we also assume that the sum of weights inside each voxel is equivalent to 1. That is to say,

$$\sum_{m=1}^{N_m} w_{j,m} = 1. \tag{3.6}$$

In this chapter, we limit the discussion to 2 to 3 materials, but note that for energy discriminating detectors, separating more materials is feasible. With (3.3), (3.4) and (3.5), the line integral in Model (3.2) is expressed by

$$\int_{t\in l} \mu\left(\vec{r}\left(t\right), e\right) \mathrm{d}\, t = \sum_{m=1}^{N_m} \sum_{j=1}^{N_v} u_{m,e} w_{j,m} \int_{t\in l} \phi_j\left(\vec{r}\left(t\right)\right) \mathrm{d}\, t = \sum_{j=1}^{N_v} \sum_{m=1}^{N_m} a_{i,j} w_{j,m} u_{m,e}. \quad (3.7)$$

If we also discretize the integral over energy E and ignore quadrature errors, then the discrete model of Equation (3.2) can be written as:

$$y_i = \sum_{e=1}^{N_e} s_e \exp\left(-\sum_{j=1}^{N_v} \sum_{m=1}^{N_m} a_{i,j} w_{j,m} u_{m,e}\right) + \eta_i, \quad (3.8)$$

where N_e is the number of discrete energies. If we collect $a_{i,j}$, $w_{i,j}$ and $u_{m,e}$ in matrix form and concatenate y_i, s_e, η_i into vectors, then the corresponding equation can be represented as

$$\boldsymbol{y} = \exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{s} + \boldsymbol{\eta}, \quad (3.9)$$

where

- $\boldsymbol{y}$ is a vector of the size $N_d \cdot N_p$ that gathers x-ray photons.
- $\boldsymbol{A}$ 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 e-th energy and the m-th material.
- $\boldsymbol{s}$ is a vector of the size N_e and collects the spectrum energy of a specific range.
- $\boldsymbol{\eta}$ is the noise vector that is of the size $N_d \cdot N_p$.

In Equation (3.9), the exponential function is point-wise exponential rather than

matrix function. In addition to Equation (3.9), we also require that weight fractions should be nonnegative and this can be illustrated by the constraint $\boldsymbol{W} \geqslant \boldsymbol{0}$. We have so far obtained the standard form of the polyenergetic multi-material model. Based on Equation (3.9), the goal is to solve for the unknown weight matrix $\boldsymbol{W}$ such that $\boldsymbol{W} \in \boldsymbol{P}$, where $\boldsymbol{P} = \{\boldsymbol{W} \mid \boldsymbol{W}\boldsymbol{1}_{N_m} = \boldsymbol{1}_{N_v},\ \boldsymbol{0} \leqslant \boldsymbol{W} \leqslant \boldsymbol{1}\}$. $\boldsymbol{1}_{N_m}$ and $\boldsymbol{1}_{N_v}$ are vectors of ones of the lengths N_m and N_v, respectively.

3.2 Poisson Log-likelihood Function

With different energy levels, the forward model is nonlinear and it is not possible to transform it into an equivalent linear model. Mejia-Bustamante et al. [7] use the assumption that each entry in $\boldsymbol{y}$ follows a Poisson distribution,

$$\boldsymbol{y} \sim \text{Poisson}\left(\exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{s}\right). \tag{3.10}$$

With this assumption, one can formulate a maximum likelihood estimator (MLE) based on Poisson distribution. Before we show the expressions of objective function, gradient and Hessian, we first note that if the weights (unknowns) are stored as a matrix, then differentiation results in tensors, requiring tedious bookkeeping in the computations. In this work we instead rewrite the function to put the unknowns in vector form, and differentiate the objective function with respect to this vector. Notice that

$$\text{vec}(\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T) = (\boldsymbol{C} \otimes \boldsymbol{A})\,\text{vec}\,(\boldsymbol{W}), \tag{3.11}$$

where $\text{vec}(\cdot)$ reshapes a given matrix into a vector by stacking the columns on top of each other. Therefore, we can rewrite Equation (3.9) as

$$\boldsymbol{y} = (\boldsymbol{s}^T \otimes \boldsymbol{I})\exp\left[-\left(\boldsymbol{C} \otimes \boldsymbol{A}\right)\text{vec}\,(\boldsymbol{W})\right] + \boldsymbol{\eta}. \tag{3.12}$$

If we let

$$\boldsymbol{w} = \mathrm{vec}\left(\boldsymbol{W}\right) \quad \text{and} \quad K\left(\boldsymbol{w}\right) = \exp\left\{-\left(\boldsymbol{C} \otimes \boldsymbol{A}\right)\boldsymbol{w}\right\}, \tag{3.13}$$

then Equation (3.12) is equivalent to

$$\boldsymbol{y} = (\boldsymbol{s}^T \otimes \boldsymbol{I})K\left(\boldsymbol{w}\right) + \boldsymbol{\eta}. \tag{3.14}$$

In this problem, we assume that each element of measured data, $\boldsymbol{y}_i$, follows a Poisson distribution with mean $\left[\left(\boldsymbol{s}^T \otimes \boldsymbol{I}\right) K(\boldsymbol{w})\right]_i$. That is to say,

$$y_i \sim \text{Poisson}\left(\left[\left(\boldsymbol{s}^T \otimes \boldsymbol{I}\right) K(\boldsymbol{w})\right]_i\right). \tag{3.15}$$

Based on this assumption, the corresponding probability density function can be expressed as

$$f_{\boldsymbol{y}}\left(\boldsymbol{y};\boldsymbol{w}\right) = \prod_{i=1}^{N_d \times N_p} \frac{[(\boldsymbol{s}^T \otimes \boldsymbol{I})K(\boldsymbol{w})]_i^{y_i} \exp\left([-\left(\boldsymbol{s}^T \otimes \boldsymbol{I}\right) K(\boldsymbol{w})]_i\right)}{y_i!}. \tag{3.16}$$

If we ignore the constant terms, the corresponding likelihood function is

$$L\left(\boldsymbol{w};\boldsymbol{y}\right) = \prod_{i=1}^{N_d \times N_p} [(\boldsymbol{s}^T \otimes \boldsymbol{I})K(\boldsymbol{w})]_i^{y_i} \exp\left([-\left(\boldsymbol{s}^T \otimes \boldsymbol{I}\right) K(\boldsymbol{w})]_i\right). \tag{3.17}$$

Taking the logarithm, the log-likelihood function is expressed by

$$l\left(\boldsymbol{w};\boldsymbol{y}\right) = \sum_{i=1}^{N_d \times N_p} \left\{y_i \log[(\boldsymbol{s}^T \otimes \boldsymbol{I})K(\boldsymbol{w})]_i - [\left(\boldsymbol{s}^T \otimes \boldsymbol{I}\right) K(\boldsymbol{w})]_i\right\}. \tag{3.18}$$

To maximize the log-likelihood function, it is equivalent to minimizing the negative log-likelihood function. So the objective function is

$$\begin{aligned} l\left(\boldsymbol{w};\boldsymbol{y}\right) = & \sum_{i=1}^{N_d\times N_p} \left\{ \left[\left(\boldsymbol{s}^T \otimes \boldsymbol{I}\right) K(\boldsymbol{w}) \right]_i - y_i \log[(\boldsymbol{s}^T \otimes \boldsymbol{I})K(\boldsymbol{w})]_i \right\}. \\ = & \mathbf{1}_{N_p\times N_\theta}^T (\boldsymbol{s}^T \otimes \boldsymbol{I})K(\boldsymbol{w}) - \boldsymbol{y}^T \log\left[(\boldsymbol{s}^T \otimes \boldsymbol{I})K(\boldsymbol{w})\right], \end{aligned} \tag{3.19}$$

where $\mathbf{1}_{N_d\times N_p}$ is a vector of all ones of the length $N_d \times N_p$. The gradient to the objective function can be expressed as

$$\nabla l(\boldsymbol{w}) = -\left(\boldsymbol{C}^T \otimes \boldsymbol{A}^T\right) \operatorname{diag}\left\{K(\boldsymbol{w})\right\} (\boldsymbol{s} \otimes \boldsymbol{I}) \left\{\mathbf{1}_{N_p\times N_\theta} - \boldsymbol{y} \oslash \left[(\boldsymbol{s}^T \otimes \boldsymbol{I})K(\boldsymbol{w})\right]\right\}. \tag{3.20}$$

Differentiating the gradient with respect to $\boldsymbol{w}$, the Hessian can be represented as a summation of two parts:

$$\nabla^2 l(\boldsymbol{w}) = H_1\left(\boldsymbol{w}\right) + H_2\left(\boldsymbol{w}\right), \tag{3.21}$$

where

$$\begin{aligned} H_1(\boldsymbol{w}) = & (\boldsymbol{C}^T \otimes \boldsymbol{A}^T)\operatorname{diag}\left\{K(\boldsymbol{w})\right\} \operatorname{diag}\left\{(\boldsymbol{s} \otimes \boldsymbol{I})\left(\mathbf{1}_{N_p\times N_\theta} - \boldsymbol{y} \oslash \left[(\boldsymbol{s}^T \otimes \boldsymbol{I})K(\boldsymbol{w})\right]\right)\right\} (\boldsymbol{C} \otimes \boldsymbol{A}) \\ H_2(\boldsymbol{w}) = & (\boldsymbol{C}^T \otimes \boldsymbol{A}^T)\operatorname{diag}\left\{K(\boldsymbol{w})\right\} (\boldsymbol{s} \otimes \boldsymbol{I})\operatorname{diag}\left\{\boldsymbol{y} \oslash [(\boldsymbol{S}^T \otimes \boldsymbol{I})K(\boldsymbol{w})].^2\right\} \\ & (\boldsymbol{s}^T \otimes \boldsymbol{I})\operatorname{diag}\left\{K(\boldsymbol{w})\right\} (\boldsymbol{C} \otimes \boldsymbol{A}). \end{aligned} \tag{3.22}$$

From the previous expressions, we can see that $H_2\left(\boldsymbol{w}\right)$ is the Gauss-Newton approximation to the true Hessian and it is always positive semi-definite. Even if $H_1\left(\boldsymbol{w}\right)$ is indefinite, we can transform it into a positive semi-definite matrix by setting a

threshold. Specifically, let $T(\boldsymbol{w})$ be defined as

$$T(\boldsymbol{w}) = \max\left[\mathbf{0},\ (\boldsymbol{s}\otimes\boldsymbol{I})\left(\mathbf{1}_{N_p\times N_\theta} - \boldsymbol{y}\oslash\left[(\boldsymbol{s}^T\otimes\boldsymbol{I})K(\boldsymbol{w})\right]\right)\right]. \tag{3.23}$$

Then the new $\hat{H}_1(\boldsymbol{w})$ can be represented as

$$\hat{H}_1(\boldsymbol{w}) = \left(\boldsymbol{C}^T\otimes\boldsymbol{A}^T\right)\mathrm{diag}\left\{K(\boldsymbol{w})\right\}\mathrm{diag}\left\{T(\boldsymbol{w})\right\}(\boldsymbol{C}\otimes\boldsymbol{A}). \tag{3.24}$$

Moreover, we define the modified Hessian as

$$\hat{H}(\boldsymbol{w}) = \hat{H}_1(\boldsymbol{w}) + H_2(\boldsymbol{w}). \tag{3.25}$$

With the modified Hessian, we can include most information about the true Hessian as well as keep it positive semi-definite. Furthermore, both the bound constraint and the constraint (3.6) should be included. The bound constraint is equivalent to $\mathbf{0}\leqslant\boldsymbol{w}\leqslant\mathbf{1}$ and we can rewrite the constraint (3.6) in a matrix-vector form as

$$\boldsymbol{A}_{eq}\boldsymbol{w} = \mathbf{1}_{N_v}, \tag{3.26}$$

where $\boldsymbol{A}_{eq}$ is a matrix of the form

$$\boldsymbol{A}_{eq} = \mathbf{1}_{N_m}^T\otimes\boldsymbol{I}_{N_v}. \tag{3.27}$$

$\mathbf{1}_{N_m}$ is a vector of ones of the length N_m. $\boldsymbol{I}_{N_v}$ is an identity matrix of the size $N_v\times N_v$. In the following sections, we use $\boldsymbol{I}$ to represent the identity matrix if the size of this matrix is clear to identify.

With the objective function (3.19), we can construct an optimization problem by

combining the regularization term and the constraints:

$$\begin{aligned} \min_{\mathbf{0} \leqslant \boldsymbol{w} \leqslant \mathbf{1}} \quad & f(\boldsymbol{w}) + \alpha R(\boldsymbol{w}) \\ \text{subject to} \quad & \boldsymbol{A}_{eq}\boldsymbol{w} = \mathbf{1}_{N_v}. \end{aligned} \tag{3.28}$$

In Problem (3.28), $f(\boldsymbol{w}) = l(\boldsymbol{w}; \boldsymbol{y})$ and $R(\boldsymbol{w})$ represents the regularization term, which is used to penalize the variable $\boldsymbol{w}$. α is the corresponding regularization parameter. For this problem, we choose the total variation regularization to preserve edges. Using the previous notations, we can express the bound constraint and the equality constraint as $\boldsymbol{Q} = \{\boldsymbol{w} \mid \boldsymbol{A}_{eq}\boldsymbol{w} = \mathbf{1}_{N_v},\ \mathbf{0} \leqslant \boldsymbol{w} \leqslant \mathbf{1}\}$. So the notation of Problem (3.28) can be simplified as

$$\min_{\boldsymbol{w} \in \boldsymbol{Q}} \quad f(\boldsymbol{w}) + \alpha R(\boldsymbol{w}). \tag{3.29}$$

We let $\boldsymbol{w}_k$ be the k-th column in the matrix $\boldsymbol{W}$ and $\boldsymbol{W}_k$ be the corresponding reshaped image of $\boldsymbol{w}_k$. The total variation regularization for the k-th material can be represented as [30]:

$$R(\boldsymbol{W}_k) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\left(\boldsymbol{W}_k\boldsymbol{D}^T\right)_{ij}^2 + \left(\boldsymbol{D}\boldsymbol{W}_k\right)_{ij}^2\right)^{1/2}, \tag{3.30}$$

where $\boldsymbol{D}$ is either a forward, backward or central first order finite difference matrix and n is the dimension of the corresponding material map. For all m materials, the regularization term can be expressed as

$$R(\boldsymbol{W}) = \sum_{k=1}^{m} R(\boldsymbol{W}_k). \tag{3.31}$$

Other forms of regularization, such as the discrete Laplacian, can also be used with this framework. For simplicity, we use zero boundary conditions and assume that the regularization parameters for different materials are equal to α. With the regulariza-

tion terms, we can reduce the influence of noise and stabilize the solution.

With the modified Hessian and regularizations, we want to use a second order method to solve the optimization problem (3.28). However, direct implementation of the Newton's method is not effective. To guarantee the feasibility of each step, we need to project the current step onto the boundaries. The projected solution might be far away from the desired solution and hard to improve later. Furthermore, it is difficult to maintain the equality constraint in each step without changing the original model.

3.3 Implementation of Nonlinear Interior Point Trust Region Method

Recall that the optimization problem can be expressed as

$$\begin{aligned} \min_{\mathbf{0} \leqslant \boldsymbol{w} \leqslant \mathbf{1}} \quad & f(\boldsymbol{w}) + \alpha R(\boldsymbol{w}) \\ \text{subject to} \quad & \boldsymbol{A}_{eq}\boldsymbol{w} = \mathbf{1}, \end{aligned} \tag{3.32}$$

where $f(\boldsymbol{w}) = l(\boldsymbol{w}; \boldsymbol{y})$ and $R(\boldsymbol{w})$ is the regularization term.

To solve this constrained optimization problem, we use a nonlinear interior point trust region method, which combines sequential quadratic programming (SQP), a trust region dogleg method, and a projected conjugate gradient algorithm [8, 48]. To apply this method, we firstly establish a barrier problem based on (3.32):

$$\begin{aligned} \min_{\boldsymbol{w}, \boldsymbol{z}} \quad & f(\boldsymbol{w}) + \alpha R(\boldsymbol{w}) - \beta \sum_{i=1}^{2N_v} \ln(z_i) \\ \text{subject to} \quad & \boldsymbol{A}_{eq}\boldsymbol{w} - \mathbf{1} = \mathbf{0}, \\ & \boldsymbol{A}_{ieq}\boldsymbol{w} + \boldsymbol{y}_{ieq} + \boldsymbol{z} = \mathbf{0}, \end{aligned} \tag{3.33}$$

where

$$\boldsymbol{A}_{ieq} = \begin{bmatrix} -\boldsymbol{I} \\ \boldsymbol{I} \end{bmatrix} \quad \text{and} \quad \boldsymbol{y}_{ieq} = \begin{bmatrix} -\mathbf{1} \\ \mathbf{0} \end{bmatrix}. \tag{3.34}$$

β is the barrier parameter that should decrease to 0 and $\boldsymbol{z}$ is the vector of slack variables that ensure all entries remain positive. The *permuted* KKT condition corresponding to (3.33) can be written as

$$\begin{aligned} \nabla f(\boldsymbol{w}) + \alpha \nabla R(\boldsymbol{w}) + \boldsymbol{A}_{eq}^T \boldsymbol{\lambda}_{eq} + \boldsymbol{A}_{ieq}^T \boldsymbol{\lambda}_{ieq} &= \mathbf{0}, \\ \boldsymbol{Z}\boldsymbol{\lambda}_{ieq} - \beta \mathbf{1} &= \mathbf{0}, \\ \boldsymbol{A}_{eq}\boldsymbol{w} - \mathbf{1} &= \mathbf{0}, \\ \boldsymbol{A}_{ieq}\boldsymbol{w} + \boldsymbol{y}_{ieq} + \boldsymbol{z} &= \mathbf{0}, \end{aligned} \tag{3.35}$$

where $\boldsymbol{\lambda}_{eq}$ and $\boldsymbol{\lambda}_{ieq}$ are the Lagrange multipliers corresponding to the equality and the inequality constraints, respectively. Furthermore, we should keep $\boldsymbol{\lambda}_{ieq}$ nonnegative. $\boldsymbol{Z}$ is a diagonal matrix and $\boldsymbol{Z} = \text{diag}\{z_1, z_2, \cdots, z_{2N_e}\}$. Compared with the original KKT system, this permuted KKT system is preferred because the matrix $\boldsymbol{Z}$ is bounded when the entries of $\boldsymbol{z}$ approach 0. We can construct an error function based on this system:

$$\begin{aligned} E(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\lambda}_{eq}, \boldsymbol{\lambda}_{ieq}; \beta) = \max\{ &\| f(\boldsymbol{w}) + \alpha \nabla R(\boldsymbol{w}) \\ &+ \boldsymbol{A}_{eq}^T \boldsymbol{\lambda}_{eq} + \boldsymbol{A}_{ieq}^T \boldsymbol{\lambda}_{ieq} \|_\infty, \\ &\| \boldsymbol{Z}\lambda_{ieq} - \beta \|_\infty, \| \boldsymbol{A}_{eq}\boldsymbol{w} - \mathbf{1} \|_\infty, \\ &\| \boldsymbol{A}_{ieq}\boldsymbol{w} + \boldsymbol{y}_{ieq} + \boldsymbol{z} \|_\infty \}. \end{aligned} \tag{3.36}$$

If the error function is less than a tolerance, for example, 10^{-8}, then we assume that we have solved this system accurately. The Newton system corresponding to the

permuted KKT system can be written as

$$D(\boldsymbol{w})\,\boldsymbol{p} = -g(\boldsymbol{w}), \tag{3.37}$$

where

$$D(\boldsymbol{w}) = \begin{bmatrix} \hat{H}(\boldsymbol{w}) + \alpha\nabla^2 R(\boldsymbol{w}) & \mathbf{0} & \boldsymbol{A}_{eq}^T & \boldsymbol{A}_{ieq}^T \\ \mathbf{0} & \boldsymbol{\Lambda}_{ieq} & \mathbf{0} & \boldsymbol{Z} \\ \boldsymbol{A}_{eq} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{A}_{ieq} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$g(\boldsymbol{w}) = \begin{bmatrix} \nabla f(\boldsymbol{w}) + \alpha\nabla R(\boldsymbol{w}) + \boldsymbol{A}_{eq}^T\boldsymbol{\lambda}_{eq} + \boldsymbol{A}_{ieq}^T\boldsymbol{\lambda}_{ieq} \\ \boldsymbol{Z}\boldsymbol{\lambda}_{ieq} - \beta\mathbf{1} \\ \boldsymbol{A}_{eq}\boldsymbol{w} - \mathbf{1} \\ \boldsymbol{A}_{ieq}\boldsymbol{w} + \boldsymbol{y}_{ieq} + \boldsymbol{z} \end{bmatrix}, \tag{3.38}$$

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{p}_x \\ \boldsymbol{p}_z \\ \Delta\boldsymbol{\lambda}_{eq} \\ \Delta\boldsymbol{\lambda}_{ieq} \end{bmatrix}.$$

In the matrix $D(\boldsymbol{w})$, $\boldsymbol{\Lambda}_{ieq}$ is a diagonal matrix with $\boldsymbol{\lambda}_{ieq}$ in the diagonal: $\boldsymbol{\Lambda}_{ieq} = \text{diag}\{\,\boldsymbol{\lambda}_{ieq}\}$. However, both direct and iterative methods to solve this system are computationally expensive, especially for large-scale problems. We therefore follow the strategy of Byrd et al. [9] by transforming the original problem into a sequential quadratic programming problem and solve by separating it into two subproblems. The first subproblem is called the normal subproblem, which is solved by trust-region dogleg method, while the second subproblem, the tangential subproblem, can be solved by the projected conjugate gradient method. By applying the idea of sequential

quadratic programming, we can construct an optimization problem as

$$\begin{aligned}
\min_{\boldsymbol{p}_x,\boldsymbol{p}_z} \quad & \nabla\left[f\left(\boldsymbol{w}\right)+\alpha R\left(\boldsymbol{w}\right)\right]^T\boldsymbol{p}_x+\frac{1}{2}\boldsymbol{p}_x^T\left[\hat{H}\left(\boldsymbol{w}\right)+\alpha\nabla^2 R\left(\boldsymbol{w}\right)\right]\boldsymbol{p}_x \\
& -\beta\mathbf{1}^T\boldsymbol{Z}^{-1}\boldsymbol{p}_z+\frac{1}{2}\boldsymbol{p}_z^T\boldsymbol{\Sigma}\boldsymbol{p}_z \\
\text{subject to} \quad & \boldsymbol{A}_{eq}\boldsymbol{p}_x+\boldsymbol{A}_{eq}\boldsymbol{w}-\mathbf{1}=\boldsymbol{r}_{eq}, \\
& \boldsymbol{A}_{ieq}\boldsymbol{p}_x+\boldsymbol{p}_z+\boldsymbol{A}_{ieq}\boldsymbol{w}+\boldsymbol{y}_{ieq}+\boldsymbol{z}=\boldsymbol{r}_{ieq}, \\
& \left\|\begin{bmatrix}\boldsymbol{p}_x^T & \boldsymbol{p}_z^T\boldsymbol{Z}^{-1}\end{bmatrix}\right\|_2\leqslant\Delta, \\
& \boldsymbol{p}_z\geqslant-\tau\boldsymbol{z},
\end{aligned} \tag{3.39}$$

where $\boldsymbol{\Sigma}=\boldsymbol{Z}^{-1}\boldsymbol{\Lambda}_{ieq}$. For this primal-dual system, we regard $\boldsymbol{\Sigma}=\boldsymbol{Z}^{-1}\boldsymbol{\Lambda}_{ieq}$ as an approximation to the second order derivative $\beta\boldsymbol{Z}^{-2}$. $\boldsymbol{r}_{eq}$ and $\boldsymbol{r}_{ieq}$ are auxiliary variables to the linearized constraints. Moreover, the trust region constraint is used to guarantee a sufficient reduction in each step and stop the loop after a fixed number of iterations. The scale term $\boldsymbol{Z}^{-1}$ in the trust region constraint is used to keep slack variables away from zero without enough iterations. Meanwhile, the bound constraint is applied to guarantee the positivity of the slack variable $\boldsymbol{z}$. That is to say, if the step is accepted, then we should have

$$\boldsymbol{z}+\boldsymbol{p}_z\geqslant\boldsymbol{z}-\tau\boldsymbol{z}=\left(1-\tau\right)\boldsymbol{z}>0. \tag{3.40}$$

τ is a constant that is close to 1, for example, $\tau=0.98$. If we let $\tilde{\boldsymbol{p}}_z=\boldsymbol{Z}^{-1}\boldsymbol{p}_z$, then

we can rewrite (3.39) as

$$\begin{aligned}
\min_{\boldsymbol{p}_x,\tilde{\boldsymbol{p}}_z} \quad & \nabla\left[f\left(\boldsymbol{w}\right)+\alpha R\left(\boldsymbol{w}\right)\right]^T\boldsymbol{p}_x+\frac{1}{2}\boldsymbol{p}_x^T\left[\hat{H}\left(\boldsymbol{w}\right)+\alpha\nabla^2 R\left(\boldsymbol{w}\right)\right]\boldsymbol{p}_x \\
& -\beta\mathbf{1}^T\tilde{\boldsymbol{p}}_z+\frac{1}{2}\tilde{\boldsymbol{p}}_z^T\boldsymbol{Z}\boldsymbol{\Sigma}\boldsymbol{Z}\tilde{\boldsymbol{p}}_z \\
\text{subject to} \quad & \boldsymbol{A}_{eq}\boldsymbol{p}_x+\boldsymbol{A}_{eq}\boldsymbol{w}-\mathbf{1}=\boldsymbol{r}_{eq}, \\
& \boldsymbol{A}_{ieq}\boldsymbol{p}_x+\boldsymbol{Z}\tilde{\boldsymbol{p}}_z+\boldsymbol{A}_{ieq}\boldsymbol{w}+\boldsymbol{y}_{ieq}+\boldsymbol{z}=\boldsymbol{r}_{ieq}, \\
& \|\begin{bmatrix}\boldsymbol{p}_x^T & \tilde{\boldsymbol{p}}_z^T\end{bmatrix}\|_2\leqslant\Delta, \\
& \tilde{\boldsymbol{p}}_z\geqslant-\tau.
\end{aligned} \tag{3.41}$$

With the help of auxiliary variables, we can separate (3.41) into two subproblems, the normal subproblem and the tangential subproblem. The normal subproblem can be expressed as

$$\begin{aligned}
\min_{\boldsymbol{v}_x,\boldsymbol{v}_z} \quad & \|\boldsymbol{A}_{eq}\boldsymbol{v}_x+\boldsymbol{A}_{eq}\boldsymbol{w}-\mathbf{1}\|_2^2+\|\boldsymbol{A}_{ieq}\boldsymbol{v}_x+\boldsymbol{Z}\boldsymbol{v}_z+\boldsymbol{A}_{ieq}\boldsymbol{w}+\boldsymbol{y}_{ieq}+\boldsymbol{z}\|_2^2 \\
\text{subject to} \quad & \|\begin{bmatrix}\boldsymbol{v}_x^T & \boldsymbol{v}_z^T\end{bmatrix}\|_2\leqslant\zeta\Delta, \\
& \boldsymbol{v}_z\geqslant-\frac{\tau}{2},
\end{aligned} \tag{3.42}$$

where ζ is a constant and $0<\zeta<1$, for example, $\zeta=0.8$. Without the bound constraint, the normal subproblem is a standard form of trust region problem, which can be solved by the trust region dogleg method. So we solve it first by ignoring the bound constraint and test if the solution satisfies this constraint later. If not, we conduct backtracking to maintain feasibility [54]. After solving the normal subproblem approximately, we obtain the residuals as

$$\begin{aligned}
\boldsymbol{r}_{eq}&=\boldsymbol{A}_{eq}\boldsymbol{v}_x+\boldsymbol{A}_{eq}\boldsymbol{w}-\mathbf{1}, \\
\boldsymbol{r}_{ieq}&=\boldsymbol{A}_{ieq}\boldsymbol{v}_x+\boldsymbol{Z}\boldsymbol{v}_z+\boldsymbol{A}_{ieq}\boldsymbol{w}+\boldsymbol{y}_{ieq}+\boldsymbol{z}.
\end{aligned} \tag{3.43}$$

By substituting the residuals for the same terms in (3.41), the original optimization

problem can be represented as

$$\begin{aligned}
\min_{\boldsymbol{p}_x, \tilde{\boldsymbol{p}}_z} \quad & \nabla \left[f\left(\boldsymbol{w}\right) + \alpha R\left(\boldsymbol{w}\right) \right]^T \boldsymbol{p}_x + \frac{1}{2} \boldsymbol{p}_x^T \left[\hat{H}\left(\boldsymbol{w}\right) + \alpha \nabla^2 R\left(\boldsymbol{w}\right) \right] \boldsymbol{p}_x \\
& - \beta \mathbf{1}^T \tilde{\boldsymbol{p}}_z + \frac{1}{2} \tilde{\boldsymbol{p}}_z^T \boldsymbol{Z} \boldsymbol{\Sigma} \boldsymbol{Z} \tilde{\boldsymbol{p}}_z \\
\text{subject to} \quad & \boldsymbol{A}_{eq} \left(\boldsymbol{p}_x - \boldsymbol{v}_x \right) = 0, \\
& \boldsymbol{A}_{ieq} \left(\boldsymbol{p}_x - \boldsymbol{v}_x \right) + \boldsymbol{Z} \left(\tilde{\boldsymbol{p}}_z - \boldsymbol{v}_z \right) = 0, \\
& \| \begin{bmatrix} \boldsymbol{p}_x^T & \tilde{\boldsymbol{p}}_z^T \end{bmatrix} \|_2 \leqslant \Delta, \\
& \tilde{\boldsymbol{p}}_z \geqslant -\tau.
\end{aligned} \tag{3.44}$$

If we ignore the last two constraints, this optimization problem is a standard form of quadratic programming problem under linear equality constraints, which can be solved by the projected conjugate gradient method. In this case, we ignore the bound constraint at first and stop the iteration when the desired tolerance is attained or the current step crosses the trust region boundary. If the solution does not satisfy the bound constraint, we backtrack and choose the last feasible step as the solution.

After we have obtained $\boldsymbol{p}_x$ and $\boldsymbol{p}_z$, we need to decide if we should accept them and update the current step as well as the size of the trust region. To realize this idea, we can construct a merit function based on the objective function and constraints from the original barrier problem to decide the actual reduction. For example, a merit function can be expressed as

$$\phi_\nu \left(\boldsymbol{w}, \boldsymbol{z} \right) = f\left(\boldsymbol{w}\right) + \alpha R\left(\boldsymbol{w}\right) - \beta \sum_{i=1}^{2N_v} \ln\left(z_i\right) + \nu \| \boldsymbol{A}_{eq} \boldsymbol{w} - \mathbf{1} \|_2 + \nu \| \boldsymbol{A}_{ieq} \boldsymbol{w} + \boldsymbol{y}_{ieq} + \boldsymbol{z} \|_2, \tag{3.45}$$

where $\nu > 0$ is a penalty parameter. The actual reduction can be represented as

$$\text{ared}\left(\boldsymbol{p}\right) = \phi_\nu \left(\boldsymbol{w}, \boldsymbol{z} \right) - \phi_\nu \left(\boldsymbol{w} + \boldsymbol{p}_x, \boldsymbol{z} + \boldsymbol{p}_z \right). \tag{3.46}$$

The predicted reduction can be constructed in many ways, most of which are based on the SQP problem and its constraints. For example, we can set up a function as

$$\begin{aligned} q_\nu(\boldsymbol{p}) =& \nabla\left[f(\boldsymbol{w}) + \alpha R(\boldsymbol{w})\right]^T \boldsymbol{p}_x + \frac{1}{2}\boldsymbol{p}_x^T\left[\hat{H}(\boldsymbol{w}) + \alpha R(\boldsymbol{w})\right]\boldsymbol{p}_x - \beta \mathbf{1}^T \boldsymbol{Z}^{-1}\boldsymbol{p}_z \\ &+ \frac{1}{2}\boldsymbol{p}_z^T \boldsymbol{\Sigma}\boldsymbol{p}_z + \left\| \begin{bmatrix} \boldsymbol{A}_{eq}\boldsymbol{p}_x + \boldsymbol{A}_{eq}\boldsymbol{w} - \mathbf{1} \\ \boldsymbol{A}_{ieq}\boldsymbol{p}_x + \boldsymbol{p}_z + \boldsymbol{A}_{ieq}\boldsymbol{w} + \boldsymbol{y}_{ieq} + \boldsymbol{z} \end{bmatrix} \right\|_2 . \end{aligned} \tag{3.47}$$

For $q_\nu(\boldsymbol{p})$, the predicted reduction is the difference between not taking any step and taking the obtained step $\boldsymbol{p}$, which can be indicated as

$$\operatorname{pred}(\boldsymbol{p}) = q_\nu(\mathbf{0}) - q_\nu(\boldsymbol{p}), \tag{3.48}$$

where the variable $\boldsymbol{p}$ is a concatenation of $\boldsymbol{p}_x$ and $\boldsymbol{p}_z$. For a tiny constant $\eta = 10^{-8}$, if $\operatorname{ared}(\boldsymbol{p}) \geqslant \eta \operatorname{pred}(\boldsymbol{p})$, we accept $\boldsymbol{p}$ and update the current step. We will also update the trust region with a standard criterion based on the ratio $\operatorname{ared}(\boldsymbol{p})/\operatorname{pred}(\boldsymbol{p})$.

In conclusion, we implement this problem-specific nonlinear interior point trust region method to solve the corresponding optimization problem (3.33). Problem (3.33) is a nonlinear optimization problem under linear and bound constraints. The objective function contains a nonlinear log-likelihood term and a regularization term. For the log-likelihood term, we calculate the gradient and the modified Hessian as (3.20) and (3.25). For the regularization term, the total variation regularization is chosen to stabilize the solution. The modified Hessian is close to the true Hessian and it is positive semidefinite so solutions to the augmented Newton system are robust. Furthermore, the problem is prone to large-scale application since it is unnecessary to save the modified Hessian. We only need the Hessian-vector multiplication when we implement the Newton-CG method to solve the augmented Newton system. The cost of memory in each conjugate gradient iteration is close to an iteration of gradient descent.

3.4 Numerical Experiments

To test the method, we generate a 2D image of the size 128 by 128 and assume that the object is made up of three simulated materials that arise in polyenergetic image reconstruction – adipose, air and bones. For bones, we use the main component, calcium, to represent it. One application of polyenergetic image reconstruction is breast imaging, which requires low dose radiation for patients. To realize this application, we generate an energy spectrum with potential 26 keV with the help of function "spektrSpectrum" [53]. We also select a low radiation dose of 1e5 total photons for the x-ray energy spectrum. The corresponding spectrum is shown in Figure 3.1. From Figure 3.1, we can find that the photon flux density is above zero when the energy is between 3 kev and 28 kev. Based on this observation, the discrete energies for the simulated source x-ray beam are chosen from 3 keV to 28 keV, with an interval of 1 keV.

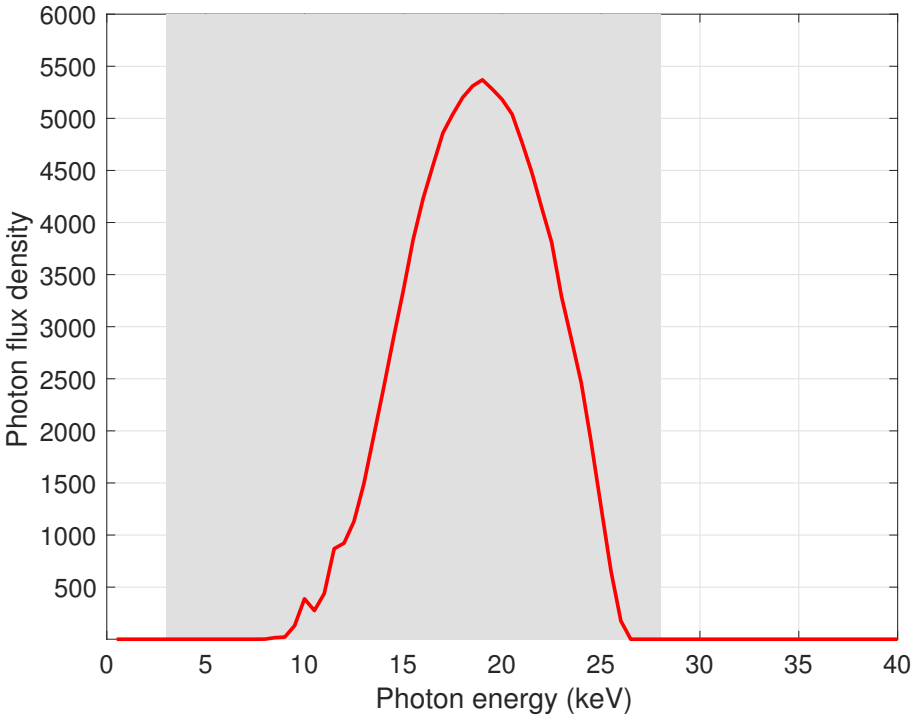

Figure 3.1: Photon flux density versus photon energy.

The plots of linear attenuation coefficients to materials adipose, air and calcium are shown in Figure 3.2. In Figure 3.2, the red, blue and black curves represent

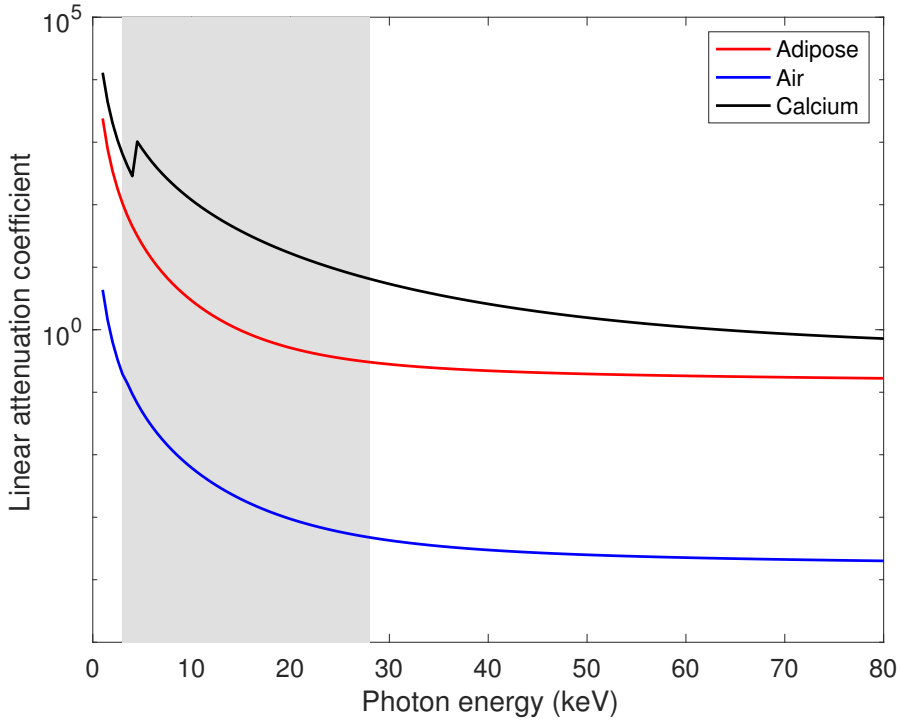

Figure 3.2: The linear attenuation curves for adipose (red), air (blue) and calcium (black).

adipose tissue, air and calcium, respectively, and the gray patch corresponds to the area of energy flux that is not equivalent to zero. From Figure 3.2, we can see that the curvatures of air and adipose are similar, while the curve of calcium has a K-edge [52]. The similarity of curvatures between adipose and air might cause the collinearity of linear attenuation coefficient matrix $\boldsymbol{C}$ and so as the ill-conditioning of Hessian, while the K-edge might result in difficulty for reconstruction.

The simulations of the true object, shown in Figure 3.3, contain four distinct regions: 100% adipose, 0% air, 0% calcium; 0% adipose, 100% air, 0% calcium; 0% adipose, 0% air, 100% calcium; 50% adipose, 50% air, 0% calcium[1]. In Figure 3.3, the

[1]We actually tested many different combinations of mixed materials, for example, 20% adipose, 60% air and 20% calcium. The results are are very similar to the one case considered in this experiment, thus to conserve space, we omit the results.

yellow color represents regions that contain 100% of the corresponding material, the turquoise color indicates regions that contain 50% of the adipose and air materials, the blue color indicates that the corresponding material does not exist in this area. Since we only have three materials, the weights corresponding to these three plots in Figure 3.3 should add to one. Moreover, since we use a 2-dimensional object for the

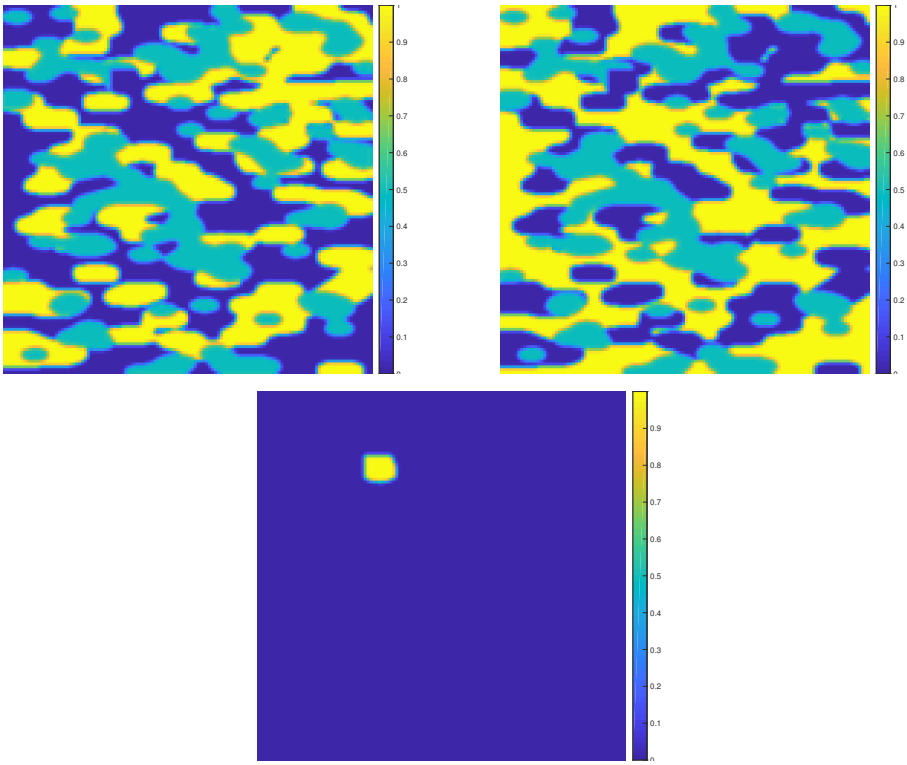

Figure 3.3: The true images for air (upper left), adipose (upper right) and calcium (middle). The turquoise colored regions are those areas in the object with a mixture of 50% glandular and 50% adipose tissue.

simulation, we use fan-beam (instead of cone beam) tomography model to generate a projection matrix $\boldsymbol{A}$ using the AIR Tools software [32]. The distance between the source and the detector is 70 cm, with 2.5 cm air gap between the object and the detector. In order to keep the gauge of projection matrix the same under different size of images, we scale the projection matrix by the grid size and the dimension of images. For example, we choose the grid size as 2 cm and the dimension as 128 pixels so the scaling results in a pixel size of 2/128 cm/pixel.

To avoid the inverse crime, we use spectral energies discretized on a finer grid and images with higher resolution to build the forward problem, but then use a coarser grid and lower resolution when solving the inverse problem. In particular, we collect the photon flux density corresponding to energies from 3 keV to 28 keV, with an interval of 0.5 keV for the forward problem, but then use an interval of 1 keV when we solve the inverse problem. Moreover, the resolution of object is initially 256×256 when we build the forward problem, and for the inverse problem, we solve (using a function included in the package IR Tools [22]) on a 128×128 grid. The number of x-rays used in building the ray trace matrix $\boldsymbol{A}$ is scaled to match the projected data generated with higher resolution images. Both full CT and limited angle reconstructions are presented in the following sections.

3.4.1 Full Angle Reconstruction

First, we only consider the full CT case, where the range of projection angle is from 0 to 179 degrees in one degree increment. We use Poisson distribution to generate the measurements as (3.10). The initial guess is a random vector whose entries are between 0 and 1 and it is not required to satisfy the equality constraint. Moreover, a total variation regularization is introduced to preserve the edges; specifically, we use the forward difference operator and zero boundary conditions for this regularization term. The regularization parameter is chosen among the set $\{10^{-1},\ 10^{-2},\ 10^{-3},\ 10^{-4},\ 10^{-5}\}$ and the most effective parameter is used to compute the result. With our (unoptimized) MATLAB implementation on a laptop computer, we need around 20 minutes to finish 30 Newton iterations where the stationary point is achieved. For the full CT case, the reconstructed images are presented in Figure 3.4. From Figure 3.4, we can see that the reconstructed images are of high quality in general. It successfully separates the areas corresponding to adipose, air and calcium as well as mixture of adipose and air. Edges of the reconstructed images are clear, which might be con-

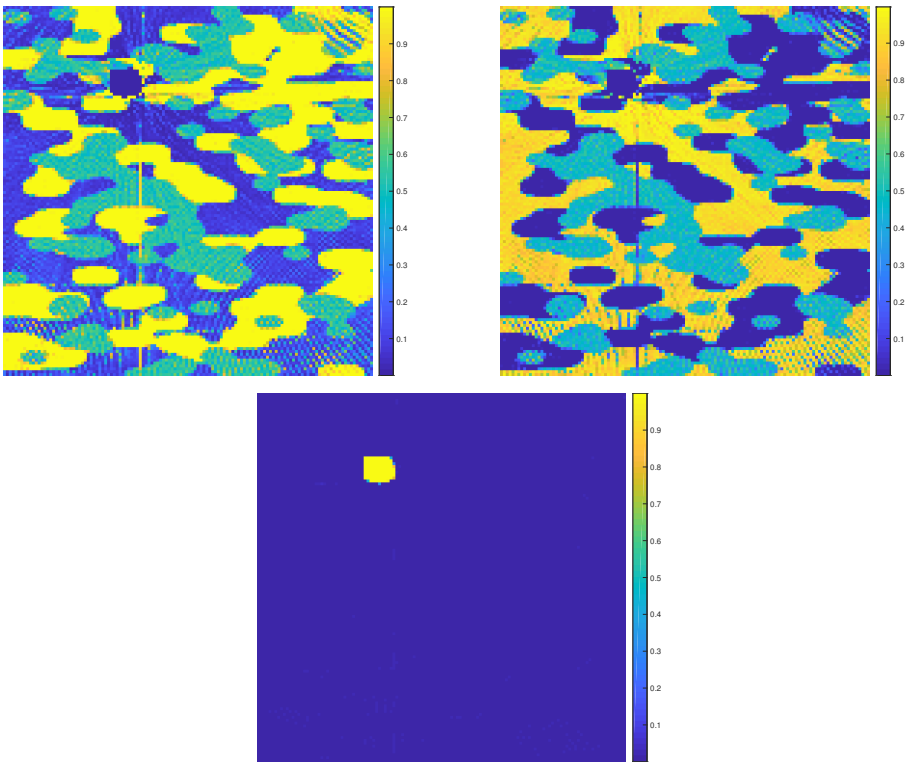

Figure 3.4: The reconstructed images for air (upper left), adipose (upper right) and calcium (middle) for the full CT reconstruction.

tributed from the total variation regularization. On the other hand, we can also find several artifacts that appear as blurred spots concentrating in the upper right corner, as well as other small artifacts scattered around the image. This results from measured data that are generated by Poisson distribution. With lower radiation dose, the relative noise level is higher compared with higher dose. Moreover, we can illustrate the convergence behavior by investigating the curves of relative errors. This plot is shown in Figure 3.5.

From Figure 3.5, we can find that the relative errors of materials air and adipose decrease in a similar way while the relative error of material calcium drops much faster. It is likely that calcium is only composed of a small part of the area and it can achieve faster convergence and higher accuracy. We also observe that the relative errors of three materials decrease to a particular level and then stagnate. The relative error between the last step and the true solution is about 19% for each material. Note

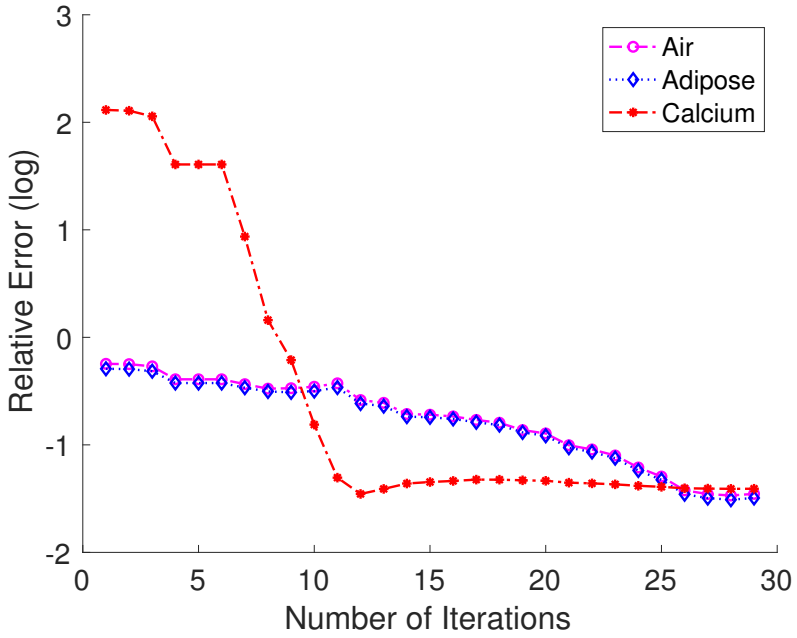

Figure 3.5: The plot of relative errors of air (magenta), adipose (blue) and calcium (red) for the full CT simulation.

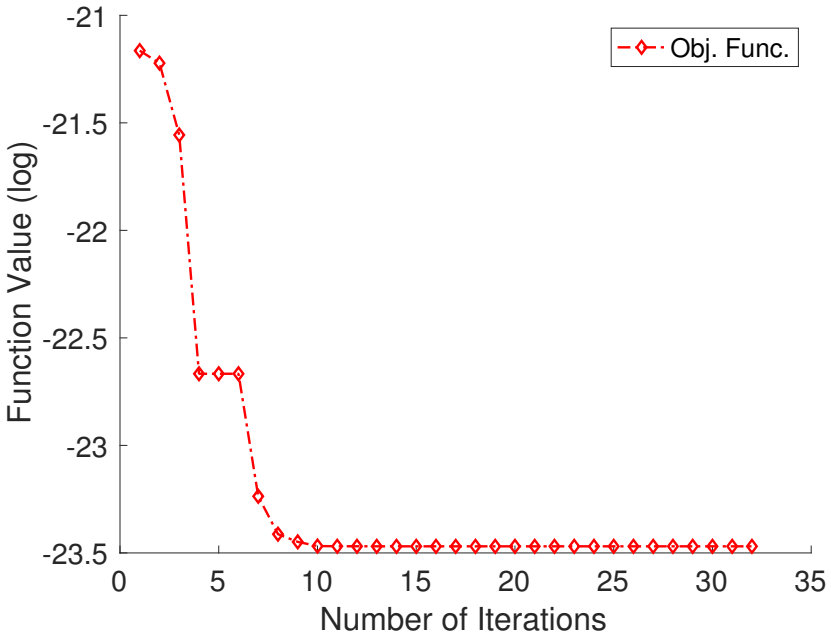

Figure 3.6: The plot of decrease of the objective function value.

that this stagnation occurs because of the regularization – without regularization, the relative errors may actually increase as the iterations proceed, which is a well-known behavior of ill-posed inverse problems, referred to as semi-convergence. We observe that when the relative errors stagnate, the current step approaches the first order optimality condition, which is an approximate solution to the KKT system.

To further validate convergence behavior of the proposed algorithm, we plot the curve of function value in Figure 3.6. From Figure 3.6, we can clearly identify that the function value drops fast in the beginning and then it stops for two iterations. After that, it starts to drop again and then stagnate. It cannot achieve lower value when reaching a specific level.

3.4.2 Limited Angle Reconstruction

In addition to the full CT case, it is important to also consider the case of limited angle reconstructions. Specifically, in the area of digital tomography, the limited angle reconstruction known as tomosynthesis has become an important diagnostic tool in breast imaging. The motivations for limited angle reconstruction are to reduce radiation dose to patients as well as to reduce the cost of this procedure. The limited angle reconstruction provides significantly more challenges to image reconstruction because the mathematical problems is much more ill-posed than the full CT case. This also means that the reconstruction quality is much more sensitive to the noise. In addition, the original objective function might have more stationary points and several of them are likely to satisfy the KKT condition. Under this situation, when the problem is effectively underdetermined, a poor initial guess may lead to an undesirable local minimum.

To test the limited angle reconstruction, the same test problem is used but with fewer projection angles. We shrink the projection angles from 180 degrees to 90 degrees, which ranges from 0 to 90 degrees. Furthermore, Poisson distribution is

used to generate the projection data. After implementing the previous algorithm, the reconstructed images are presented in Figure 3.7.

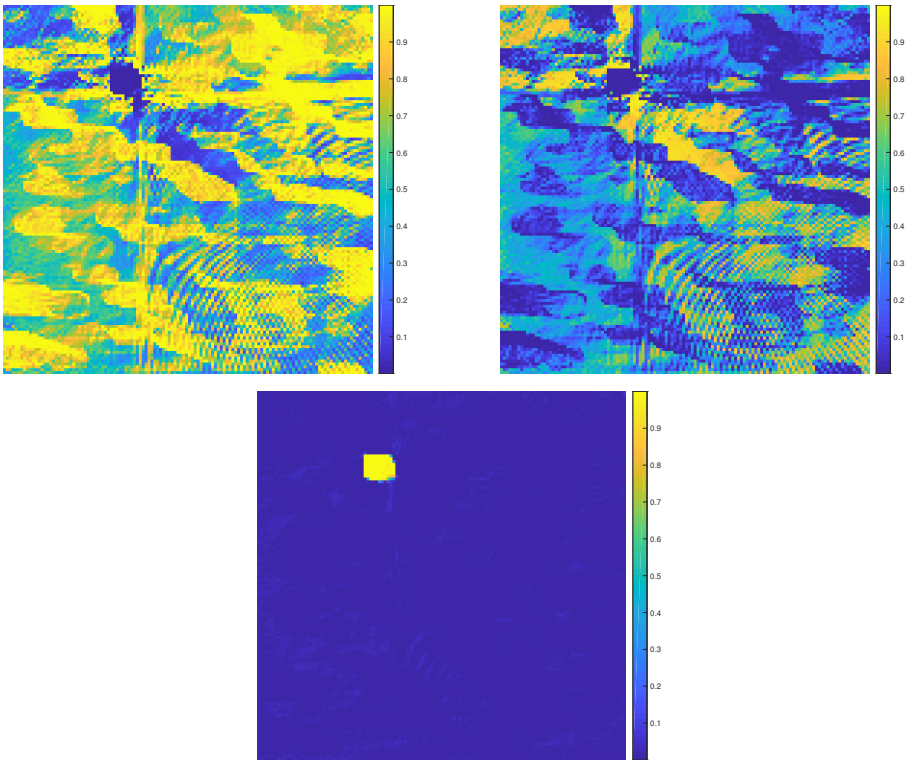

Figure 3.7: The reconstructed images for air (upper left), adipose (upper right) and calcium (middle) with 90 degrees projection.

From Figure 3.7, we can clearly see that the reconstructed images obtained from the limited angle case are more blurred than the images from the full CT case. As expected, with fewer projection angles, the images are of poorer quality. For the 90 degrees case, we can basically identify the distributions of materials roughly, while the details are more difficult to recognize. Only the material map of calcium is nearly fully separated from other materials. Moreover, we can see that the boundaries of different materials are not as clear as the boundaries in the full CT case. In several areas, the pixels are surrounded by shadows, which means that the materials are not completely separated. In the area of mixture, several pixels are colorful and the results depart slightly from the true solution. However, it is well known that due to the limited angle data, there are fundamental limitations when computing reconstructions [20].

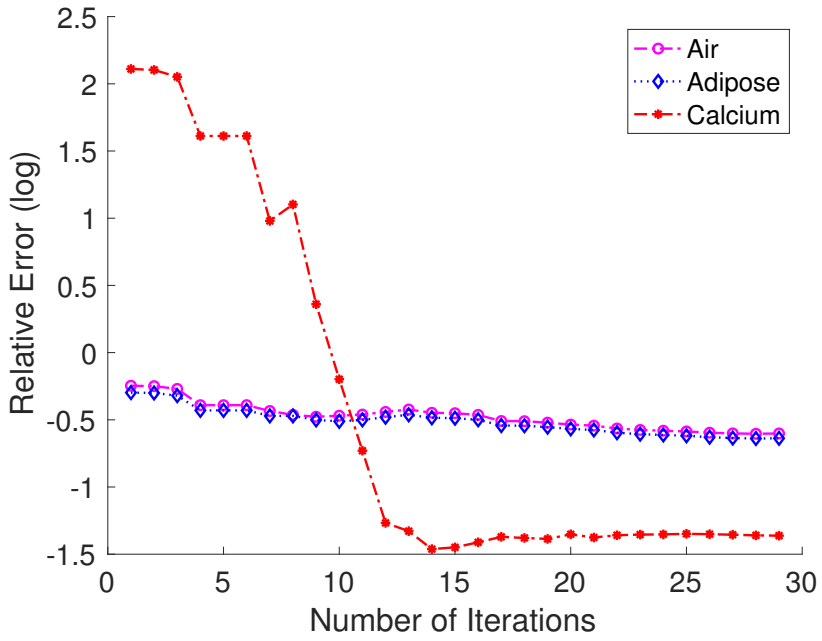

Figure 3.8: The plot of relative errors of air (magenta), adipose (blue) and calcium (red) for the 90 degrees limited angle simulation.

The plot of relative errors for 90 degrees case is presented in Figure 3.8. From Figure 3.8, we can find that the relative error curves corresponding to air and adipose decrease slowly and then stagnate. On the other hand, the relative error curve corresponding to calcium drops fast and converges to a lower level. This observation matches the phenomenon we conclude for the reconstructed images. Compared with the plot of full CT, the relative errors stagnate at higher levels. Moreover, we can find that the speed of convergence is not as rapid as the full CT case. However, we do observe that even if the regularization cannot completely compensate for the limited angle limitations, it does help to stabilize the solution.

3.5 Conclusions and Remarks

By taking multiple materials into consideration, the reconstructed images can reveal the weights of materials that compose the object, providing substantially more useful

information for the clinicians. Furthermore, the objective function and the gradient are uncomplicated to implement and the modified Hessian is a sufficient and stable estimate to the true Hessian. In addition, the merits of using a nonlinear interior point method are easy to identify. It is a globally convergent method with superlinear rate of convergence. It is also a stable and robust algorithm that can handle large-scale problems. Furthermore, there is substantial flexibility in choosing the initial guess because it does not need to satisfy the constraints.

Although this method has advantages such as faster convergence, robust computation and flexibility, it still has a few limitations. For example, implementation of nonlinear interior point method is not straightforward for large-scale problems. It requires solving a normal subproblem as well as a tangential subproblem. Furthermore, we need to decide the size of trust region in each iteration. Meanwhile, this method involves many parameters that we need to choose manually. So far, we have only tested 2D images rather than 3D images. For 3D images, the evaluation of each part might be more complicated, which is likely to increase the expense for solving this problem. For further research, we might consider the gradient-based methods such as the scaled gradient descent method or splitting methods such as the alternating direction method of multipliers (ADMM) [21].

Chapter 4

Nonlinear Optimization for Energy-windowed Spectral Computed Tomography

The development of new energy-windowed spectral computed tomography (CT) machines have received a great deal of interest in recent years; see, e.g. [2, 57]. 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 [45] and improve quality of reconstructed images. To reconstruct the material maps of an object, we need to solve a nonlinear equation of the form

$$\boldsymbol{Y} = \exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}, \tag{4.1}$$

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. Moreover, $\boldsymbol{\mathcal{E}}$ represents the noise term and we assume that $y_{il} \sim \text{Poisson}\left(\left[\exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{S}\right]_{il}\right)$ for each component y_{il} in $\boldsymbol{Y}$ and the corresponding entry in the exponential term. We also assume that these data are known and the target is to solve the unknown weight matrix $\boldsymbol{W}$. $\boldsymbol{W}$ is of 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, it must be nonnegative and we need to add a lower bound constraint $\boldsymbol{W} \geqslant \boldsymbol{0}$.

To obtain the quantitative information about the material composition, we need to solve a nonlinear inverse problem. This nonlinear inverse problem is extremely ill-posed and direct implementation of regular methods is not effective. In this case, Barber et al. [2] suggests a preconditioner that is based on the eigenvalue decomposition of the matrix of linear attenuation coefficients, $\boldsymbol{C}^T\boldsymbol{C}$. With this preconditioner, they use the *Chambolle-Pock* (CP) primal-dual algorithm [10] to solve the inverse problem with Poisson log-likelihood loss function. However, they construct the preconditioner with only linear attenuation coefficients and do not include energy information. Moreover, the Poisson log-likelihood function contains multiple exponential terms and it is hard to handle. Because of these problems, we propose a new preconditioner that both involves linear attenuation coefficients and energy spectrum information based on a rank-1 approximation. To implement this preconditioner, we use a two-step method that includes finding an approximate Cauchy point in the first step and solving a quadratic programming problem in the second step.

This chapter is organized as follows. We review the energy-windowed spectral CT model in Section 4.1. Because of the challenges raised by this model, a new

preconditioning framework is introduced in Section 4.2. In order to implement this preconditioner, we suggest in Section 4.3 an optimization algorithm based on the projected line search and the trust region method. In Section 4.4 we illustrate the strength of the new preconditioner and the algorithm using numerical experiments. Finally, comments, limitations and future work are provided in Section 4.5.

4.1 The Energy-windowed Spectral CT Model

In computed tomography (CT), source x-ray beams are composed of a spectrum of different energies [7]. 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 of energy-windowed spectral CT is expressed by

$$y_i^{(k)} = \int_E S^{(k)}(e) \exp\left(- \int_{t \in l} \mu\left(\vec{r}\,(t)\,, e\right) \mathrm{d}\, t \right) \mathrm{d}\, e + \eta_i^{(k)}, \quad \begin{cases} i = 1, 2, \cdots, N_d \times N_p, \\ k = 1, 2, \cdots, N_b, \end{cases} \tag{4.2}$$

where

- $y_i^{(k)}$ is the x-ray intensity of the i-th pixel in the k-th detector bin.
- E is the photon flux density. Figure 4.4 shows a curve of E versus photon energy with relative high potential (120 keV).
- N_d is the number of detector pixels. For a material map of the size n by n, we assume $N_d = n$ and the number of projection rays for each angle is also equivalent to N_d.

- N_p is the number of projections. For cone/fan beam CT, projections are distributed equally from 0 to 360 degrees.

- N_b is the number of detector bins (windows). For an energy-windowed CT machine, we usually assume that it has 5 to 6 energy bins.

- $S^{(k)}(e)$ represents the photon flux density for the k-th detector bin, which is the number of incident photons at the energy level e in the k-th energy window.

- $\mu\left(\vec{r}\left(t\right), e\right)$ denotes the linear attenuation coefficient that is related to the position function $\vec{r}\left(t\right)$ and energy level e.

- $\eta_i^{(k)}$ is the error term for the i-th element in k-th energy bin and it is assumed to be Gaussian for this model.

With the introduction of energy window, we need to use Beer's law k times to express the corresponding equation. Compared with Equation (3.2) in Chapter 3, this discrete model has multiple columns corresponding to the projected data of specified detector bins so it can be expressed in a form of matrix equation. However, we can still expand the unknown linear attenuation coefficient $\mu\left(\vec{r}\left(t\right), e\right)$ into a summation of multiplication of $u_{m,e}$, the linear attenuation coefficient for the m-th material at the energy level e, and $w_m\left(\vec{r}\right)$, the unknown weight of the m-th material at the position $\vec{r}$:

$$\mu\left(\vec{r}\left(t\right), e\right) = \sum_{m=1}^{N_m} u_{m,e} w_m\left(\vec{r}\right), \tag{4.3}$$

where

- N_m is the number of materials that form the object.

Again, we can shrink the size of unknown variable with Expansion (4.3). The size of the unknown variable is made up of two dimensions and one dimension, the resolution of material map, remain the same but the other dimension, the number of

discrete energies, has been reduced to the number of materials. Usually, the number of discrete energies can be hundreds but the number of materials are 2 or 3 so the dimension of new solution space is significantly decreased. As we have seen in Chapter 3, the weight fraction, $w_m(\vec{r})$, can be represented as a summation of product of the weight $w_{j,m}$ and the basis function $\phi_j(\vec{r})$

$$w_m(\vec{r}) = \sum_{j=1}^{N_v} w_{j,m}\phi_j(\vec{r}), \tag{4.4}$$

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).
- $\phi_j(\vec{r})$ 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 number of projections N_p:

$$a_{i,j} = \int_{t\in l} \phi_j(\vec{r}(t))\,\mathrm{d}\,t. \tag{4.5}$$

With (4.4) and (4.5), the unknown linear attenuation coefficients can be represented as

$$\int_{t\in l} \mu(\vec{r}(t), e)\,\mathrm{d}\,t = \sum_{m=1}^{N_m}\sum_{j=1}^{N_v} u_{m,e}w_{j,m}\int_{t\in l}\phi_j(\vec{r}(t))\,\mathrm{d}\,t = \sum_{j=1}^{N_v}\sum_{m=1}^{N_m} a_{i,j}w_{j,m}u_{m,e}. \tag{4.6}$$

By ignoring quadrature errors and discretizing the integral with respect to the energy E, the discretization of the basic model (4.2) can be written as:

$$y_i^{(k)} = \sum_{e=1}^{N_e} s_e^{(k)} \exp\left(-\sum_{j=1}^{N_v}\sum_{m=1}^{N_m} a_{i,j}w_{j,m}u_{m,e}\right) + \eta_i^{(k)}, \tag{4.7}$$

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

$$\boldsymbol{Y} = \exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}, \tag{4.8}$$

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 m-th material at the energy level e. For similar materials, we expect their linear attenuation coefficients to be similar so it might introduce the collinearity of the matrix $\boldsymbol{C}$.
- $\boldsymbol{S}$ is a matrix of the size $N_e \times N_b$ and each column collects the spectrum energy of a specific range.
- $\boldsymbol{\mathcal{E}}$ is the noise matrix that is of the size $(N_d \cdot N_p) \times N_b$. The assumption for noise is that $y_{il} \sim \text{Poisson}\left(\left[\exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{S}\right]_{il}\right)$ for y_{il} in $\boldsymbol{Y}$.

In Equation (4.8), the exponential function is point-wise rather than matrix function. In addition to Equation (4.8), we also require that weight fractions should be non-negative and this can be illustrated by the constraint $\boldsymbol{W} \geqslant \boldsymbol{0}$. Recall that in Chapter 3, we also require that the weight fractions in each voxel should add to 1. That is to say, $\sum_{m=1}^{N_m} w_{j,m} = 1$ for $j = 1, 2, \cdots, N_v$. For the energy-windowed spectral CT model,

we drop this normalization requirement and allow unnormalized weights. So the only constraint is $\boldsymbol{W} \geqslant \mathbf{0}$.

Even if the constraints are simplified, we still do not want to handle the matrix form of this equation directly since the second order derivative of the objective function will introduce tensors. To avoid introducing tensors, we need to vectorize Equation (4.8) on both sides. By taking vectorization and using the properties of Kronecker product, we can transform Equation (4.8) into

$$\operatorname{vec}(\boldsymbol{Y}) = \left(\boldsymbol{S}^T \otimes \boldsymbol{I}\right) \exp\left\{-\left(\boldsymbol{C} \otimes \boldsymbol{A}\right) \operatorname{vec}(\boldsymbol{W})\right\} + \operatorname{vec}(\boldsymbol{\mathcal{E}}), \tag{4.9}$$

where $\boldsymbol{I}$ is the identity matrix of the size $N_d \cdot N_p$ by $N_d \cdot N_p$. To simplify the notations, we let $\boldsymbol{y} = \operatorname{vec}(\boldsymbol{Y})$, $\boldsymbol{w} = \operatorname{vec}(\boldsymbol{W})$ and $\boldsymbol{\eta} = \operatorname{vec}(\boldsymbol{\mathcal{E}})$. Then Equation (4.9) can be rewritten as

$$\boldsymbol{y} = \left(\boldsymbol{S}^T \otimes \boldsymbol{I}\right) \exp\left\{-\left(\boldsymbol{C} \otimes \boldsymbol{A}\right) \boldsymbol{w}\right\} + \boldsymbol{\eta}. \tag{4.10}$$

In addition to Equation (4.10), we also require that the weight fractions should be bounded blow by zero. So we introduce the constraint $\boldsymbol{w} \geqslant \mathbf{0}$.

4.2 Problem Set-up and Preconditioning

4.2.1 The Constrained Least Squares Problem

Based on Equation (4.10) and the nonnegative constraint $\boldsymbol{w} \geqslant \mathbf{0}$, we can formulate a constrained nonlinear least squares problem:

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2} \left\| \boldsymbol{y} - \left(\boldsymbol{S}^T \otimes \boldsymbol{I}\right) \exp\left\{-\left(\boldsymbol{C} \otimes \boldsymbol{A}\right) \boldsymbol{w}\right\} \right\|_2^2 \\ \text{subject to} \quad & \boldsymbol{w} \geqslant \mathbf{0}. \end{aligned} \tag{4.11}$$

With the least squares loss function, we want to use the Gauss-Newton method to solve this problem. If we assume the residual is $r(\boldsymbol{w})$, then $r(\boldsymbol{w})$ can be represented as:

$$r(\boldsymbol{w}) = \boldsymbol{y} - \left(\boldsymbol{S}^T \otimes \boldsymbol{I}\right) \exp\left\{-\left(\boldsymbol{C} \otimes \boldsymbol{A}\right) \boldsymbol{w}\right\}. \tag{4.12}$$

So the Jacobian can be calculated as

$$\boldsymbol{J}(\boldsymbol{w}) = \nabla \boldsymbol{r}(\boldsymbol{w})^T = \left(\boldsymbol{S}^T \otimes \boldsymbol{I}\right) \operatorname{diag}\left(\exp\left\{-\left(\boldsymbol{C} \otimes \boldsymbol{A}\right) \boldsymbol{w}\right\}\right) \left(\boldsymbol{C} \otimes \boldsymbol{A}\right). \tag{4.13}$$

With the Jacobian and residual, we can represent the gradient in terms of these two terms:

$$\nabla f(\boldsymbol{w}) = \boldsymbol{J}(\boldsymbol{w})^T \boldsymbol{r}(\boldsymbol{w}). \tag{4.14}$$

The Gauss-Newton approximation of Hessian can be expressed in terms of $\boldsymbol{J}(\boldsymbol{w})$:

$$\boldsymbol{H}(\boldsymbol{w}) = \boldsymbol{J}(\boldsymbol{w})^T \boldsymbol{J}(\boldsymbol{w}) = \left(\boldsymbol{C}^T \otimes \boldsymbol{A}^T\right) \boldsymbol{D} \left(\boldsymbol{S}\boldsymbol{S}^T \otimes \boldsymbol{I}\right) \boldsymbol{D} \left(\boldsymbol{C} \otimes \boldsymbol{A}\right), \tag{4.15}$$

where $\boldsymbol{D} = \operatorname{diag}\left(\exp\left\{-\left(\boldsymbol{C} \otimes \boldsymbol{A}\right) \boldsymbol{w}\right\}\right)$. In the k-th iteration, we need to solve a Gauss-Newton system for the step $\boldsymbol{p}_k$:

$$\boldsymbol{H}(\boldsymbol{w}_k) \boldsymbol{p}_k = -\nabla f(\boldsymbol{w}_k). \tag{4.16}$$

The naive way to solve this problem is based on two steps. At first, we solve the Gauss-Newton system to obtain step $\boldsymbol{p}_k$ in each iteration. Then we update the current step and project this step onto boundary $\boldsymbol{w} \geqslant \boldsymbol{0}$. However, this Gauss-Newton method with simple projections does not guarantee convergence [37]. Furthermore, the sparsity of the matrix $\boldsymbol{A}$ and the possible collinearity of the matrix $\boldsymbol{C}$ might contribute to the ill-conditioning of the matrix $\boldsymbol{H}(\boldsymbol{w})$. Therefore, direct implementation of Gauss-Newton method is ineffective and does not produce satisfactory results in

reality. Because of these problems, we think about using preconditioners and try to apply another optimization framework to solve this problem. For this new optimization framework, we will solve the same constrained least squares problem and use the same Gauss-Newton approximation.

4.2.2 Preconditioning of the Hessian

Since the Hessian $\boldsymbol{H}(\boldsymbol{w})$ is extremely ill-conditioned, the corresponding constrained least squares problem is hard to solve. To overcome this difficulty, we want to add preconditioners to the Hessian matrix $\boldsymbol{H}(\boldsymbol{w})$. It is easy to see that the Hessian matrix $\boldsymbol{H}(\boldsymbol{w})$ is a product of several matrices, and among these matrices, it is hard to modify either $\boldsymbol{C} \otimes \boldsymbol{A}$ or $\boldsymbol{S}\boldsymbol{S}^T \otimes \boldsymbol{I}$. On the other hand, $\boldsymbol{D}$ is a diagonal matrix so it might be convenient to construct preconditioners based on this matrix. If we can decompose the matrix $\boldsymbol{D}$ into a Kronecker product of two matrices, then we can use the properties of Kronecker product to combine the left and right terms. With this idea, we try to decompose the matrix $\boldsymbol{D}$ into a Kronecker product of two diagonal matrices, $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$, where $\boldsymbol{D}_1$ is of the size N_e by N_e and $\boldsymbol{D}_2$ is of the size $N_d \cdot N_p$ by $N_d \cdot N_p$:

$$\boldsymbol{D} \approx \boldsymbol{D}_1 \otimes \boldsymbol{D}_2. \tag{4.17}$$

Moreover, $\boldsymbol{D}_1 \otimes \boldsymbol{D}_2$ is chosen to minimize the distance to $\boldsymbol{D}$ with respect to the Frobenius norm:

$$\min_{\boldsymbol{D}_1, \boldsymbol{D}_2} \quad \|\boldsymbol{D} - \boldsymbol{D}_1 \otimes \boldsymbol{D}_2\|_F \tag{4.18}$$

This is a nearest Kronecker product (NKP) problem [55] and the solution has already been studied extensively. Since we require that $\boldsymbol{D}$, $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are diagonal matrices, it is equivalent to minimizing the distance on behalf of their diagonal entries. In this case, we let $\boldsymbol{D} = \operatorname{diag}\{\boldsymbol{d}\}$, $\boldsymbol{D}_1 = \operatorname{diag}\{\boldsymbol{d}_1\}$ and $\boldsymbol{D}_2 = \operatorname{diag}\{\boldsymbol{d}_2\}$, where $\boldsymbol{d}$, $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ are diagonal entries of the matrices $\boldsymbol{D}$, $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$, respectively. So Problem (4.18)

is equivalent to a least squares problem that is based on their diagonal elements with respect to the 2-norm:

$$\min_{\boldsymbol{d}_1,\boldsymbol{d}_2} \quad \|\boldsymbol{d}-\boldsymbol{d}_1\otimes\boldsymbol{d}_2\|_2 \tag{4.19}$$

It is easy to see that $\boldsymbol{d}$, $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ are vectors, which are also rank-1 matrices. In this case, the goal of this problem is to find two rank-1 matrices that will minimize the distance with respect to the 2-norm. The result is given by the largest singular value and its corresponding singular vectors. By using the singular value decomposition (SVD), one solution to this NKP problem is $\boldsymbol{d}_1 = \sqrt{\sigma_1}\boldsymbol{v}_1$ and $\boldsymbol{d}_2 = \sqrt{\sigma_1}\boldsymbol{u}_1$, where σ_1 is the largest singular value and $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ are the first left and right singular vectors of the SVD of the matrix $\tilde{\boldsymbol{D}} = \text{reshape}\,(\boldsymbol{d}, N_d\cdot N_p, N_e)$. In practice, we use an efficient MATLAB package, "PROPACK" [39], to compute the largest singular value and the corresponding singular vectors.

After we have obtained $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$, we can estimate $\boldsymbol{H}\,(\boldsymbol{w})$ using the properties of Kronecker product:

$$\begin{aligned}\boldsymbol{H}\,(\boldsymbol{w}) &\approx \left(\boldsymbol{C}^T\otimes\boldsymbol{A}^T\right)(\boldsymbol{D}_1\otimes\boldsymbol{D}_2)\left(\boldsymbol{S}\boldsymbol{S}^T\otimes\boldsymbol{I}\right)(\boldsymbol{D}_1\otimes\boldsymbol{D}_2)(\boldsymbol{C}\otimes\boldsymbol{A})\\ &= \left(\boldsymbol{C}^T\boldsymbol{D}_1\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{D}_1\boldsymbol{C}\right)\otimes\left(\boldsymbol{A}^T\boldsymbol{D}_2\boldsymbol{D}_2\boldsymbol{A}\right).\end{aligned} \tag{4.20}$$

The size of the matrix $\boldsymbol{C}$ is N_e by N_m and N_m is the number of materials. In practice, we usually consider only 2 or 3 materials as the composition of the object. Therefore, the size of the matrix product, $\boldsymbol{C}^T\boldsymbol{D}_1\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{D}_1\boldsymbol{C}$, is usually 2 by 2 or 3 by 3. Using this approximation, we can use the preconditioners to transform this term into identity so the new matrix should not depend on this part and thus be better-conditioned. To be specific, we let $\boldsymbol{M}_1$ be the preconditioner of the size $N_d\cdot N_p$ by $N_d\cdot N_p$ and $\boldsymbol{M}_2$ be the preconditioner of the size N_e by N_e. Then the preconditioned Hessian can be

represented as:

$$\begin{aligned}\tilde{\boldsymbol{H}}(\boldsymbol{w}) &= \left(\boldsymbol{M}_2^T \otimes \boldsymbol{M}_1^T\right) \boldsymbol{H}(\boldsymbol{w}) \left(\boldsymbol{M}_2 \otimes \boldsymbol{M}_1\right) \\ &\approx \left(\boldsymbol{M}_2^T \otimes \boldsymbol{M}_1^T\right) \left(\left(\boldsymbol{C}^T \boldsymbol{D}_1 \boldsymbol{S}\boldsymbol{S}^T \boldsymbol{D}_1 \boldsymbol{C}\right) \otimes \left(\boldsymbol{A}^T \boldsymbol{D}_2 \boldsymbol{D}_2 \boldsymbol{A}\right)\right) \left(\boldsymbol{M}_2 \otimes \boldsymbol{M}_1\right) \\ &= \left(\boldsymbol{M}_2^T \boldsymbol{C}^T \boldsymbol{D}_1 \boldsymbol{S}\boldsymbol{S}^T \boldsymbol{D}_1 \boldsymbol{C} \boldsymbol{M}_2\right) \otimes \left(\boldsymbol{M}_1^T \boldsymbol{A}^T \boldsymbol{D}_2 \boldsymbol{D}_2 \boldsymbol{A} \boldsymbol{M}_1\right).\end{aligned} \tag{4.21}$$

Since we can choose $\boldsymbol{D}_1$ to guarantee $\boldsymbol{C}^T \boldsymbol{D}_1 \boldsymbol{S}\boldsymbol{S}^T \boldsymbol{D}_1 \boldsymbol{C}$ to be symmetric positive definite, then we can calculate the Cholesky decomposition to this matrix as

$$\boldsymbol{C}^T \boldsymbol{D}_1 \boldsymbol{S}\boldsymbol{S}^T \boldsymbol{D}_1 \boldsymbol{C} = \boldsymbol{G}^T \boldsymbol{G}, \tag{4.22}$$

where $\boldsymbol{G}$ is an upper triangular matrix with positive diagonal entries. In this case, we can choose $\boldsymbol{M}_2 = \boldsymbol{G}^{-1}$ and the first part in $\tilde{\boldsymbol{H}}(\boldsymbol{w})$ has become identity. On the other hand, $\boldsymbol{A}^T \boldsymbol{D}_2 \boldsymbol{D}_2 \boldsymbol{A}$ is still large-scale because of the dimension of $\boldsymbol{A}$. It is challenging to find a preconditioner $\boldsymbol{M}_1$ that best fits all conditions. In this situation, we might choose $\boldsymbol{M}_1$ to be a diagonal matrix such that each column of the matrix $\boldsymbol{M}_1^T \boldsymbol{A}^T \boldsymbol{D}_2 \boldsymbol{D}_2 \boldsymbol{A} \boldsymbol{M}_1$ is scaled to 1 with respect to either 1-norm or 2-norm. Or we can even choose it to be identity so that we do not add any preconditioners to this part.

Using the SVD, we can analyze the condition number before and after preconditioning. Before preconditioning, the matrix $\boldsymbol{H}(\boldsymbol{w})$ depends on two parts, $\boldsymbol{C}^T \boldsymbol{D}_1 \boldsymbol{S}\boldsymbol{S}^T \boldsymbol{D}_1 \boldsymbol{C}$ and $\boldsymbol{A}^T \boldsymbol{D}_2 \boldsymbol{D}_2 \boldsymbol{A}$. If we assume that the singular value decompositions of these two matrices are $\boldsymbol{C}^T \boldsymbol{D}_1 \boldsymbol{S}\boldsymbol{S}^T \boldsymbol{D}_1 \boldsymbol{C} = \boldsymbol{U}_1 \boldsymbol{\Sigma}_1 \boldsymbol{V}_1^T$ and $\boldsymbol{A}^T \boldsymbol{D}_2 \boldsymbol{D}_2 \boldsymbol{A} = \boldsymbol{U}_2 \boldsymbol{\Sigma}_2 \boldsymbol{V}_2^T$, then the condition number of $\boldsymbol{H}(\boldsymbol{w})$ is closely related to the diagonal entries of $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$. Let the largest and smallest diagonal elements of $\boldsymbol{\Sigma}_1$ be $\sigma_{1\max}$ and $\sigma_{1\min}$. In addition, let the largest and smallest diagonal elements of $\boldsymbol{\Sigma}_2$ be $\sigma_{2\max}$ and $\sigma_{2\min}$, then the condition number of $\boldsymbol{H}(\boldsymbol{w})$ can be estimated as

$$\kappa\left(\boldsymbol{H}(\boldsymbol{w})\right) \approx \frac{\sigma_{1\max}\sigma_{2\max}}{\sigma_{1\min}\sigma_{2\min}}, \tag{4.23}$$

where κ is the condition number. In contrast, the condition number of $\tilde{\boldsymbol{H}}(\boldsymbol{w})$ is mostly dependent on $\boldsymbol{A}^T\boldsymbol{D}_2\boldsymbol{D}_2\boldsymbol{A}$ after preconditioning, which is related to the largest and smallest diagonal entries of $\boldsymbol{\Sigma}_2$:

$$\kappa\left(\tilde{\boldsymbol{H}}(\boldsymbol{w})\right) \approx \frac{\sigma_{2\max}}{\sigma_{2\min}}. \tag{4.24}$$

In most cases, $\sigma_{1\max}/\sigma_{1\min}$ is not close to 1 and $\boldsymbol{M}_1 = \boldsymbol{I}$, then we can conclude that

$$\kappa(\tilde{\boldsymbol{H}}(\boldsymbol{w})) \ll \kappa(\boldsymbol{H}(\boldsymbol{w})). \tag{4.25}$$

In practice, the reduction of condition number can be at least two orders of magnitude. In addition to the condition number, we can check if the eigenvalues are more clustered after preconditioning. With more clustered eigenvalues, optimization methods such as conjugate gradient (CG) or generalized minimal residual method (GMRES) [51] will converge with faster speed. To check the eigenvalues before and after preconditioning, we construct an object of two materials and the material map corresponding to each material is of the size 16 by 16. Therefore, the resulting Hessian matrices, $\boldsymbol{H}(\boldsymbol{w})$ and $\tilde{\boldsymbol{H}}(\boldsymbol{w})$, are of the size 512 by 512. The plot of eigenvalues of these two matrices is presented in Figure 4.1. In Figure 4.1, we take the logarithm of eigenvalues to compare the clusters. From this plot, we can easily see that the eigenvalues of the original Hessian $\boldsymbol{H}(\boldsymbol{w})$ are scattered in a larger span, while the eigenvalues of the preconditioned Hessian $\tilde{\boldsymbol{H}}(\boldsymbol{w})$ are clustered within a smaller range. Usually, the clustered eigenvalues provide favorable convergence speed. Since the preconditioned Hessian, $\tilde{\boldsymbol{H}}(\boldsymbol{w})$, relies on $\boldsymbol{w}$, we need to compute the preconditioners, $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$, in each iteration. However, we only need the largest singular value and the corresponding singular vectors, which is very cheap.

Even if we have obtained the new preconditioners, it is still not straightforward about how to implement it for the optimization problem. If we follow the steps in

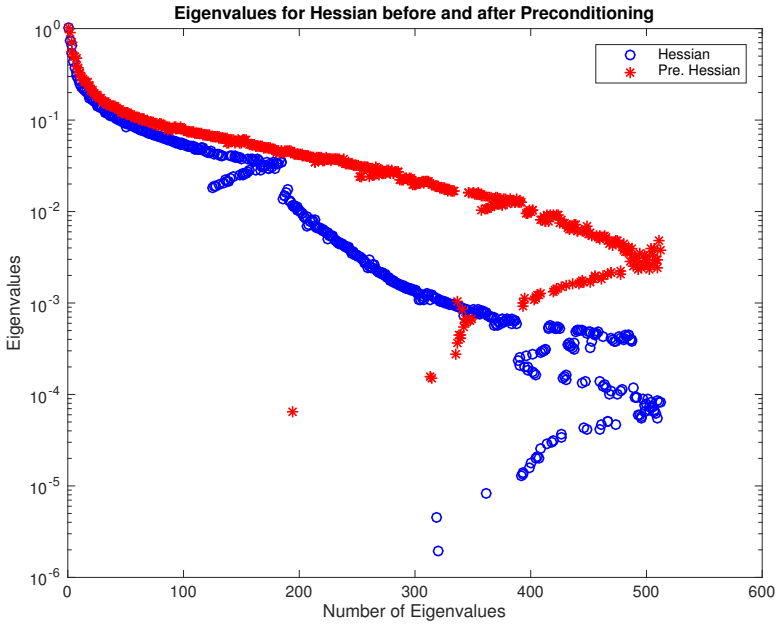

Figure 4.1: The comparison of eigenvalues before and after preconditioning (with scaling).

Section 4.2 directly, we need to solve a preconditioned system:

$$\left(\boldsymbol{M}_2^T \otimes \boldsymbol{M}_1^T\right) \boldsymbol{H}\left(\boldsymbol{w}_k\right)\left(\boldsymbol{M}_2 \otimes \boldsymbol{M}_1\right)\left(\boldsymbol{M}_2^{-1} \otimes \boldsymbol{M}_1^{-1}\right) \boldsymbol{p}_k=-\left(\boldsymbol{M}_2^T \otimes \boldsymbol{M}_1^T\right) \nabla f\left(\boldsymbol{w}_k\right). \tag{4.26}$$

If we let $\tilde{\boldsymbol{p}}_k = \left(\boldsymbol{M}_2^{-1} \otimes \boldsymbol{M}_1^{-1}\right) \boldsymbol{p}_k$ and $\nabla \tilde{f}\left(\boldsymbol{w}_k\right) = \left(\boldsymbol{M}_2^T \otimes \boldsymbol{M}_1^T\right) \nabla f\left(\boldsymbol{w}_k\right)$, then this system can be rewritten as

$$\tilde{\boldsymbol{H}}\left(\boldsymbol{w}_k\right) \tilde{\boldsymbol{p}}_k=-\nabla \tilde{f}\left(\boldsymbol{w}_k\right). \tag{4.27}$$

After solving this system, we need to retrieve the step $\boldsymbol{p}_k$ from $\tilde{\boldsymbol{p}}_k$ and tailor the step to meet the nonnegative constraint $\boldsymbol{w} \geqslant \mathbf{0}$. In practice, if we update the current step and project the new step onto the boundary, it is not efficient and unable to guarantee convergence. Therefore, how to implement preconditioners efficiently under the constraint is worth considering.

4.3 Optimization and Regularization

4.3.1 Optimization with the Proposed Preconditioner

With the proposed preconditioners, we can implement them directly into Equation (4.16) and project the new step onto the boundary. However, numerical experiments are not in favor of this method. Without constraints, it is likely that the step obtained by this method can offer sufficient reduction. After we project the new step onto the boundary, it might not be a feasible choice anymore. Furthermore, we cannot guarantee convergence even if we add a line search scheme. To improve convergence speed and further stabilize solution, we apply the preconditioners into a two-step method. In the first step, we use the projected line search method to find the approximate Cauchy point. In the second step, we fix the elements of this Cauchy point that are active on the boundary and minimize the objective function with points that are inactive. In both steps, we include trust regions in terms of the infinite norm to further restrict the step size. The idea is that the step obtained using this method should be at least better than the approximate Cauchy point.

In the first step, we use a projected line search method to find the approximate Cauchy point. The basic idea is to find the point that will both satisfy the Wolfe condition [56] and the trust region constraint:

$$\begin{aligned} f\left(\boldsymbol{w}_k\left(\alpha\right)\right) &\leqslant f\left(\boldsymbol{w}_k\right) + c_1\alpha\nabla f_k^T\left(\boldsymbol{w}_k\left(\alpha\right) - \boldsymbol{w}_k\right), \\ \left\|\boldsymbol{w}_k\left(\alpha\right) - \boldsymbol{w}_k\right\|_\infty &\leqslant c_2\Delta_k, \end{aligned} \tag{4.28}$$

where $\boldsymbol{w}_k\left(\alpha\right) = P\left(\boldsymbol{w}_k - \alpha\nabla f_k\right)$ and P is the operator that is used to project $\boldsymbol{w}$ onto the boundary. Δ_k is the size of trust region for the k-th step. c_1, c_2 are constants and c_1, $c_2 \in (0,\ 1)$. c_2 is chosen to be close to 1 to guarantee the trust region in the first step is smaller, but also close, to the trust region in the second step. Details are presented in Algorithm 1.

Algorithm 1 Projected Line Search

Initialization:
$c_1 = 0.1$; $c_2 = 0.8$; $\alpha = \beta = 0.5$;
Set up the maximum number of iterations: maxIter.
while $LS \leqslant$ maxIter **do**
    $\boldsymbol{w}_k(\alpha) = P(\boldsymbol{w}_k - \alpha \nabla f_k)$;
    if $f(\boldsymbol{w}_k(\alpha)) \leqslant f(\boldsymbol{w}_k) + c_1 \alpha \nabla f_k^T (\boldsymbol{w}_k(\alpha) - \boldsymbol{w}_k)$ and $\|\boldsymbol{w}_k(\alpha) - \boldsymbol{w}_k\|_\infty \leqslant c_2 \Delta_k$ **then**
        Return $\boldsymbol{w}_k(\alpha)$;
        Break;
    $\alpha = \alpha * \beta$;
    $LS = LS + 1$;
if $LS >$ maxIter **then**
    Line search failed;
    Break;

If we assume that the approximate Cauchy point found in the first step is $\boldsymbol{w}^c$ and the active set corresponding to this point is $\mathcal{A}(\boldsymbol{w}^c)$, then we can construct a quadratic programming problem that is based on the current step $\boldsymbol{w}_k$:

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & f(\boldsymbol{w}_k) + \nabla f(\boldsymbol{w}_k)^T (\boldsymbol{w} - \boldsymbol{w}_k) + \frac{1}{2} (\boldsymbol{w} - \boldsymbol{w}_k)^T \boldsymbol{H}(\boldsymbol{w}_k) (\boldsymbol{w} - \boldsymbol{w}_k) \\ \text{subject to} \quad & \boldsymbol{w} \geqslant \boldsymbol{0}, \qquad i \notin \mathcal{A}(\boldsymbol{w}^c), \\ & \boldsymbol{w}_i = \boldsymbol{w}_i^c, \qquad i \in \mathcal{A}(\boldsymbol{w}^c), \\ & \|\boldsymbol{w} - \boldsymbol{w}_k\|_\infty \leqslant \Delta_k. \end{aligned} \tag{4.29}$$

Moreover, we should notice that $f(\boldsymbol{w}_k)$ is a constant and we can ignore this term. If we let $\boldsymbol{d} = \boldsymbol{w} - \boldsymbol{w}_k$, then we can simplify the objective function as $\nabla f(\boldsymbol{w}_k)^T \boldsymbol{d} + \frac{1}{2} \boldsymbol{d}^T \boldsymbol{H}(\boldsymbol{w}_k) \boldsymbol{d}$. The inequality constraint $\boldsymbol{w} \geqslant \boldsymbol{0}$ is equivalent to $\boldsymbol{w} - \boldsymbol{w}_k \geqslant -\boldsymbol{w}_k$, which is $\boldsymbol{d} \geqslant -\boldsymbol{w}_k$ using our notations. The third inequality constraint, $\|\boldsymbol{w} - \boldsymbol{w}_k\|_\infty \leqslant \Delta_k$, can be rewritten as $\|\boldsymbol{d}\|_\infty \leqslant \Delta_k$. This inequality constraint is equivalent to $-\Delta_k \leqslant \boldsymbol{d} \leqslant \Delta_k$. By combining the previous two constraints, we can obtain that $\max\{-\Delta_k, -\boldsymbol{w}_k\} \leqslant \boldsymbol{d} \leqslant \Delta_k$. For the elements in the active set, we have $\boldsymbol{w} - \boldsymbol{w}_k = \boldsymbol{w}^c - \boldsymbol{w}_k$, which is $\boldsymbol{d} = \boldsymbol{w}^c - \boldsymbol{w}_k$. So the previous optimization problem is equivalent

to

$$\begin{aligned} \min_{\boldsymbol{d}} \quad & \nabla f\left(\boldsymbol{w}_k\right)^T \boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T \boldsymbol{H}\left(\boldsymbol{w}_k\right)\boldsymbol{d} \\ \text{subject to} \quad & \max\left\{-\Delta_k, -\boldsymbol{w}_k\right\} \leqslant \boldsymbol{d} \leqslant \Delta_k, \\ & \boldsymbol{P}\boldsymbol{d} = \boldsymbol{w}^c - \boldsymbol{w}_k. \end{aligned} \tag{4.30}$$

where $\boldsymbol{P}$ is the projection matrix that keeps $\boldsymbol{d}_i$ invariant for $i \in \mathcal{A}\left(\boldsymbol{w}^c\right)$. We can construct the matrix $\boldsymbol{P}$ by assigning the i-th element in the i-th column to be 1 and otherwise 0, where the size of $\boldsymbol{P}$ is the number of active elements by $N_v \cdot N_m$. It is obvious that this problem is a quadratic programming problem with both bound and equality constraints. However, for the large-scale cases, solving this problem exactly is likely to be as expensive as solving the original problem [48]. Since we have obtained the approximate Cauchy point in the first step, we only want the solution to be at least better than the approximate Cauchy point, which means that we do not require an exact solution for the second step. So we can either stop the iteration when the current step crosses the boundaries or ignore the bound constraints, project the step back to the boundaries after it meets the stopping criteria and compare it with the approximate Cauchy point. In this chapter, we will ignore the bound constraints at first and then project the step back to the boundaries because the step can cross the boundaries within only 1 or 2 iterations, and in this case, it does not give us a sufficient reduction.

So far, we have not added any preconditioners to this problem. As we can see, the preconditioners proposed in the last section are used to precondition the Hessian, so it is not necessary to apply the preconditioners in the first step. In the second step, we substitute the Hessian of Gauss-Newton approximation for the true Hessian so we can try to use the preconditioners in this step. If we ignore the bound constraints,

we can formulate a quadratic programming problem with the preconditioners:

$$\begin{aligned} \min_{\tilde{\boldsymbol{d}}} \quad & \nabla \tilde{f}\left(\boldsymbol{w}_k\right)^T \tilde{\boldsymbol{d}} + \frac{1}{2} \tilde{\boldsymbol{d}}^T \tilde{\boldsymbol{H}}\left(\boldsymbol{w}_k\right) \tilde{\boldsymbol{d}} \\ \text{subject to} \quad & \tilde{\boldsymbol{P}} \tilde{\boldsymbol{d}} = \boldsymbol{w}^c - \boldsymbol{w}_k, \end{aligned} \tag{4.31}$$

where $\nabla \tilde{f}\left(\boldsymbol{w}_k\right) = \left(\boldsymbol{M}_2^T \otimes \boldsymbol{M}_1^T\right) \nabla f\left(\boldsymbol{w}_k\right)$, $\tilde{\boldsymbol{d}} = \left(\boldsymbol{M}_2^{-1} \otimes \boldsymbol{M}_1^{-1}\right) \boldsymbol{d}$, $\tilde{\boldsymbol{P}} = \boldsymbol{P}\left(\boldsymbol{M}_2 \otimes \boldsymbol{M}_1\right)$ and $\tilde{\boldsymbol{H}}\left(\boldsymbol{w}\right) = \left(\boldsymbol{M}_2^T \otimes \boldsymbol{M}_1^T\right) \boldsymbol{H}\left(\boldsymbol{w}\right)\left(\boldsymbol{M}_2 \otimes \boldsymbol{M}_1\right)$. We use the projected conjugated gradient (PCG) method to solve $\tilde{\boldsymbol{d}}$ directly and retrieve $\boldsymbol{d}$ using $\boldsymbol{d} = \left(\boldsymbol{M}_2 \otimes \boldsymbol{M}_1\right) \tilde{\boldsymbol{d}}$. The details are shown in Algorithm 2. In Algorithm 2, we need to use the approxi-

Algorithm 2 Projected Conjugate Gradient

Initialization:
$\boldsymbol{d} = \boldsymbol{w}^c - \boldsymbol{w}_k$;
$\tilde{\boldsymbol{d}} = \left(\boldsymbol{M}_2^{-1} \otimes \boldsymbol{M}_1^{-1}\right) \boldsymbol{d}$;
$\boldsymbol{r} = \tilde{\boldsymbol{H}}\left(\boldsymbol{w}_k\right) \tilde{\boldsymbol{d}} + \nabla \tilde{f}\left(\boldsymbol{w}_k\right)$;
$\boldsymbol{g} = \boldsymbol{r} - \boldsymbol{P}^T \left(\boldsymbol{P}\boldsymbol{P}^T\right)^{-1} \boldsymbol{P}\boldsymbol{r}$;
$\boldsymbol{p} = -\boldsymbol{g}$;
Set up the stopping criterion: tol, and the maximum number of iterations: maxIter;
while $i \leqslant$ maxIter **do**
    $\alpha = \boldsymbol{r}^T \boldsymbol{g} / \boldsymbol{p}^T \tilde{\boldsymbol{H}}\left(\boldsymbol{w}_k\right) \boldsymbol{p}$;
    $\tilde{\boldsymbol{d}} = \tilde{\boldsymbol{d}} + \alpha \boldsymbol{p}$;
    $\boldsymbol{r}^+ = \boldsymbol{r} + \alpha \tilde{\boldsymbol{H}}\left(\boldsymbol{w}_k\right) \boldsymbol{p}$;
    $\boldsymbol{g}^+ = \boldsymbol{r}^+ - \boldsymbol{P}^T \left(\boldsymbol{P}\boldsymbol{P}^T\right)^{-1} \boldsymbol{P}\boldsymbol{r}^+$;
    if $\boldsymbol{g}^{+T} \boldsymbol{r}^+ <$ tol **then**
        Return $\tilde{\boldsymbol{d}}$;
        Break;
    $\beta = \boldsymbol{r}^{+T} \boldsymbol{g}^+ / \boldsymbol{r}^T \boldsymbol{g}$;
    $\boldsymbol{p} = -\boldsymbol{g}^+ + \beta \boldsymbol{p}$;
    $\boldsymbol{g} = \boldsymbol{g}^+$;
    $\boldsymbol{r} = \boldsymbol{r}^+$;
    $i = i + 1$;

mate Cauchy point obtained in Algorithm 1 as the initial guess. Otherwise, we cannot guarantee the elements that are active will be fixed on the boundaries. Moreover, we can see that we need $\left(\boldsymbol{P}\boldsymbol{P}^T\right)^{-1}$ in each iteration, where $\boldsymbol{P}\boldsymbol{P}^T$ is large-scale, sparse and symmetric positive semi-definite. So we can use the Cholesky factorization to

find an upper triangular matrix $\boldsymbol{R}$ such that $\boldsymbol{P}\boldsymbol{P}^T = \boldsymbol{R}^T\boldsymbol{R}$. Compared with inverting $\boldsymbol{P}\boldsymbol{P}^T$ directly, it is more efficient to invert $\boldsymbol{R}^T\boldsymbol{R}$ because $\boldsymbol{R}$ is upper triangular and we only need to calculate the factorization once in Algorithm 2.

After we have obtained the new direction $\boldsymbol{d}$, we can compute the new step as $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \boldsymbol{d}$. To decide if we should accept the new step and the size of trust regions, we need to compute the actual reduction and the predicted reduction. The actual reduction is the difference between function values of the previous step and the new step:

$$ared = f(\boldsymbol{w}_k) - f(\boldsymbol{w}_{k+1}). \tag{4.32}$$

The predicted reduction is based on the quadratic expansion of the previous step:

$$pred = -\nabla f\left(\boldsymbol{w}_k\right)^T \boldsymbol{d} - \frac{1}{2}\boldsymbol{d}^T \boldsymbol{H}\left(\boldsymbol{w}_k\right)\boldsymbol{d}. \tag{4.33}$$

If $ared < 0$, then we accept the new step and update the trust region with respect to the ratio $ared/pred$. In this chapter, we use the standard way to update trust regions. After updating the current step, we should check if we should stop the iteration. If the current step meets the stopping criteria for local minimum, then we stop the iteration and report the results. Otherwise, we keep iterating until we find a local minimizer. By combining Algorithm 1 and Algorithm 2, we can express the main framework in Algorithm 3.

From the description, we can see that Algorithm 3 includes the strengths of both the projected line search method and the trust region method. With the projected line search method, we can guarantee a descent direction with proper reduction. With the trust region method, we are likely to obtain a better step and thus further reduction. However, this problem is a nonlinear least squares problem and the objective function might have multiple local minimizers. In addition, we truncate each step with respect to the lower bounds and it is hard to analyze the convergence properties. It is possible

Algorithm 3 Main Framework

Initialization:
Set up the test problem, the initial guess $\boldsymbol{w}_0$ and the stopping criteria.
while the stopping criteria do not satisfy **do**
    Use Algorithm 1 to compute the approximate Cauchy point $\boldsymbol{w}^c$;
    Find the active set of $\boldsymbol{w}^c$;
    Construct the project matrix $\boldsymbol{P}$ based on the active set;
    Calculate the new boundaries based on the original boundaries and the current trust region;
    Generate the preconditioners based on the current step $\boldsymbol{w}_i$;
    Use Algorithm 2 to compute the new step $\boldsymbol{w}_{new}$;
    Project the new step $\boldsymbol{w}_{new}$ onto the new boundaries to obtain $\boldsymbol{w}_{i+1}$;
    Calculate *pred* and *ared* using $\boldsymbol{w}_i$ and $\boldsymbol{w}_{i+1}$;
    Decide if we should accept $\boldsymbol{w}_{i+1}$ and update the trust region;

that the new step might not offer sufficient reduction but it is closer to the global minimizer. For example, we can think about an 1D problem of climbing the hill. On the other side of the hill, it might have another point that has even lower altitude. During the process of climbing, we might need to increase the altitude to a certain degree in order to achieve better results later. Therefore, it is not possible to decide specific criteria that fit all conditions. For example, we can even accept the new step every time and only decide the size of trust region with respect to the ratio *ared*/*pred*. In practice, this method still gives us a fast convergence and high-quality image reconstruction.

4.3.2 Regularization and Scaling

In this chapter, we assume that the measured data obtained from the detector follow a Poisson distribution. To remove the noise and stabilize the solution, we need to introduce regularization terms. As the weights of each material correspond to an image, we can construct the regularization term with respect to the specified material. If we assume $\boldsymbol{w}$ is concatenated by $\boldsymbol{w}_1,\ \boldsymbol{w}_2,\ \cdots,\ \boldsymbol{w}_m$ where $\boldsymbol{w}_i$ is the vectorization

of the i-th material map, then we can build the regularization term as

$$R(\boldsymbol{w}) = \sum_{i=1}^{N_m} \alpha_i R(\boldsymbol{w}_i), \tag{4.34}$$

where $R(\boldsymbol{w}_i)$ represents the regularization term for the i-th material and α_i is the corresponding regularization parameter. Using this method, it gives us more freedom to choose the regularization term with respect to the material. In practice, we can usually find that one or several specified materials dominate the object and it is likely that the reconstructed material maps will contain many edges. So we select generalized Tikhonov regularization for these materials to smooth the edges. The forward difference operator and zero boundary conditions are used to build the discrete differential operator. On the other hand, other materials might only occupy a small area so we think about restrict the sum of weights for these materials. To realize this idea, we introduce ℓ_1 regularization to penalize the sum of weights. With the generalized Tikhonov regularization and the ℓ_1 regularization, we need to select the corresponding regularization parameters. It is not obvious how we can use regular methods, such as L-curve and generalized cross-validation, to find proper parameters. In this case, we generate a log space for each regularization and use the grid search method to find the "best" regularization parameters.

Another challenge associated with the regularization parameter is how to scale the problem. The point such that the objective function is zero might not be feasible and the residual corresponding to the global minimizer might still be a large number. In this case, choosing a proper regularization parameter is hard if the magnitude is large. In addition to selecting the regularization parameter, the infeasible step might result in the difficulty of meeting stopping criteria and thus increase the number of CG iterations. For this problem, we want to scale the objective function, gradient and Hessian with the spectrum radius of the Hessian based on the 2-norm. However,

it is not necessary to compute the largest singular value of the Hessian. It is easy to see that $\boldsymbol{H}(\boldsymbol{w})$ is a nonnegative and symmetric matrix, so we can use the Shur's test [40] to find an upper bound for this value with the initial step $\boldsymbol{w}_0$:

$$\|\boldsymbol{H}(\boldsymbol{w}_0)\|_2 \leqslant \sqrt{\|\boldsymbol{H}(\boldsymbol{w}_0)\|_1 \|\boldsymbol{H}(\boldsymbol{w}_0)\|_\infty}. \tag{4.35}$$

With the nonnegativity of $\boldsymbol{H}(\boldsymbol{w}_0)$, we can calculate the right hand side of Inequality (4.35) Using matrix-vector multiplication rather than form the Hessian explicitly. Using the symmetry of the Hessian, we can obtain that $\|\boldsymbol{H}(\boldsymbol{w}_0)\|_1 = \|\boldsymbol{H}(\boldsymbol{w}_0)\|_\infty$. Then Inequality (4.35) can be simplified as

$$\|\boldsymbol{H}(\boldsymbol{w}_0)\|_2 \leqslant \|\boldsymbol{H}(\boldsymbol{w}_0)\|_\infty. \tag{4.36}$$

So we can use $\|\boldsymbol{H}(\boldsymbol{w}_0)\|_\infty$ as the scaling parameter for the objective function, the gradient and the Hessian. With this scaling parameter, we can choose regularization parameters with less effort.

4.4 Numerical Experiments

To test the preconditioners and the optimization method, we generate a 2D image of the size 128 by 128 as the object. We also assume that this object is made up of two materials, plexiglass and polyvinyl chloride (PVC). Thus we can obtain 2 material maps corresponding to the weights of these two materials. The original material maps are shown in Figure 4.2. In Figure 4.2, the yellow color represents that it has the corresponding material in this area, while the blue color shows that it does not have the corresponding material in this area. Therefore, we can see that the object is a circle and both materials are distributed inside this circle. Inside this circle, plexiglass dominates most areas except three dots occupied by PVC. We can also

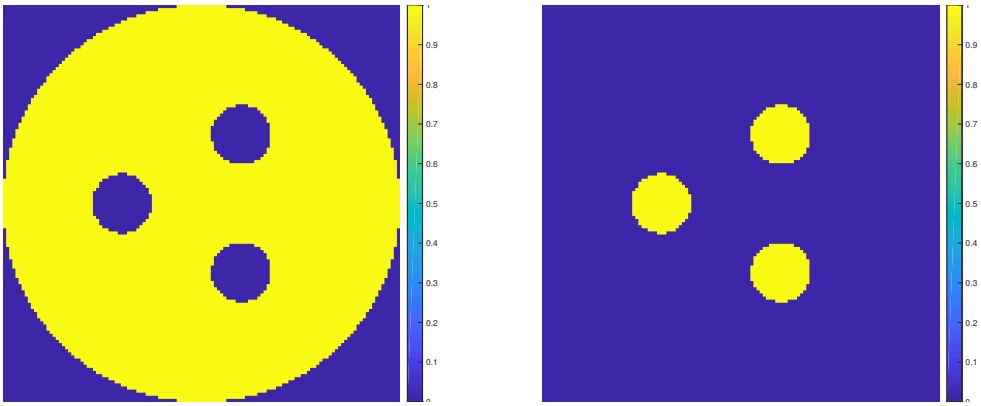

Figure 4.2: The original images for plexiglass (left) and PVC (right).

see that the images corresponding to these two materials compensate each other and they are completely separable. The goal of this numerical experiment is to reconstruct these two images such that different material maps present the corresponding material compositions.

Moreover, we present the plot of the linear attenuation coefficients for these two materials in Figure 4.3. In Figure 4.3, we can see that the slopes of these two curves

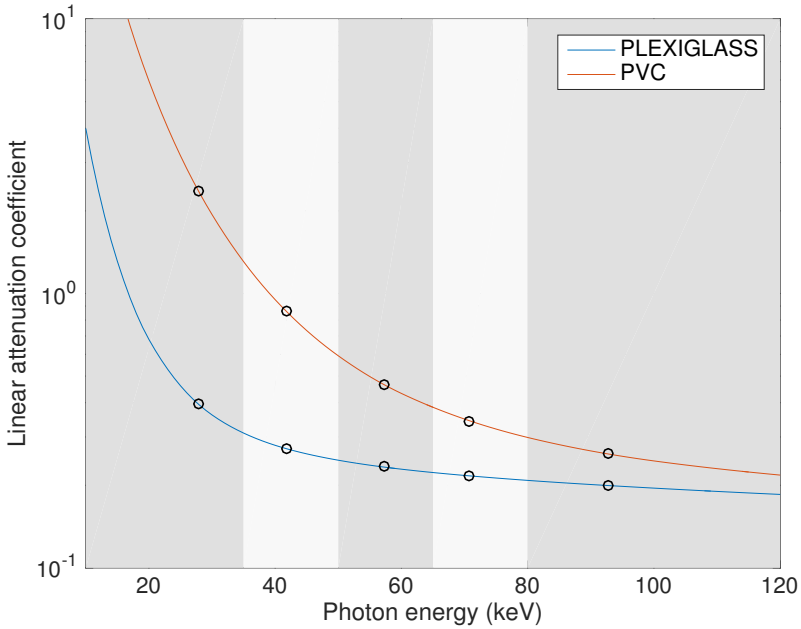

Figure 4.3: The linear attenuation coefficients for plexiglass and PVC.

are close to each other so it might indicate the collinearity of linear attenuation

coefficients and thus the linear correlation of columns of the matrix $\boldsymbol{C}$. It might also introduce small singular values and strengthen the ill-posedness.

In addition to the previous images, the geometry parameters of the CT machine are set as Table 4.1. With these parameters, we generate a distance matrix, $\boldsymbol{A}$, to represent the fan-beam geometry for flat detector using the MATLAB function `fanbeamtomolinear` [35]. Moreover, we choose 180 projections and they are dis-

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 4.1: Geometry parameters of the CT machine.

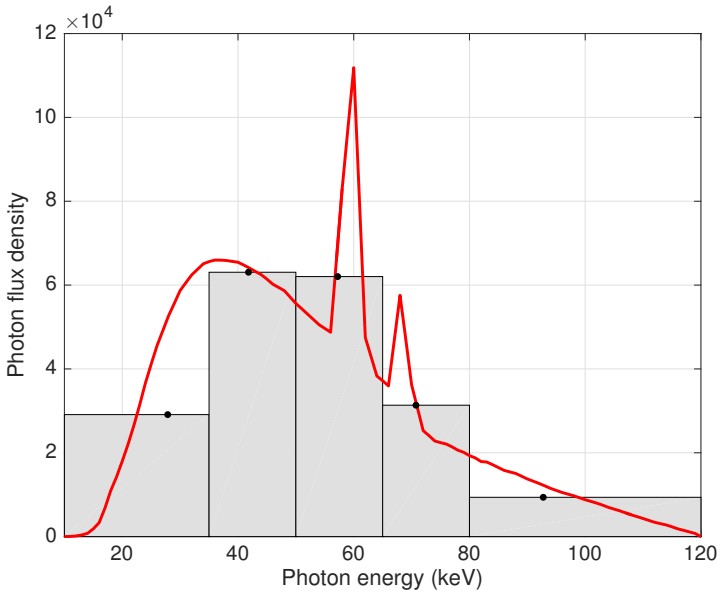

Figure 4.4: Detector bins and photon flux density.

tributed equally from 0 to 360 degrees. The voltage of the x-ray source is assumed to be 120 keV and the corresponding spectrum is generated using the MATLAB function `spektrSpectrum` [53]. We also assume that we have 5 detector bins and each of them can detect the specific range of photon energy: 10 to 34 keV, 35 to 49 keV, 50 to 64

keV, 65 to 79 keV and 80 to 120 keV. The detector is assumed to be photon counting detector and the data obtained from the detector are positive integers. Both the x-ray spectrum and the detector bins are presented in Figure 4.4. In Figure 4.4, the small black dots represent the values of mean energy in each bin. When we construct the forward problem, we use the full spectrum and all linear attenuation coefficients. However, we only use the mean energy and the corresponding linear attenuation coefficients to compute reconstruction. In this way, we can avoid inverse crime but the grid for reconstruction is coarser. With all these parameters, the goal is to reconstruct two material maps with the proposed methods.

With previous preparations, we set up random numbers between 0 and 1 as the initial guess and run the main algorithm. It only takes us around 116 seconds to converge even if we do not use the optimal implementation. The reconstructed images are presented in Figure 4.5. In Figure 4.5, we can see that we have successfully

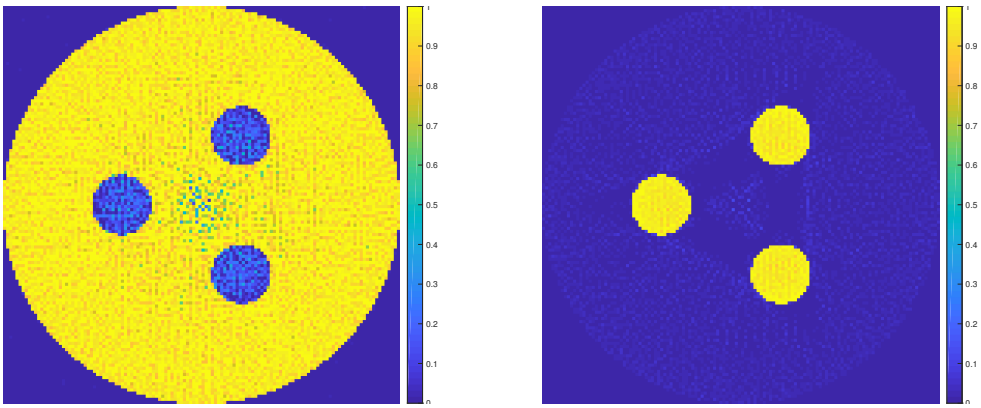

Figure 4.5: The reconstructed images for plexiglass (Left) and PVC (Right).

separated these two materials and it is hard to find many overlaps betwen these two material maps. Moreover, we can see that the shapes of these two material maps are exactly the same as the original images. In addition, the boundaries of large circle and three small dots are clear to identify, which shows the strength of results. By comparing Figure 4.5 with Figure 4.2, we can see that the reconstructed images have more shades than the original ones. For the first material map, most weights

are yellow but several of them are not exact 1. These light yellow weights represent the values that are not 1 but significantly close to 1. It might result from the noise contained in the data. In the center of this material map, we can see several dots of high frequency, which depart from the true solution to a certain degree. On the other hand, the reconstruction of the second material map is of higher quality and it is hard to tell the difference between the original image and the reconstructed one. It is possible that the weights corresponding to the second material map are much less than the weights of the first material map so that they contain less noise.

To check the convergence, we can also plot the relative errors for these two materials. The plot of relative errors is shown in Figure 4.6. In Figure 4.6, we can see that the curves of both materials drop and stagnate, which indicates the convergence. Moreover, the blue curve decreases faster than the red one. It shows that the convergence rate for the second material is faster than the first one. We can also see that

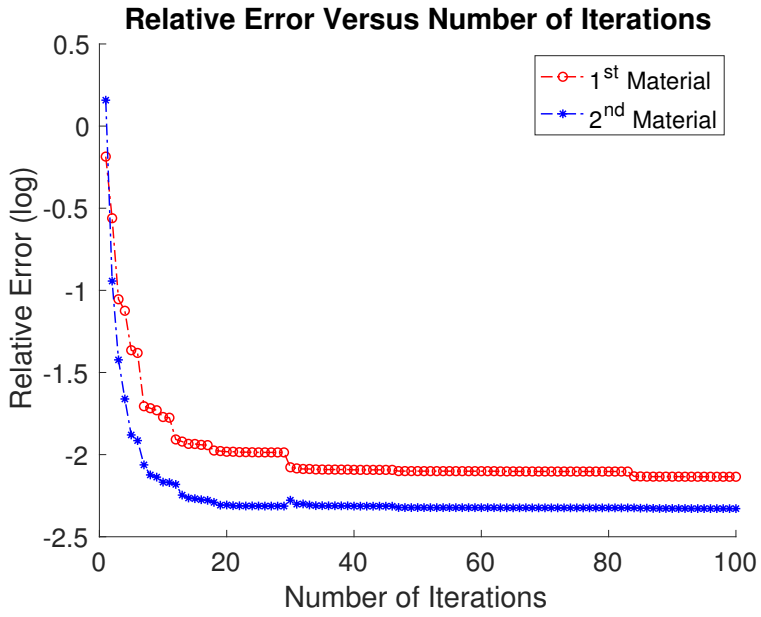

Figure 4.6: The Relative Errors for Plexiglass (red) and PVC (blue dash).

the blue curve reaches a lower level than the red one when they stagnate. It reflects the observation that the reconstruction of the second material map has better quality than the first one. Generally speaking, the ultimate relative errors for both materials

are around 10% and the overall reconstruction is satisfactory.

To further confirm the convergence, we can plot the decay of norm of the gradient. This plot is shown in Figure 4.7. In Figure 4.7, we can see that most of times, the

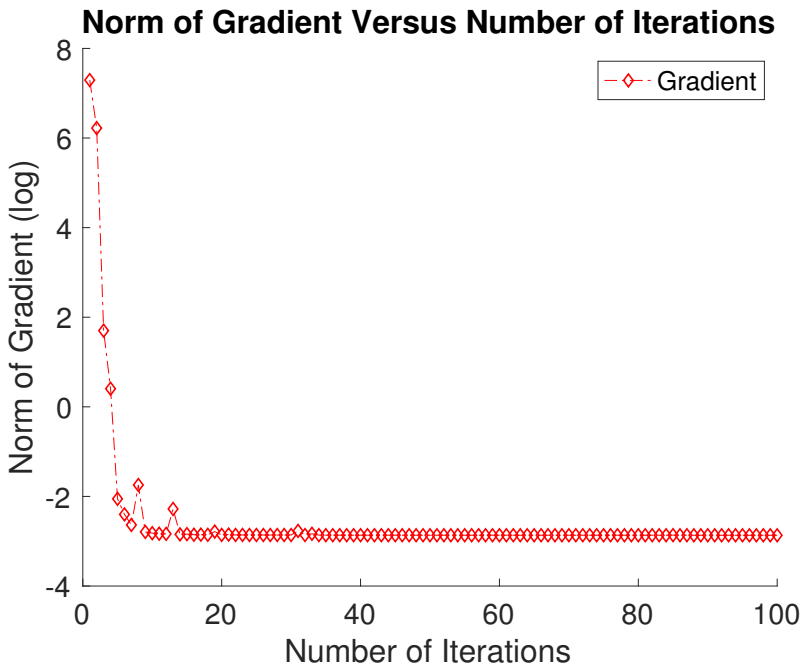

Figure 4.7: The decay of norm of the gradient.

curve drops as expected. However, there are two "spikes" that dissatisfied the trend of decrease. I think this might result from the projections onto the boundary or the irregular updating scheme of trust region. Furthermore, the best reconstructed images might not match the optimal solutions to the optmization problem.

To see the significance of the preconditioner, we can compare the number of CG iterations with and without preconditioners before the relative errors reach a specified level. We keep other parameters identical but shrink the size of images to 64 by 64 and the number of projections to 96. Moreover, we set the maximum number of CG iterations to 5000. After 5 Newton iterations, the methods with and without preconditioner both reach relative errors around 23%. The results are shown in Table 4.2. From Table 4.2, we can find that the number of CG iterations in each Newton step is significantly reduced after preconditioning. With the proposed method, the New-

	No Precond.			Precond.		
#	Rel. Err.	Num. CG	Time (s)	Rel. Err.	Num. CG	Time (s)
1	1.25	2625	87.8	0.84	9	0.3
2	1.67	318	11.5	0.46	6	0.3
3	1.26	111	3.8	0.30	13	0.5
4	0.27	4051	132.3	0.27	8	0.3
5	0.23	1347	44.2	0.22	12	0.5

Table 4.2: The comparison of CG iterations.

ton system is sufficiently better-conditioned or the eigenvalues are remarkably more clustered.

4.5 Conclusions and Remarks

With the energy-windowed spectral CT model, we set up a nonlinear least squares problem under bound constraints. To solve this optimization problem, we propose a new preconditioner and then implement it into a two-step method. The new preconditioner can transform the eigenvalues of the original Gauss-Newton system into more clustered ones, which will lead to faster convergence and higher accuracy. With the introduction of the two-step method, we can guarantee that the obtained step is at least better than the approximate Cauchy point. By solving the Gauss-Newton system in the second step, we expect further reduction from the solution. Therefore, the convergence rate should be better than linearity. Moreover, we further restrict the size of each step with the help of trust regions. In addition, we can remove parts of noise and speed up the convergence rate with the scaling parameter and the regularization terms.

On the other hand, it still has several limitations. Because of the nonlinearity of the objective function, it is hard to decide if we should accept the new step or not under certain circumstances. Furthermore, with multiple materials, we cannot use regular methods to choose regularization parameters. In each iteration, we need

to solve a NKP problem to obtain the preconditioner. Even if the computational cost is cheap, it might be better if we can find a preconditioner that is feasible for all steps. We might also think about how to implement the preconditioner into first order methods such as FISTA [3].

Chapter 5

Preconditioning and Optimization for Energy-windowed Spectral Computed Tomography

In Chapter 4, we discussed the energy-windowed spectral CT model and presented a preconditioning framework and a nonlinear optimization approach to compute the solution. In this chapter, we still focus on the energy-windowed spectral CT model

$$\boldsymbol{Y} = \exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}, \tag{5.1}$$

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 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 constraint $\boldsymbol{W} \geqslant \boldsymbol{0}$.

To solve Equation (5.1), we follow the instruction in Chapter 4 and take a vectorization at first. Rather than building a nonlinear optimization problem, 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 5.1, we can transform this equation into a weighted least squares problem under bound constraints:

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2}\|\mathcal{A}\boldsymbol{w}-\boldsymbol{b}\|^2_{\boldsymbol{\Sigma}^{-1}} \\ \text{subject to} \quad & \boldsymbol{w} \geqslant \boldsymbol{0}, \end{aligned} \tag{5.2}$$

where $\mathcal{A} = \boldsymbol{C} \otimes \boldsymbol{A}$, $\boldsymbol{b} = -\log(\boldsymbol{y})$, $\boldsymbol{y} = \mathrm{vec}(\boldsymbol{Y})$ and $\boldsymbol{w} = \mathrm{vec}(\boldsymbol{W})$. $\boldsymbol{\Sigma}^{-1}$, which combines information of $\boldsymbol{S}$ and $\boldsymbol{y}$, is the inverse covariance matrix generated by the Gaussian noise and the 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 $\boldsymbol{C}$ collects the corresponding linear attenuation coefficients and two materials, such as adipose and glandular, might be similar to each other, the matrix $\boldsymbol{C}$ is likely to be ill-conditioned. On the other hand, the problem (5.2) is similar to a quadratic programming problem under bound constraints. However, direct implementation of optimization solvers 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.

To handle these problems, we propose a new preconditioner that is based on rank-

1 approximation of the matrix $\boldsymbol{Y}$. With this rank-1 approximation, we can estimate the Hessian of the objective function in (5.2) using 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 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 the identity matrix, 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. In [2], Barber et al. propose an alternative preconditioner that is based on the eigenvalue decomposition of $\boldsymbol{C}^T\boldsymbol{C}$, where $\boldsymbol{C}$ is the matrix of linear attenuation coefficients. Compared with this, the preconditioner proposed in this chapter involves not only $\boldsymbol{C}$, but also includes information of the energy spectrum $\boldsymbol{S}$ and parts of the photon counting data, $\boldsymbol{Y}$, and thus provides 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) [3, 46, 47] 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 the form $f\left(\boldsymbol{x}\right) + g\left(\boldsymbol{x}\right)$, 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 (5.2) and $g\left(\boldsymbol{x}\right)$ can represent a nonsmooth regularization 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 complement each other and exhibit high-quality reconstructed images and fast convergence results.

This chapter is organized as follows. In Section 5.1, we review the continuous energy-windowed spectral CT model and the corresponding discretized nonlinear matrix equation. The key idea of this chapter, preconditioning, is introduced in Section 5.2. 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 5.3, we study FISTA and how we construct and solve the projection problems. Moreover, numerical experiments are presented in Section 5.4 and concluding remarks are given in Section 5.5.

5.1 The Weighted Least Squares Problem

In Chapter 4, we have discussed energy-windowed spectral CT model and introduced a new preconditioner based on the corresponding nonlinear least squares problem and the Gauss-Newton approximation of the Hessian. A two-step optimization method that includes projected line search and trust region method is implemented to solve this problem. However, there are two main concerns related to the preconditioner and optimization. At first, the preconditioner requires the information of the current iteration and even if it is cheap to compute, we still need to repeat the computational process in each Newton iteration. Secondly, since the nonlinear optimization is based

on the Gauss-Newton approximation of the Hessian, it raises a question if we can come up with a new preconditioner and use a first order method to solve it. In this chapter, we still focus on the energy-windowed spectral CT model and we present a linearization technique to transform the nonlinear equation into an optimization problem that is based on a weighted least squares term and a bound constraint. Recall that the basic energy-windowed spectral CT model is expressed by

$$y_i^{(k)} = \int_E S^{(k)}(e) \exp\left(- \int_{t \in l} \mu\left(\vec{r}(t), e\right) \mathrm{d}\, t \right) \mathrm{d}\, e + \eta_i^{(k)}. \quad \begin{cases} i = 1, 2, \cdots, N_d \times N_p, \\ k = 1, 2, \cdots, N_b, \end{cases} \tag{5.3}$$

Using the material decomposition, $\mu\left(\vec{r}(t), e\right) = \sum_{m=1}^{N_m} u_{m,e} w_m\left(\vec{r}\right)$, discretizing Equation (5.3) over energy E, and concatenating corresponding variables with respect to energy windows, we can obtain a matrix equation

$$\boldsymbol{Y} = \exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}, \tag{5.4}$$

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 m-th material at the energy level e. For similar materials such as adipose and glandular, the collinearity might cause the ill-conditioning of $\boldsymbol{C}$.
- $\boldsymbol{S}$ is a matrix of the size $N_e \times N_b$ and each column collects the spectrum energy of a specific range. We assume that $\boldsymbol{S}$ is square and invertible.

- $\boldsymbol{\mathcal{E}}$ is the noise matrix that is of size $(N_d \cdot N_p) \times N_b$. The assumption for noise is that $E_{il} \sim \mathcal{N}(0, y_{il})$ for each element E_{il} in $\boldsymbol{\mathcal{E}}$ and y_{il} in $\boldsymbol{Y}$.

In Figure 4.4, we present a plot of spectrum energy. The y-axis represents photon flux density while the x-axis indicates photon energy. The red curve in Figure 4.4 shows the relationship between photon flux density and energy under 120 keV voltage. The gray boxes represents the energy bins and as we can identify, there are five energy bins. The black dot in each energy bin illustrates the average photon density in the corresponding bin. If we use average energies rather than full spectrum when solving the inverse problem, the matrix $\boldsymbol{S}$ is diagonal and it gives us more flexibility on manipulating the equation.

Moreover, the current assumption for the noise matrix $\boldsymbol{\mathcal{E}}$ is that each entry follows a normal distribution with mean 0 and variance y_{il} (the corresponding entry in $\boldsymbol{Y}$). That is to say, $E_{il} \sim \mathcal{N}(0, y_{il})$. In the last chapter, we assume that $y_{il} \sim \text{Poisson}\left(\left[\exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{S}\right]_{il}\right)$ and Poisson distribution is the most fundamental assumption because of randomness of x-ray photon motion and errors of photon counting. However, a Poisson distribution is accurately approximated by a Gaussian distribution if the mean of this Poisson distribution is large enough. For spectral CT problems, this mean corresponds to the number of photons and this number is often hundreds of thousands. With this large number as the mean, the Gaussian assumption is valid.

In several cases, the composition of materials can be similar. For example, glandular and adipose have similar attenuation coefficients at the same energy levels and this feature can cause 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 these small singular values 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 (5.4):

$$\boldsymbol{Y} = \exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{M}^{-T}\boldsymbol{M}^{T}\boldsymbol{C}^{T}\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}. \tag{5.5}$$

If we let $\tilde{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{M}^{-T}$ and $\tilde{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{M}$, then Equation (5.5) is equivalent to

$$\boldsymbol{Y} = \exp\left(-\boldsymbol{A}\tilde{\boldsymbol{W}}\tilde{\boldsymbol{C}}^{T}\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}. \tag{5.6}$$

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 (5.2.1), 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 (5.6) directly because it might introduce tensors when we compute second order derivatives. In this case, we want to vectorize Equation (5.6) on both sides and linearize it to construct a weighted least squares optimization problem. In the forward problem, we use the full spectrum and the matrix $\boldsymbol{S}$ is 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, $N_e = N_b$ 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 Equation (5.6):

$$\boldsymbol{Y}\boldsymbol{S}^{-1} = \exp\left(-\boldsymbol{A}\tilde{\boldsymbol{W}}\tilde{\boldsymbol{C}}^{T}\right) + \boldsymbol{\mathcal{E}}\boldsymbol{S}^{-1}. \tag{5.7}$$

Vectorizing both sides of (5.7), and using properties of Kronecker products, we obtain

$$\left(\boldsymbol{S}^{-T} \otimes \boldsymbol{I}\right) \boldsymbol{y} = \exp\left\{-\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) \tilde{\boldsymbol{w}}\right\} + \left(\boldsymbol{S}^{-T} \otimes \boldsymbol{I}\right) \boldsymbol{e}, \tag{5.8}$$

where $\boldsymbol{y} = \text{vec}(\boldsymbol{Y})$, $\tilde{\boldsymbol{w}} = \text{vec}(\tilde{\boldsymbol{W}})$ and $\boldsymbol{e} = \text{vec}\,(\boldsymbol{E})$. 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 (5.8) and obtain

$$\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{e}} = \exp\left\{-\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) \tilde{\boldsymbol{w}}\right\}. \tag{5.9}$$

By taking the logarithm on both sides of Equation (5.9), we can obtain an equation:

$$\log\left(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{e}}\right) = -\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) \tilde{\boldsymbol{w}}. \tag{5.10}$$

However, the left-hand side of Equation (5.10) 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 of the logarithm at $\tilde{\boldsymbol{y}}$:

$$\log\left(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{e}}\right) = \log\left(\tilde{\boldsymbol{y}}\right) - \text{diag}\left(\tilde{\boldsymbol{y}}\right)^{-1} \tilde{\boldsymbol{e}} + \mathcal{O}\left(\|\tilde{\boldsymbol{e}}\|_2^2\right). \tag{5.11}$$

If we use the first two terms on the right-hand side of Equation (5.11) to estimate the term $\log\left(\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{e}}\right)$, then Equation (5.10) can be rewritten as 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 (5.10) is equivalent to

$$\boldsymbol{b} \approx \left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) \tilde{\boldsymbol{w}} - \text{diag}\left(\tilde{\boldsymbol{y}}\right)^{-1} \tilde{\boldsymbol{e}}. \tag{5.12}$$

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

$$\boldsymbol{b} | \tilde{\boldsymbol{w}} \sim \mathcal{N}\left(\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) \tilde{\boldsymbol{w}},\ \boldsymbol{\Sigma}\right), \tag{5.13}$$

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

$$\boldsymbol{\Sigma} = \operatorname{diag}(\tilde{\boldsymbol{y}})^{-1} \left(\boldsymbol{S}^{-T} \otimes \boldsymbol{I}\right) \operatorname{diag}(\boldsymbol{y}) \left(\boldsymbol{S}^{-1} \otimes \boldsymbol{I}\right) \operatorname{diag}(\tilde{\boldsymbol{y}})^{-1} , \tag{5.14}$$

and the inverse noise covariance matrix is given by

$$\boldsymbol{\Sigma}^{-1} = \operatorname{diag}(\tilde{\boldsymbol{y}}) (\boldsymbol{S} \otimes \boldsymbol{I}) \operatorname{diag}(\boldsymbol{y})^{-1} \left(\boldsymbol{S}^{T} \otimes \boldsymbol{I}\right) \operatorname{diag}(\tilde{\boldsymbol{y}}) . \tag{5.15}$$

Since $\boldsymbol{y} = \operatorname{vec}(\boldsymbol{Y})$ and $\boldsymbol{Y}$ 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 on the order of hundreds of thousands. Moreover, $\tilde{\boldsymbol{y}} = \left(\boldsymbol{S}^{-T} \otimes \boldsymbol{I}\right) \boldsymbol{y}$ and as long as the noise does not dominate the projected data, we expect the entries of $\tilde{\boldsymbol{y}}$ to be larger than zero. From the expression (5.15), we can see that the structure of $\boldsymbol{\Sigma}^{-1}$ depends on the structure of matrix $\boldsymbol{S}$. If $\boldsymbol{S}$ is diagonal, then $\boldsymbol{\Sigma}$ is also diagonal. Otherwise, $\boldsymbol{\Sigma}^{-1}$ is a block diagonal matrix. If we let $\mathcal{A} = \tilde{\boldsymbol{C}} \otimes \boldsymbol{A}$ and ignore constants, the corresponding probability density function is given by

$$f(\boldsymbol{b}; \tilde{\boldsymbol{w}}) = \exp\left\{ -\frac{1}{2} (\mathcal{A}\tilde{\boldsymbol{w}} - \boldsymbol{b})^T \boldsymbol{\Sigma}^{-1} (\mathcal{A}\tilde{\boldsymbol{w}} - \boldsymbol{b}) \right\} . \tag{5.16}$$

So the log-likelihood function is given by

$$l(\tilde{\boldsymbol{w}}; \boldsymbol{b}) = -\frac{1}{2} (\mathcal{A}\tilde{\boldsymbol{w}} - \boldsymbol{b})^T \boldsymbol{\Sigma}^{-1} (\mathcal{A}\tilde{\boldsymbol{w}} - \boldsymbol{b}) . \tag{5.17}$$

We try to maximize the log-likelihood function $l(\tilde{\boldsymbol{w}}; \boldsymbol{b}, \boldsymbol{\Sigma})$, which is equivalent to minimizing the negative log-likelihood function $-l(\tilde{\boldsymbol{w}}; \boldsymbol{b}, \boldsymbol{\Sigma})$. In addition, we require that $\boldsymbol{W} \geqslant \boldsymbol{0}$, and with the preconditioner, these constraints are transformed into $(\boldsymbol{M} \otimes \boldsymbol{I})\, \tilde{\boldsymbol{w}} \geqslant \boldsymbol{0}$. Therefore, we can formulate a weighted least squares problem

under bound constraints

$$\begin{aligned} \min_{\tilde{\boldsymbol{w}}} \quad & \frac{1}{2}\|\mathcal{A}\tilde{\boldsymbol{w}} - \boldsymbol{b}\|^2_{\boldsymbol{\Sigma}^{-1}} \\ \text{subject to} \quad & (\boldsymbol{M} \otimes \boldsymbol{I})\,\tilde{\boldsymbol{w}} \geqslant \boldsymbol{0}. \end{aligned} \tag{5.18}$$

In Equation (5.18), the norm $\|\cdot\|^2_{\boldsymbol{\Sigma}^{-1}}$ corresponds to the negative log-likelihood function $-l\left(\tilde{\boldsymbol{w}}; \boldsymbol{b}, \boldsymbol{\Sigma}\right)$. 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 choose suitable regularizations for the corresponding materials. Thirdly, we have to find an efficient method to solve the weighted least squares problem. These three challenges are related to each other and an appropriate preconditioner with feasible regularizations will be beneficial for the solver efficiency. Finally, we should handle linear inequality constraints in an efficient way. We address these four challenges in the following sections.

5.2 Preconditioning and Regularization

5.2.1 Preconditioning

The choice of the preconditiong matrix $\boldsymbol{M}$ is crucial for solving the optimization problem (5.18). 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 is expressed as

$$\boldsymbol{H} = (\boldsymbol{C}^T \otimes \boldsymbol{A}^T)\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{C} \otimes \boldsymbol{A}\right). \tag{5.19}$$

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

$$\tilde{\boldsymbol{H}} = \mathcal{A}^T \boldsymbol{\Sigma}^{-1} \mathcal{A} = \left(\tilde{\boldsymbol{C}}^T \otimes \boldsymbol{A}^T \right) \boldsymbol{\Sigma}^{-1} \left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A} \right), \tag{5.20}$$

where $\tilde{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{M}$. From this expression, it is still not obvious how to construct the preconditioner. However, if we can decompose the noise covariance matrix $\boldsymbol{\Sigma}^{-1}$ into a Kronecker product of two terms, then we can merge several terms using the properties of Kronecker product and transform parts of the products into identity. To realize this idea, we review the expression of $\boldsymbol{\Sigma}^{-1}$ in Equation (5.15), 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 $\boldsymbol{Y} \approx \boldsymbol{u}\boldsymbol{v}^T$, then

$$\operatorname{diag}\{\boldsymbol{y}\} \approx \operatorname{diag}\left\{ \operatorname{vec}\left(\boldsymbol{u}\boldsymbol{v}^T \right) \right\} = \operatorname{diag}\{\boldsymbol{v}\} \otimes \operatorname{diag}\{\boldsymbol{u}\}. \tag{5.21}$$

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

$$\min_{\boldsymbol{u},\, \boldsymbol{v}} \quad \|\boldsymbol{Y} - \boldsymbol{u}\boldsymbol{v}^T\|_F. \tag{5.22}$$

The solution to this problem is similar to the one in Chapter 4, but it has several variations. Using the singular value decomposition (SVD), one solution to Problem (5.22) 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. 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" [39], to compute 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 (5.21). In addition, the term $\operatorname{diag}\{\tilde{\boldsymbol{y}}\}$ can be represented as

$$\begin{aligned}\operatorname{diag}\{\tilde{\boldsymbol{y}}\} &= \operatorname{diag}\left\{\left(\boldsymbol{S}^{-T}\otimes\boldsymbol{I}\right)\operatorname{vec}\left(\boldsymbol{Y}\right)\right\}\\ &\approx \operatorname{diag}\left\{\left(\boldsymbol{S}^{-T}\otimes\boldsymbol{I}\right)\operatorname{vec}\left(\boldsymbol{u}\boldsymbol{v}^{T}\right)\right\}\\ &= \operatorname{diag}\left\{\operatorname{vec}\left(\boldsymbol{u}\boldsymbol{v}^{T}\boldsymbol{S}^{-1}\right)\right\}\\ &= \operatorname{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\}\otimes\operatorname{diag}\left\{\boldsymbol{u}\right\}.\end{aligned} \tag{5.23}$$

If we substitute the terms in (5.21) and (5.23) for the same terms in (5.15), then we can obtain that

$$\boldsymbol{\Sigma}^{-1}\approx\left(\operatorname{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\}\boldsymbol{S}\operatorname{diag}\left\{\boldsymbol{v}\right\}^{-1}\boldsymbol{S}^{T}\operatorname{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\}\right)\otimes\operatorname{diag}\left\{\boldsymbol{u}\right\}. \tag{5.24}$$

So the preconditioned Hessian matrix is given by

$$\begin{aligned}\tilde{\boldsymbol{H}} &= \left(\tilde{\boldsymbol{C}}^{T}\otimes\boldsymbol{A}^{T}\right)\boldsymbol{\Sigma}^{-1}\left(\tilde{\boldsymbol{C}}\otimes\boldsymbol{A}\right)\\ &\approx \left(\tilde{\boldsymbol{C}}^{T}\otimes\boldsymbol{A}^{T}\right)\left(\operatorname{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\}\boldsymbol{S}\operatorname{diag}\left\{\boldsymbol{v}\right\}^{-1}\boldsymbol{S}^{T}\operatorname{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\}\otimes\operatorname{diag}\left\{\boldsymbol{u}\right\}\right)\left(\tilde{\boldsymbol{C}}\otimes\boldsymbol{A}\right)\\ &= \left(\tilde{\boldsymbol{C}}^{T}\operatorname{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\}\boldsymbol{S}\operatorname{diag}\left\{\boldsymbol{v}\right\}^{-1}\boldsymbol{S}^{T}\operatorname{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\}\tilde{\boldsymbol{C}}\right)\otimes\left(\boldsymbol{A}^{T}\operatorname{diag}\left\{\boldsymbol{u}\right\}\boldsymbol{A}\right).\end{aligned} \tag{5.25}$$

Since the size of $\tilde{\boldsymbol{C}}$ is $N_e\times N_m$, then the first part of the Kronecker product in (5.25) is a square matrix of the size 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 to form

the object so that the size of 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 the matrix product, $\boldsymbol{C}^T\text{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\}\boldsymbol{S}\text{diag}\left\{\boldsymbol{v}\right\}^{-1}\boldsymbol{S}^T\text{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\}\boldsymbol{C}$, is a symmetric positive definite (SPD) matrix. Therefore, we can calculate $\boldsymbol{M}$ using the Cholesky decomposition:

$$\boldsymbol{C}^T\text{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\}\boldsymbol{S}\text{diag}\left\{\boldsymbol{v}\right\}^{-1}\boldsymbol{S}^T\text{diag}\left\{\boldsymbol{S}^{-T}\boldsymbol{v}\right\}\boldsymbol{C}=\boldsymbol{G}^T\boldsymbol{G}, \tag{5.26}$$

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 (5.25), we see that the preconditioned Hessian matrix $\tilde{\boldsymbol{H}}$ is dependent on a Kronecker product of two parts and the first part has been transformed into an identity matrix. 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 that 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

$$\kappa\left(\boldsymbol{H}\right)\approx\frac{\sigma_{1max}\sigma_{2max}}{\sigma_{1min}\sigma_{2min}}. \tag{5.27}$$

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

$$\kappa\left(\tilde{\boldsymbol{H}}\right)\approx\frac{\sigma_{2max}}{\sigma_{2min}}. \tag{5.28}$$

Since the fraction $\sigma_{1max}/\sigma_{1min}$ is most likely to be significantly greater than 1, then the condition number of $\tilde{\boldsymbol{H}}$ is likely to be much smaller than $\boldsymbol{H}$. Furthermore, we can build a numerical example to validate this phenomenon. 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 5.1. From Table 5.1, we can see that the difference of condition number for $\boldsymbol{H}$ and $\tilde{\boldsymbol{H}}$ is around

Matrix Types	Condition Numbers
Original Hessian	2.0042e+06
Preconditioned Hessian	2.5874e+04

Table 5.1: The comparison of condition numbers.

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. So we can solve the preconditioned system 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 5.4.

5.2.2 Regularization

With the help of our preconditioner, we can speed up an optimization algorithm and achieve higher accuracy. To further alleviate 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 instead 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 generalized Tikhonov regularization to smooth the edges. For other materials, we choose ℓ_1 regularizations to penalize the sum of weights. Based on this idea, we can

represent the regularization term as a sum of m parts:

$$R(\boldsymbol{w}) = \sum_{i=1}^{m} \frac{\alpha_i}{2} R_i(\boldsymbol{w}_i), \tag{5.29}$$

where $\boldsymbol{w}_i$ is the i-th column of the weight matrix $\boldsymbol{W}$, $R_i(\boldsymbol{w}_i)$ 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. For example, in breast imaging the object is dominated by glandular and adipose tissue, and so if the aim is to determine the weights of these two materials, 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. For example, in the breast imaging situation, the object may contain small microcalcifications or areas highlighted by an iodine tracer. In this case, one can use generalized Tikhonov regularization for the dominating materials (e.g., glandular and adipose tissue) and a ℓ_1 regularization for the sparse material. We illustrate this with two materials, one that dominates, and one that is sparse:

$$R(\boldsymbol{w}) = \frac{\alpha_1}{2}\|\boldsymbol{L}\boldsymbol{w}_1\|_2^2 + \frac{\alpha_2}{2}\|\boldsymbol{w}_2\|_1, \tag{5.30}$$

where $\boldsymbol{L}$ is the specified discrete differential operator. If we add these regularization terms to the objective function in Equation (5.18), we can rewrite it as an augmented system:

$$\begin{aligned} \min_{\tilde{\boldsymbol{w}}} \quad & \left\| \begin{bmatrix} \frac{\sqrt{2}}{2}\boldsymbol{\Sigma}^{-\frac{1}{2}}\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) \\ \sqrt{\frac{\alpha_1}{2}}\tilde{\boldsymbol{L}} \end{bmatrix} \tilde{\boldsymbol{w}} - \begin{bmatrix} \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2 + \frac{\alpha_2}{2} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} (\boldsymbol{M} \otimes \boldsymbol{I})\, \tilde{\boldsymbol{w}} \\ \text{subject to} \quad & (\boldsymbol{M} \otimes \boldsymbol{I})\, \tilde{\boldsymbol{w}} \geqslant \boldsymbol{0}, \end{aligned} \tag{5.31}$$

where $\tilde{\boldsymbol{L}} = \begin{bmatrix} \boldsymbol{L} & \boldsymbol{0} \end{bmatrix} (\boldsymbol{M} \otimes \boldsymbol{I})$. 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 nonsmooth. Because of these properties, we can think about using FISTA [3] 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.

5.3 FISTA and Projections

In this section, we first present the main algorithm FISTA briefly. To implement FISTA to solve the target optimization problem, we need to decide the step size and handle the nonnegative constraints. For the step size, we introduce how to compute the Lipschitz constant numerically then we 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 delicate decomposition and efficient algorithms.

5.3.1 FISTA

Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) is a first order method that belongs to the family of Iterative Shrinkage-Thresholding Algorithm (ISTA) [13]. This method is proposed by Beck et al., and compared with the $\mathcal{O}(1/k)$ 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. In most situations, FISTA is considered to be the "optimal" first order method with respect to convergence speed. Moreover, it best fits the problems in imaging science because it is usually used to solve the nonsmooth convex problem

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) + g(\boldsymbol{x}), \tag{5.32}$$

where $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are both convex functions and $g(\boldsymbol{x})$ might not be smooth. In imaging sciences, $f(\boldsymbol{x})$ is likely to be a least squares loss function to test the goodness of fit and $g(\boldsymbol{x})$ can be a regularization term such as ℓ_1 penalty or total variation regularization.

For Problem (5.31), we construct an augmented loss function that merges the generalized Tikhonov regularization term, which corresponds to $f(\boldsymbol{x})$ in (5.32). For the regularization term, ℓ_1 regularization is nonsmooth but convex and this matches $g(\boldsymbol{x})$ in (5.32).

The details of this algorithm are shown in Algorithm 4. For the main algorithm, we need to calculate 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 boundaries to keep the solution feasible. We would like to implement FISTA with a constant step size to solve the optimization problem (5.31). To implement this method, we need several preparations.

Algorithm 4 FISTA and Projections [3]

Initialization:
Calculate the smallest Lipschitz constant K in (5.34) by the power method.
Set up initial guess $\tilde{\boldsymbol{W}}_0$; Let $\boldsymbol{y}_0 = \mathrm{vec}(\tilde{\boldsymbol{W}}_0)$, $\boldsymbol{x}_{old} = \boldsymbol{y}_0$ and $t_1 = 1$;
for $k = 1,\ 2,\ \cdots$ **do**
    Calculate the gradients, $\nabla f(\boldsymbol{y}_k)$ and $\nabla g(\boldsymbol{y}_k)$, of $f(\boldsymbol{y}_k)$ and $g(\boldsymbol{y}_k)$ in (5.33);
    $\boldsymbol{x}_k = \boldsymbol{y}_k - \frac{1}{L(f)}\left[\nabla f(\boldsymbol{y}_k) + \nabla g(\boldsymbol{y}_k)\right]$;
    Reshape $\boldsymbol{x}_k$ into a matrix and use CVXGEN to solve the projection problems to obtain $\boldsymbol{x}_{new}$ as (5.37);
    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$;
    $\boldsymbol{y}_{k+1} = \boldsymbol{x}_{new} + \left(\frac{t_k - 1}{t_{k+1}}\right)(\boldsymbol{x}_{new} - \boldsymbol{x}_{old})$;
    $\boldsymbol{x}_{old} = \boldsymbol{x}_{new}$.

5.3.2 Lipschitz Constant

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

$$f(\tilde{\boldsymbol{w}}) = \left\| \begin{bmatrix} \frac{\sqrt{2}}{2}\boldsymbol{\Sigma}^{-\frac{1}{2}}\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) \\ \sqrt{\frac{\alpha_1}{2}}\tilde{\boldsymbol{L}} \end{bmatrix} \tilde{\boldsymbol{w}} - \begin{bmatrix} \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2, \tag{5.33}$$

$$g(\tilde{\boldsymbol{w}}) = \frac{\alpha_2}{2}\begin{bmatrix} \boldsymbol{0} & \boldsymbol{1} \end{bmatrix}(\boldsymbol{M} \otimes \boldsymbol{I})\,\tilde{\boldsymbol{w}},$$

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

$$K = \lambda_{max}\left[\left(\tilde{\boldsymbol{C}}^T \otimes \boldsymbol{A}^T\right)\boldsymbol{\Sigma}^{-1}\left(\tilde{\boldsymbol{C}} \otimes \boldsymbol{A}\right) + \alpha_1 \tilde{\boldsymbol{L}}^T\tilde{\boldsymbol{L}}\right]. \tag{5.34}$$

Since we only need the largest eigenvalue, it is not necessary for us to construct these matrices explicitly; instead we can use an iterative method, such as the power method [24]. Note that we only need to compute K once for all FISTA iterations. The details of the power method are shown in Algorithm 5.

Algorithm 5 Power Method [24]

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

5.3.3 Projections

In addition to the Lipschitz constant, we also need to handle the linear inequality constraints $(\boldsymbol{M} \otimes \boldsymbol{I})\,\tilde{\boldsymbol{w}} \geqslant \boldsymbol{0}$. Generally speaking, we can regard Problem (5.31) as a quadratic programming problem under these specific constraints. To impose the linear inequality constraints, we can construct another quadratic programming problem that can find the nearest solution to satisfy these constraints. If we assume that we have obtained $\tilde{\boldsymbol{w}}_k$ in the k-th step, then we can build a projection problem:

$$\begin{aligned} \min_{\tilde{\boldsymbol{w}}_{new}} \quad & \|\tilde{\boldsymbol{w}}_{new} - \tilde{\boldsymbol{w}}_k\|_2^2 \\ \text{subject to} \quad & (\boldsymbol{M} \otimes \boldsymbol{I})\,\tilde{\boldsymbol{w}}_{new} \geqslant \boldsymbol{0}. \end{aligned} \tag{5.35}$$

For small and medium size problems, we can solve it efficiently by direct implementation of standard optimization algorithms. For example, we can use CVX [26, 27] to solve Prob-

lem (5.35), which turns to be low-cost both in storage and computational 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 (5.35) into small pieces and try to solve each small-sized 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 (5.35) is equivalent to

$$\begin{aligned} \min_{\tilde{\boldsymbol{W}}_{new}} \quad & \left\| \tilde{\boldsymbol{W}}_{new} - \tilde{\boldsymbol{W}}_k \right\|_F^2 \\ \text{subject to} \quad & \tilde{\boldsymbol{W}}_{new} \boldsymbol{M}^T \geqslant \boldsymbol{0}. \end{aligned} \tag{5.36}$$

If we focus on each row of $\tilde{\boldsymbol{W}}_k$, $\tilde{\boldsymbol{W}}_k\left(i,\ :\right)$, then Problem (5.36) can be rewritten as

$$\begin{aligned} \min_{\tilde{\boldsymbol{W}}_{new}} \quad & \sum_{i=1}^{N_v} \left\| \tilde{\boldsymbol{W}}_{new}\left(i,\ :\right) - \tilde{\boldsymbol{W}}_k\left(i,\ :\right) \right\|_2^2 \\ \text{subject to} \quad & \tilde{\boldsymbol{W}}_{new}\left(i,\ :\right) \boldsymbol{M}^T \geqslant \boldsymbol{0}, \quad i = 1, 2, \cdots, N_v, \end{aligned} \tag{5.37}$$

where $\tilde{\boldsymbol{W}}_{new}\left(i,\ :\right)$ is the i-th row of $\tilde{\boldsymbol{W}}_{new}$. It is obvious that this problem is separable, and the original problem (5.36) 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 problem is usually 1×2 or 1×3. In this case, we can solve each small-sized problem efficiently and concatenate the solutions into a large matrix later. 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 chapter, we choose CVXGEN [41–44] 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 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 using FISTA in the main loop. Finally, the projection problem is decomposed into many small pieces and each can be solved by CVXGEN efficiently.

5.4 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.

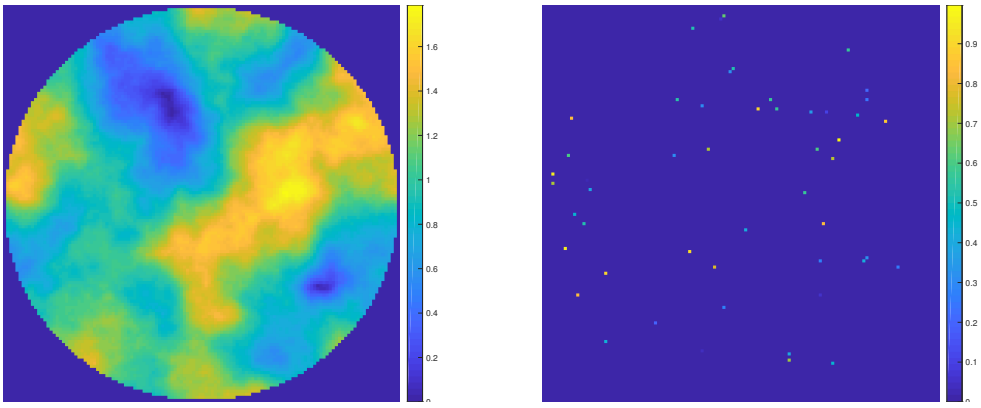

Figure 5.1: The original material maps for plexiglass (Left) and PVC (Right).

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.

In addition to the test images, we also need other parameters in Equation (5.1). To generate the ray trace matrix $\boldsymbol{A}$, we use the MATLAB function `fanbeamtomolinear` from AIR Tools [29,32,35] to simulate a fan-beam geometry with a flat detector. The parameters that we need to choose are presented in Table 4.1 in Chapter 4. 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` [53] 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.

The plot of photon flux density versus photon energy is presented in Figure 4.4. In Figure 4.4, 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.

The curves of linear attenuation coefficients versus photon energies are presented in Figure 4.3 in Chapter 4. From Figure 4.3, 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.

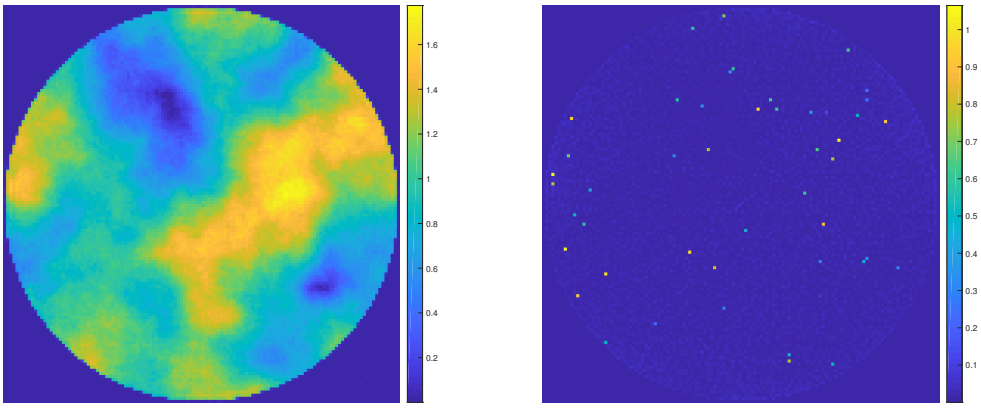

Figure 5.2: The reconstructed images for plexiglass (Left) and PVC (Right).

The reconstructed images are shown in Figure 5.2. From Figure 5.2, 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 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 the methods proposed in this chapter.

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 5.3. From this figure, we can see that the relative 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

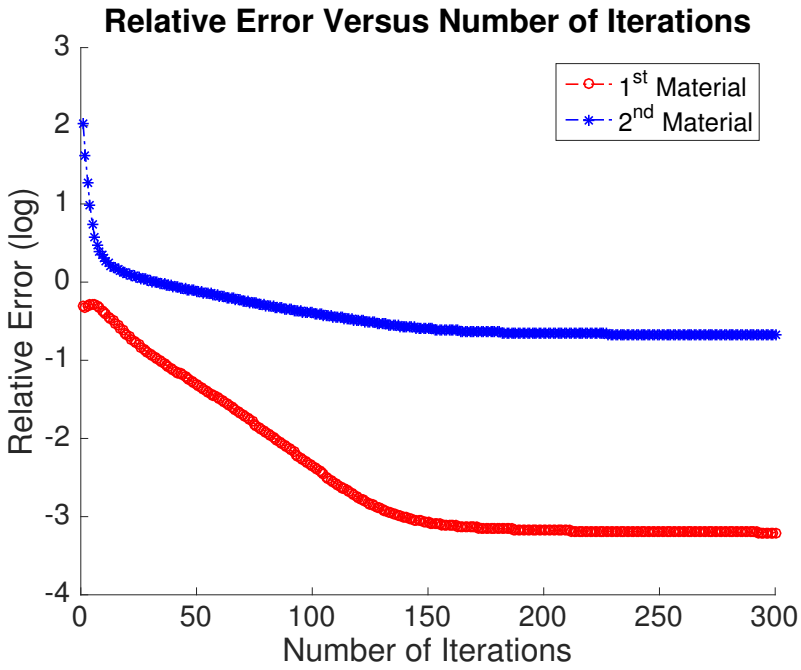

Figure 5.3: The related errors of each iteration (with preconditioner) for plexiglass and PVC.

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 reconstructed image, causing somewhat large relative errors, even though visually the result looks quite good.

Other accuracy measures illustrate this phenomenon. In Figure 5.4 we plot the mean squared error (MSE) at each iteration, in Figure 5.5 we plot the peak signal to noise ratio (PSNR), and in Figure 5.6 we plot the structural similarity index (SSIM). Not surprisingly the MSE produces information very similar to the relative errors, but it also shows a clear diminution for the second material from Figure 5.4. PSNR provides a similar measure to MSE, with an inverted interpretation (higher values correspond to better solutions). The SSIM is a metric for image quality, and as with PSNR, large values correspond to better solutions. From Figure 5.6, 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

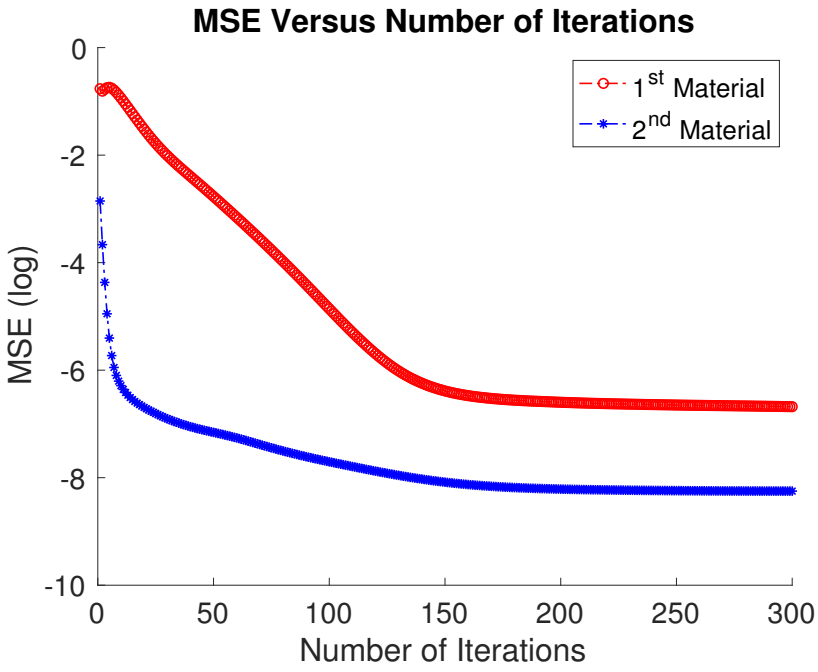

Figure 5.4: The MSE of each iteration (with preconditioner) for plexiglass and PVC.

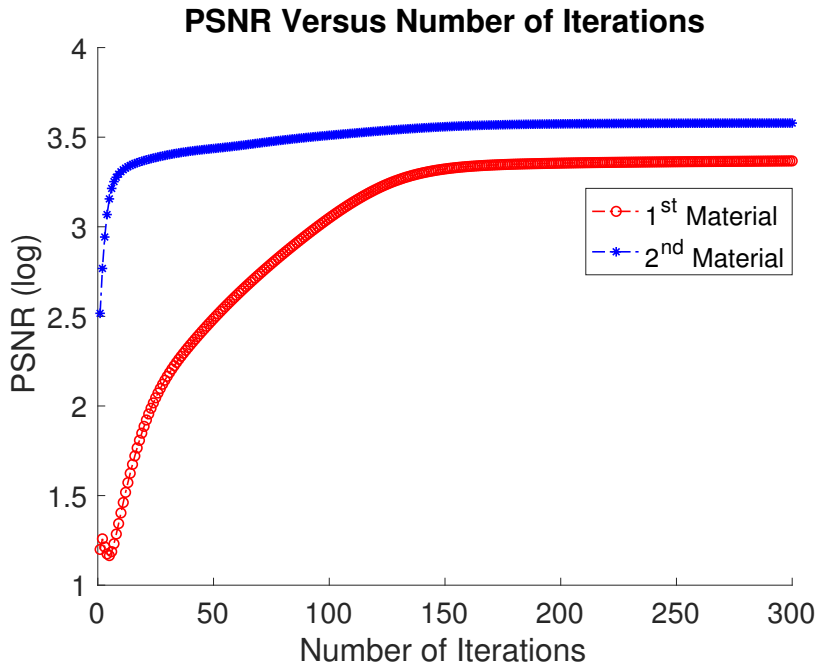

Figure 5.5: The PSNR of each iteration (with preconditioner) for plexiglass and PVC.

and quality measures illustrate fast convergence to high quality reconstructions.

It may also be of interest to observe the decay of norm of the gradient at each iteration, which is shown in Figure 5.7. From this figure we can see that the norm of the gradient

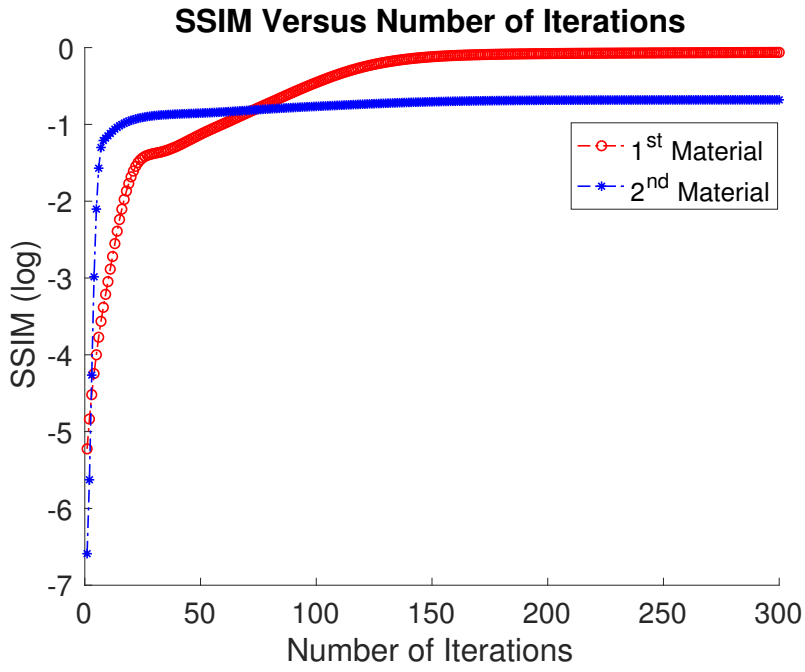

Figure 5.6: The SSIM of each iteration (with preconditioner) for plexiglass and PVC.

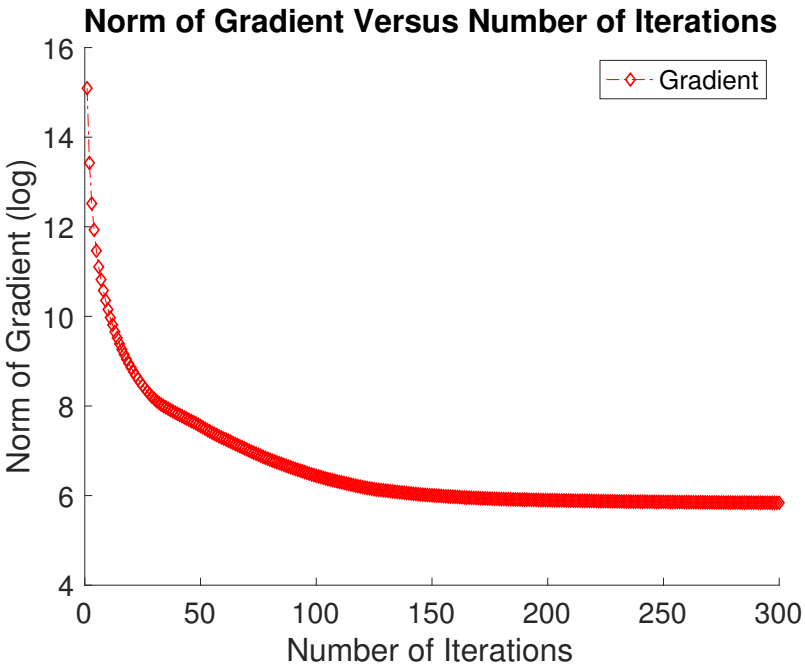

Figure 5.7: The decay of norm of the gradient for overall materials.

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 perfor-

mance with a preconditioner proposed by Barber [2], and the performance without using any preconditioners. As previously mentioned, the approach proposed in [2] is based on the eigenvalue decomposition of $\boldsymbol{C}^T\boldsymbol{C}$. The results are shown in Figure 5.8, 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 second material is the same. From

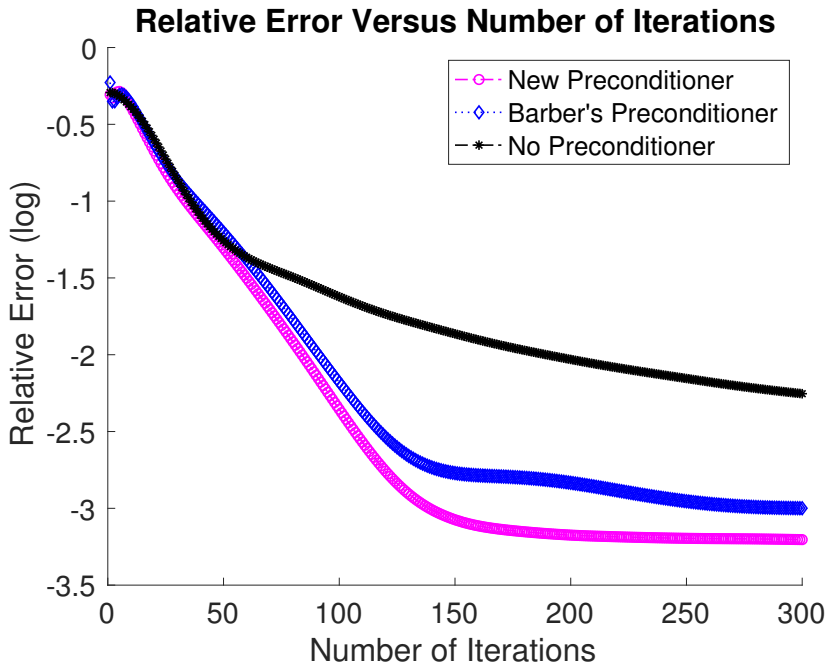

Figure 5.8: The decay of related errors with new preconditioner, Barber's [2] preconditioner, and with no preconditioner.

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.

5.5 Conclusions and Remarks

In this chapter, 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 pro-

jected 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 chapter. For future work, it would be interesting to consider other regularization schemes to emphasize the edges of the object, such as the total variation.

Chapter 6

A Two-Step Method for Energy-windowed Spectral Computed Tomography

In this chapter, we still focus on the energy-windowed spectral computed tomography model

$$\boldsymbol{Y} = \exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}, \tag{6.1}$$

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 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 constraint $\boldsymbol{W} \geqslant \mathbf{0}$.

In the previous chapters, we try to compute the solution directly. However, we can also introduce an auxiliary variable and solve it using two steps. If we let $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{W}$, then a two-step model is expressed as

$$\begin{aligned} \boldsymbol{Y} &= \exp\left(-\boldsymbol{X}\boldsymbol{C}^T\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}, \\ \boldsymbol{X} &= \boldsymbol{A}\boldsymbol{W}, \quad \boldsymbol{W} \geqslant \mathbf{0}. \end{aligned} \tag{6.2}$$

Based on the Gaussian assumption of noise, we can repeat the process of Chapter 5 to develop a weighted least squares framework. However, we do not want to solve the entire model using an iterative method. Instead, we want to construct a series of small-sized problems and solve each of them efficiently and accurately. Recall that when we implement the Algebraic Reconstruction Technique (ART) to solve a least squares problem, the iteration format is based on each row. Inspired by this idea, we want to come up with a similar method that takes advantage of each row of a weighted least squares problem. In the first step, we can build a row-wise optimization problem of small-sized and solve each one with cheap cost. In the second step, we can sum up the results obtained from the first step and solve a linear system under bound constraints. A challenge comes from the noise propagation, where we have to quantify the noise in the linear system of the second step. This quantity can be obtained using the properties of linear transform of the covariance matrix in the first step and we can use it to build another weighted least squares problem.

We should also mention that splitting the original problem into two steps and solving it in sequence might put the solution away from the truth. Instead, we can solve (6.2) directly by constructing an optimization problem of two coupled terms,

$$\begin{aligned} \min_{\bar{\boldsymbol{x}},\, \bar{\boldsymbol{w}}} \quad & \frac{1}{2}\|(\boldsymbol{I} \otimes \boldsymbol{C})\,\bar{\boldsymbol{x}} - \hat{\boldsymbol{b}}\|^2_{\boldsymbol{\Sigma}^{-1}_{\hat{\boldsymbol{b}}}} + \frac{1}{2}\,\|(\boldsymbol{A} \otimes \boldsymbol{I})\,\bar{\boldsymbol{w}} - \bar{\boldsymbol{x}}\|^2_{\boldsymbol{\Sigma}^{-1}_{\bar{\boldsymbol{x}}_{\mathrm{ml}}}}, \\ \text{subject to} \quad & \bar{\boldsymbol{w}} \geqslant \mathbf{0}, \end{aligned} \tag{6.3}$$

where $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{w}}$ are vectorizations of $\boldsymbol{X}^T$ and $\boldsymbol{W}^T$, respectively, and $\boldsymbol{\Sigma}^{-1}_{\hat{\boldsymbol{b}}}$ and $\boldsymbol{\Sigma}^{-1}_{\bar{\boldsymbol{x}}_{\mathrm{ml}}}$ are noise covariance matrices. Compared with the two-step method, the solution to (6.3) can

be obtained using the traditional iterative scheme and the solution is likely to be closer to the true solution of (6.2). On the other hand, it is likely that the solution to problem (6.3) might be better in theory, but it can be less attractive in terms of image quality. In this case, we should conduct numerical experiments to demonstrate strengths and weaknesses of these methods.

This chapter is organized as follows. In section (6.1), we review the limitations of previous methods and set up the two-step model. How to obtain the solution in two steps is also explained in this section. In section (6.2), we discuss another method, the coupled method, to solve the two-step model. This method combines the two steps together and uses an iterative framework to compute the solution. Numerical experiments are presented in (6.3). Concluding remarks and comparisons with the previous method are discussed in (6.4).

6.1 The Two-step Method

6.1.1 The Framework of Two-step Model

Compared with traditional CT models, the energy-windowed spectral CT model expands the room for developing mathematical theories and computing numerical solutions. In Chapter 4, the basic equation related to this model is reformulated as a nonlinear least squares problem and solved by nonlinear optimization. In Chapter 5, we have transformed the basic equation into a weighted least squares problem, developed a new preconditioner and used FISTA to compute the solution. Even if these two methods are based on different objective functions, they both include a preconditioner to mitigate the influence of ill-posedness. So it raises a question if we can solve this equation efficiently to a high accuracy without using preconditioners. In this section, we will focus on a two-step method which can calculate the solution to the weighted least squares model accurately and economically.

Recall that the discretized energy-windowed spectral CT model is expressed as

$$\boldsymbol{Y} = \exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}, \tag{6.4}$$

where

- $\boldsymbol{Y}$ is a matrix of 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 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 size $N_e \times N_m$ that accumulates linear attenuation coefficients and each entry corresponds to $u_{e,m}$, the linear attenuation coefficient of e-th energy and m-th material.
- $\boldsymbol{W}$ is a matrix of size $N_v \times N_m$ and each column corresponds the unknown material map we want to reconstruct.
- $\boldsymbol{S}$ is a matrix of size $N_e \times N_b$ and each column collects the spectrum energy of a specific range. We assume that $\boldsymbol{S}$ is square and invertible.
- $\boldsymbol{\mathcal{E}}$ is the noise matrix that is of size $(N_d \cdot N_p) \times N_b$. The assumption for noise is that $E_{il} \sim \mathcal{N}(0, y_{il})$ for each element E_{il} in $\boldsymbol{\mathcal{E}}$ and y_{il} in $\boldsymbol{Y}$.

As we have shown in Chapter 5, we can vectorize (6.4) and build a weighted least squares problem using the Gaussian assumption of noise

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2}\|\mathcal{A}\boldsymbol{w} - \boldsymbol{b}\|^2_{\boldsymbol{\Sigma}^{-1}} \\ \text{subject to} \quad & \boldsymbol{w} \geqslant \boldsymbol{0}, \end{aligned} \tag{6.5}$$

where $\mathcal{A} = \boldsymbol{C} \otimes \boldsymbol{A}$, $\boldsymbol{b} = -\log(\boldsymbol{y})$, $\boldsymbol{y} = \text{vec}(\boldsymbol{Y})$ and $\boldsymbol{w} = \text{vec}(\boldsymbol{W})$. $\boldsymbol{\Sigma}^{-1}$, which combines information from $\boldsymbol{S}$ and $\boldsymbol{y}$, is the inverse covariance matrix generated by the Gaussian noise and logarithmic transformation. $\|\cdot\|^2_{\boldsymbol{\Sigma}^{-1}}$ represents a weighted 2-norm and $\|\mathcal{A}\boldsymbol{w} - \boldsymbol{b}\|^2_{\boldsymbol{\Sigma}^{-1}} = (\mathcal{A}\boldsymbol{w} - \boldsymbol{b})^T \boldsymbol{\Sigma}^{-1} (\mathcal{A}\boldsymbol{w} - \boldsymbol{b})$. Because of the collinearity of $\boldsymbol{C}$ and the sparsity of $\boldsymbol{A}$, solutions to Equation (6.5) might not be satisfactory. In Chapter 5, we introduce a preconditioner

$\boldsymbol{M}$ and the optimization problem (6.5) has been transformed into

$$\begin{aligned} \min_{\tilde{\boldsymbol{w}}} \quad & \frac{1}{2}\|\tilde{\boldsymbol{A}}\tilde{\boldsymbol{w}} - \boldsymbol{b}\|^2_{\boldsymbol{\Sigma}^{-1}} \\ \text{subject to} \quad & (\boldsymbol{M} \otimes \boldsymbol{I})\,\tilde{\boldsymbol{w}} \geqslant \boldsymbol{0}. \end{aligned} \tag{6.6}$$

where $\tilde{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{M}$, $\tilde{\mathcal{A}} = \tilde{\boldsymbol{C}} \otimes \boldsymbol{A}$, $\tilde{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{M}^{-T}$ and $\tilde{\boldsymbol{w}} = \text{vec}(\tilde{\boldsymbol{W}})$. To solve problem (6.6), we implement FISTA to solve the main problem and build a projection subproblem (6.7) to maintain nonnegativity

$$\begin{aligned} \min_{\tilde{\boldsymbol{w}}_{new}} \quad & \|\tilde{\boldsymbol{w}}_{new} - \tilde{\boldsymbol{w}}_k\|_2^2 \\ \text{subject to} \quad & (\boldsymbol{M} \otimes \boldsymbol{I})\,\tilde{\boldsymbol{w}}_{new} \geqslant \boldsymbol{0}, \end{aligned} \tag{6.7}$$

where $\tilde{\boldsymbol{w}}_k$ is the current iteration of $\tilde{\boldsymbol{w}}$. To compute the solution to subproblem (6.7), we reshape all variables back to a matrix form and split this form into several pieces. Each piece is a tiny size problem with a row in the original matrix and the corresponding constraints. For each problem, we solve it using a high performance solver such as CVXGEN. Even if this method is efficient for large-scale problems, solving the projection problem might still be a very expensive cost. So we want to figure out a method that can both solve the problem (6.4) accurately and avoid large expense on keeping the nonnegativity of the solution.

Rather than computing the solution directly, we propose a two-step method with an auxiliary step and bridge it to the desired solution. Recall that our goal is to solve $\boldsymbol{W}$ from the equation

$$\boldsymbol{Y} = \exp\left(-\boldsymbol{A}\boldsymbol{W}\boldsymbol{C}^T\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}. \tag{6.8}$$

We first let $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{W}$ and solve $\boldsymbol{X}$ from Equation (6.9)

$$\boldsymbol{Y} = \exp\left(-\boldsymbol{X}\boldsymbol{C}^T\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}. \tag{6.9}$$

After we have obtained $\boldsymbol{X}$, we solve $\boldsymbol{W}$ from the equation

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{W}, \quad \boldsymbol{W} \geqslant \boldsymbol{0}. \tag{6.10}$$

Recall that the size of $\boldsymbol{W}$ is $N_v \times N_m$ and the auxiliary step is bound to $\boldsymbol{X}$, which is of size $(N_d \cdot N_p) \times N_m$. Usually, $N_d \cdot N_p$ is much larger than N_v so we seem to introduce more unknown variables and this behavior is undesirable. Another problem we need to consider is how to quantify the noise propagation from Equation (6.9) to Equation (6.10). In Equation (6.9), we solve $\boldsymbol{X}$ with the noise $\boldsymbol{\mathcal{E}}$ so the solution must include probability properties of $\boldsymbol{\mathcal{E}}$. When we solve $\boldsymbol{W}$ from (6.10), the variable $\boldsymbol{X}$ is assumed to be known so we should also identify the noise.

6.1.2 A Solution to the Two-step Model

To solve the first problem, we try to use the statistical properties of noise to decompose the entire problem into small pieces and solve each of these problems accurately. Recall that $\boldsymbol{\mathcal{E}}$ is the noise matrix and $E_{il} \sim \mathcal{N}(0, y_{il})$ for each element E_{il} in $\boldsymbol{\mathcal{E}}$ and y_{il} in $\boldsymbol{Y}$, so the noise is independent for each entry of projected data. With this independence, we can decompose Equation (6.9) in a row-wise or column-wise way. Since $\boldsymbol{Y}$ is of size $(N_d \cdot N_p) \times N_b$ and the number of energy windows, N_b, is normally much fewer than $N_d \cdot N_p$, the splitting over rows is preferable.

If we let

$$\begin{aligned} \boldsymbol{Y}^T &= \left[\boldsymbol{y}_1,\ \boldsymbol{y}_2,\ \cdots, \boldsymbol{y}_{N_d \cdot N_p}\right], \\ \boldsymbol{X}^T &= \left[\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \cdots, \boldsymbol{x}_{N_d \cdot N_p}\right], \\ \boldsymbol{\mathcal{E}}^T &= \left[\boldsymbol{e}_1,\ \boldsymbol{e}_2,\ \cdots, \boldsymbol{e}_{N_d \cdot N_p}\right], \end{aligned} \tag{6.11}$$

then Equation (6.9) can be rewritten with respect to each row

$$\boldsymbol{y}_i^T = \exp\left(-\boldsymbol{x}_i^T \boldsymbol{C}^T\right) \boldsymbol{S} + \boldsymbol{e}_i^T, \tag{6.12}$$

for $i = 1,\ 2,\ \cdots, N_d \cdot N_p$. Meanwhile, we can notice that each row of $\boldsymbol{Y}$ only depends on the corresponding row of $\boldsymbol{X}$ and $\boldsymbol{\mathcal{E}}$ as well as matrices $\boldsymbol{C}$ and $\boldsymbol{S}$. If we take the transpose

for Equation (6.12), then we have

$$\boldsymbol{y}_i = \boldsymbol{S}^T \exp\left(-\boldsymbol{C}\boldsymbol{x}_i\right) + \boldsymbol{e}_i. \tag{6.13}$$

By assumption, $\boldsymbol{S}$ is square and invertible, so we can multiply $\boldsymbol{S}^{-T}$ on both sides

$$\boldsymbol{S}^{-T}\boldsymbol{y}_i = \exp\left(-\boldsymbol{C}\boldsymbol{x}_i\right) + \boldsymbol{S}^{-T}\boldsymbol{e}_i. \tag{6.14}$$

If we let $\hat{\boldsymbol{y}}_i = \boldsymbol{S}^{-T}\boldsymbol{y}_i$ and $\hat{\boldsymbol{e}}_i = \boldsymbol{S}^{-T}\boldsymbol{e}_i$, then Equation (6.14) is equivalent to

$$\hat{\boldsymbol{y}}_i = \exp\left(-\boldsymbol{C}\boldsymbol{x}_i\right) + \hat{\boldsymbol{e}}_i. \tag{6.15}$$

By subtracting $\hat{\boldsymbol{e}}_i$ on both sides and taking the logarithm, we have

$$\log\left(\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{e}}_i\right) = -\boldsymbol{C}\boldsymbol{x}_i. \tag{6.16}$$

Using the Taylor expansion, we can expand the left hand side of Equation (6.16) as

$$\log\left(\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{e}}_i\right) = \log\left(\hat{\boldsymbol{y}}_i\right) - \operatorname{diag}\left(\hat{\boldsymbol{y}}_i\right)^{-1}\hat{\boldsymbol{e}}_i + \mathcal{O}\left(\|\hat{\boldsymbol{e}}_i\|_2^2\right). \tag{6.17}$$

Therefore, Equation (6.16) can be estimated as

$$\hat{\boldsymbol{b}}_i \approx \boldsymbol{C}\boldsymbol{x}_i - \operatorname{diag}\left(\hat{\boldsymbol{y}}_i\right)^{-1}\hat{\boldsymbol{e}}_i. \tag{6.18}$$

where $\hat{\boldsymbol{b}}_i = -\log\left(\hat{\boldsymbol{y}}_i\right)$. By assumption, we have $\boldsymbol{e}_i \sim \mathcal{N}\left(\boldsymbol{0},\ \operatorname{diag}\left(\boldsymbol{y}_i\right)\right)$. Then we can obtain that

$$\hat{\boldsymbol{b}}_i | \boldsymbol{x}_i \sim \mathcal{N}\left(\boldsymbol{C}\boldsymbol{x}_i, \boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}_i}\right), \tag{6.19}$$

where $\boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}_i} = \operatorname{diag}\left(\hat{\boldsymbol{y}}_i\right)^{-1}\boldsymbol{S}^{-T}\operatorname{diag}\left(\boldsymbol{y}_i\right)\boldsymbol{S}^{-1}\operatorname{diag}\left(\hat{\boldsymbol{y}}_i\right)^{-1}$. By ignoring constants, the corresponding probability density function can be expressed as

$$f\left(\hat{\boldsymbol{b}}_i; \boldsymbol{x}_i\right) = \exp\left\{-\frac{1}{2}\left(\boldsymbol{C}\boldsymbol{x}_i - \hat{\boldsymbol{b}}_i\right)^T \boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}_i}^{-1}\left(\boldsymbol{C}\boldsymbol{x}_i - \hat{\boldsymbol{b}}_i\right)\right\}. \tag{6.20}$$

So the log-likelihood function is given by

$$l\left(\boldsymbol{x}_i;\hat{\boldsymbol{b}}_i\right)=-\frac{1}{2}\left(\boldsymbol{C}\boldsymbol{x}_i-\hat{\boldsymbol{b}}_i\right)^T\boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}_i}^{-1}\left(\boldsymbol{C}\boldsymbol{x}_i-\hat{\boldsymbol{b}}_i\right). \tag{6.21}$$

Our goal is to maximize the log-likelihood function $l\left(\boldsymbol{x}_i;\hat{\boldsymbol{b}}_i\right)$ and it is equivalent to minimizing the negative log-likelihood function $-l\left(\boldsymbol{x}_i;\hat{\boldsymbol{b}}_i\right)$. Based on Equation (6.21), the maximum likelihood estimator for $\boldsymbol{x}_i$ can be represented as

$$\left(\boldsymbol{x}_i\right)_{\mathrm{ml}}=\underset{\boldsymbol{x}_i}{\operatorname{argmin}}\left\{\left\|\boldsymbol{C}\boldsymbol{x}_i-\hat{\boldsymbol{b}}_i\right\|_{\boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}_i}^{-1}}^2\right\}. \tag{6.22}$$

This equation can be solved analytically and the corresponding solution is

$$\left(\boldsymbol{x}_i\right)_{\mathrm{ml}}=\left(\boldsymbol{C}^T\boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}_i}^{-1}\boldsymbol{C}\right)^{-1}\boldsymbol{C}^T\boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}_i}^{-1}\hat{\boldsymbol{b}}_i. \tag{6.23}$$

Since the matrix $\boldsymbol{C}$ is of size N_e by N_m, then solving each $\left(\boldsymbol{x}_i\right)_{\mathrm{ml}}$ is of low cost. So for the first step, we can loop around all $i=1,\ 2,\ \cdots,N_d\cdot N_p$ and solve each linear system to a high accuracy. After we have obtained all $\left(\boldsymbol{x}_i\right)_{\mathrm{ml}}$, we can build the maximum likelihood solution for the matrix $\boldsymbol{X}$ by concatenation

$$\boldsymbol{X}_{ml}^T=\left[\left(\boldsymbol{x}_1\right)_{\mathrm{ml}},\ \left(\boldsymbol{x}_2\right)_{\mathrm{ml}},\ \cdots,\left(\boldsymbol{x}_{N_d\cdot N_p}\right)_{\mathrm{ml}}\right]. \tag{6.24}$$

So far we have obtained the maximum likelihood solution, $\boldsymbol{X}_{\mathrm{ml}}^T$, for Equation (6.9). However, we cannot substitute it for the same variable in Equation (6.10) because of the unknown noise.

To quantify the noise propagation, we repeat a similar process as we have shown in Chapter 5. To be consistent with the row-wise model, we first take a transpose of Equation (6.9):

$$\boldsymbol{Y}^T=\boldsymbol{S}^T\exp\left(-\boldsymbol{C}\boldsymbol{X}^T\right)+\boldsymbol{\mathcal{E}}^T. \tag{6.25}$$

Multiplying $\boldsymbol{S}^{-T}$ on both sides, we can obtain that

$$\boldsymbol{S}^{-T}\boldsymbol{Y}^{T} = \exp\left(-\boldsymbol{C}\boldsymbol{X}^{T}\right) + \boldsymbol{S}^{-T}\boldsymbol{\mathcal{E}}^{T}. \tag{6.26}$$

By vectorizing the variables on both sides, we have

$$\left(\boldsymbol{I} \otimes \boldsymbol{S}^{-T}\right)\bar{\boldsymbol{y}} = \exp\left(-\left(\boldsymbol{I} \otimes \boldsymbol{C}\right)\bar{\boldsymbol{x}}\right) + \left(\boldsymbol{I} \otimes \boldsymbol{S}^{-T}\right)\bar{\boldsymbol{e}}, \tag{6.27}$$

where $\bar{\boldsymbol{y}} = \operatorname{vec}\left(\boldsymbol{Y}^{T}\right)$, $\bar{\boldsymbol{x}} = \operatorname{vec}\left(\boldsymbol{X}^{T}\right)$ and $\bar{\boldsymbol{e}} = \operatorname{vec}\left(\boldsymbol{\mathcal{E}}^{T}\right)$. Using the former notations, we define $\hat{\boldsymbol{y}} = \left(\boldsymbol{I} \otimes \boldsymbol{S}^{-T}\right)\bar{\boldsymbol{y}}$, $\hat{\boldsymbol{e}} = \left(\boldsymbol{I} \otimes \boldsymbol{S}^{-T}\right)\bar{\boldsymbol{e}}$ and $\hat{\boldsymbol{b}} = -\log(\hat{\boldsymbol{y}})$. Then Equation (6.27) can be written as

$$\hat{\boldsymbol{y}} = \exp\left(-\left(\boldsymbol{I} \otimes \boldsymbol{C}\right)\bar{\boldsymbol{x}}\right) + \hat{\boldsymbol{e}}. \tag{6.28}$$

We take the logarithm as in (6.16) and expand the logarithmic term as Equation (6.17), then Equation (6.28) can be approximated as

$$\hat{\boldsymbol{b}} \approx \left(\boldsymbol{I} \otimes \boldsymbol{C}\right)\bar{\boldsymbol{x}} - \operatorname{diag}\left(\hat{\boldsymbol{y}}\right)^{-1}\left(\boldsymbol{I} \otimes \boldsymbol{S}^{-T}\right)\bar{\boldsymbol{e}}, \tag{6.29}$$

By assumption, we assume that $\bar{\boldsymbol{e}} \sim \mathcal{N}\left(\boldsymbol{0},\ \operatorname{diag}\left(\bar{\boldsymbol{y}}\right)\right)$. Then we have

$$\hat{\boldsymbol{b}}|\bar{\boldsymbol{x}} \sim \mathcal{N}\left(\left(\boldsymbol{I} \otimes \boldsymbol{C}\right)\bar{\boldsymbol{x}},\ \boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}}\right), \tag{6.30}$$

where $\boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}} = \operatorname{diag}\left(\hat{\boldsymbol{y}}\right)^{-1}\left(\boldsymbol{I} \otimes \boldsymbol{S}^{-T}\right)\operatorname{diag}\left(\bar{\boldsymbol{y}}\right)\left(\boldsymbol{I} \otimes \boldsymbol{S}^{-1}\right)\operatorname{diag}\left(\hat{\boldsymbol{y}}\right)^{-1}$ is the block-diagonal covariance matrix. If we build the maximum likelihood function with $\boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}}$, then the maximum likelihood estimator of $\bar{\boldsymbol{x}}$ can be expressed as

$$\begin{aligned}\bar{\boldsymbol{x}}_{\mathrm{ml}} &= \underset{\boldsymbol{x}}{\operatorname{argmin}}\left\{\frac{1}{2}\|\left(\boldsymbol{I} \otimes \boldsymbol{C}\right)\boldsymbol{x} - \hat{\boldsymbol{b}}\|^{2}_{\boldsymbol{\Sigma}^{-1}_{\hat{\boldsymbol{b}}}}\right\} \\ &= \left(\left(\boldsymbol{I} \otimes \boldsymbol{C}^{T}\right)\boldsymbol{\Sigma}^{-1}_{\hat{\boldsymbol{b}}}\left(\boldsymbol{I} \otimes \boldsymbol{C}\right)\right)^{-1}\left(\boldsymbol{I} \otimes \boldsymbol{C}^{T}\right)\boldsymbol{\Sigma}^{-1}_{\hat{\boldsymbol{b}}}\hat{\boldsymbol{b}}.\end{aligned} \tag{6.31}$$

By using assumption (6.30), we know that $\bar{\boldsymbol{x}}_{\mathrm{ml}}$ is itself a random variable with mean

$(\boldsymbol{I} \otimes \boldsymbol{C})\,\bar{\boldsymbol{x}}$ and covariance

$$\boldsymbol{\Sigma}_{\bar{\boldsymbol{x}}_{\mathrm{ml}}} = \left(\left(\boldsymbol{I} \otimes \boldsymbol{C}^T \right) \Sigma_{\hat{\boldsymbol{b}}}^{-1} \left(\boldsymbol{I} \otimes \boldsymbol{C} \right) \right)^{-1}. \tag{6.32}$$

So far we have obtained the noise covariance matrix for the second step. With this covariance matrix, we can build another optimization problem under bound constraints. For the second step, we transpose the equation first in order to be consistent with $\bar{\boldsymbol{x}}$:

$$\boldsymbol{X}_{\mathrm{ml}}^T = \boldsymbol{W}^T \boldsymbol{A}^T. \tag{6.33}$$

Taking vectorization on both sides, we can obtain that

$$\bar{\boldsymbol{x}}_{\mathrm{ml}} = (\boldsymbol{A} \otimes \boldsymbol{I})\,\bar{\boldsymbol{w}}. \tag{6.34}$$

Using the noise covariance matrix (6.32), we can formulate a weighted least squares problem under bound constraints to solve $\bar{\boldsymbol{w}}$:

$$\begin{aligned} \underset{\bar{\boldsymbol{w}}}{\operatorname{argmin}} \quad & \frac{1}{2} \left\| (\boldsymbol{A} \otimes \boldsymbol{I})\,\bar{\boldsymbol{w}} - \bar{\boldsymbol{x}}_{\mathrm{ml}} \right\|^2_{\boldsymbol{\Sigma}^{-1}_{\bar{\boldsymbol{x}}_{\mathrm{ml}}}} \\ \text{subject to} \quad & \bar{\boldsymbol{w}} \geqslant \boldsymbol{0}. \end{aligned} \tag{6.35}$$

For the regularization term, we take ℓ_1 regularization to keep sparsity. So the optimization problem we need to solve is

$$\begin{aligned} \underset{\bar{\boldsymbol{w}}}{\operatorname{argmin}} \quad & \frac{1}{2} \left\| (\boldsymbol{A} \otimes \boldsymbol{I})\,\bar{\boldsymbol{w}} - \bar{\boldsymbol{x}}_{\mathrm{ml}} \right\|^2_{\boldsymbol{\Sigma}^{-1}_{\bar{\boldsymbol{x}}_{\mathrm{ml}}}} + \beta \|\bar{\boldsymbol{w}}\|_1 \\ \text{subject to} \quad & \bar{\boldsymbol{w}} \geqslant \boldsymbol{0}, \end{aligned} \tag{6.36}$$

where β is the regularization parameter. For problem (6.36), we solve it using FISTA with a projection step onto the boundary. The iteration involved in FISTA only requires the current gradient as well as the current and previous steps. The current and previous steps are generated iteratively while the current gradient can be computed using the properties of the Kronecker product. In this case, we only need a matrix-vector multiplication in each

iteration and thus can avoid forming the Hessian explicitly.

To conclude, this two-step method can damp the influence of the matrix $\boldsymbol{C}$ and it has a similar effect as the preconditioner in the last chapter. However, rather than building a preconditioned system, we solve several row-wise, small-size problems to high accuracy. Compared with the preconditioning method, it is not necessary to solve a projection problem in each iteration so the convergence speed is fast. Moreover, the solution of each auxiliary system does not depend on each other, so we can further implement the solver for the first step in parallel. On the other hand, the possible drawbacks come from the cost of solving the linear system in the first step, which might be increased significantly when the size of image is large. These limitations can also be mitigated by parallel computation.

6.2 The Coupled Method

In the last section, we have discussed how to solve Equation (6.4) using a two step method. We use an auxiliary variable $\boldsymbol{X}$ such that $\boldsymbol{X} = \boldsymbol{AW}$. Then we solve $\boldsymbol{X}$ in the first step and $\boldsymbol{W}$ in the second step. We derive the optimization problem using a log-likelihood function and use the Gaussian noise assumption to obtain the propagation of noise in the second step. Then we solve these two problems in sequence and obtain the final result. However, in each of these two steps, we might either overestimate or underestimate the solutions and the combined solution is likely to be distorted. To compare and evaluate the two-step method, we can come up with an integrated framework that merges these separated steps.

Instead of solving these two equations individually, we can try to combine previous two steps and calculate the solution alternatively. In each iteration, we solve the first equation and evaluate the result to the corresponding part in the second equation. Then we solve the second equation under the current circumstances. In this case, we have coupled the results generated from two separated steps and they might be influential to the following iterations. Moreover, each equation is still separable and we can use optimization techniques, such as the coordinate descent method, to update each step. In the previous inferences, we try to

solve the following two problems

$$\begin{aligned} \boldsymbol{Y} &= \exp\left(-\boldsymbol{X}\boldsymbol{C}^T\right)\boldsymbol{S} + \boldsymbol{\mathcal{E}}, \\ \boldsymbol{X} &= \boldsymbol{A}\boldsymbol{W}, \quad \boldsymbol{W} \geqslant \boldsymbol{0}. \end{aligned} \tag{6.37}$$

The noise covariance matrix is $\boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}^{-1}}$ for the first equation and for the second equation, the noise covariance matrix can be represented as $\boldsymbol{\Sigma}_{\bar{\boldsymbol{x}}_{\mathrm{ml}}^{-1}}$. Recall that the noise covariance matrix for the second step is based on the maximum likelihood estimate of the first step so if we couple these two problems, it is only based on the current maximum likelihood estimator of the first step. With these two covariance matrices, we can build two optimization problems

$$\begin{aligned} \min_{\bar{\boldsymbol{x}}} \quad & \frac{1}{2}\|(\boldsymbol{I}\otimes\boldsymbol{C})\,\bar{\boldsymbol{x}} - \hat{\boldsymbol{b}}\|^2_{\boldsymbol{\Sigma}^{-1}_{\hat{\boldsymbol{b}}}}, \\ \min_{\bar{\boldsymbol{w}}} \quad & \frac{1}{2}\left\|(\boldsymbol{A}\otimes\boldsymbol{I})\,\bar{\boldsymbol{w}} - \bar{\boldsymbol{x}}_{\mathrm{ml}}\right\|^2_{\boldsymbol{\Sigma}^{-1}_{\bar{\boldsymbol{x}}_{\mathrm{ml}}}} \\ \text{subject to} \quad & \bar{\boldsymbol{w}} \geqslant \boldsymbol{0}. \end{aligned} \tag{6.38}$$

where

$$\begin{aligned} \boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}} &= \operatorname{diag}(\hat{\boldsymbol{y}})^{-1}\left(\boldsymbol{I}\otimes\boldsymbol{S}^{-T}\right)\operatorname{diag}(\bar{\boldsymbol{y}})\left(\boldsymbol{I}\otimes\boldsymbol{S}^{-1}\right)\operatorname{diag}(\hat{\boldsymbol{y}})^{-1}, \\ \boldsymbol{\Sigma}_{\bar{\boldsymbol{x}}_{\mathrm{ml}}} &= \left(\left(\boldsymbol{I}\otimes\boldsymbol{C}^T\right)\Sigma^{-1}_{\hat{\boldsymbol{b}}}\left(\boldsymbol{I}\otimes\boldsymbol{C}\right)\right)^{-1}. \end{aligned} \tag{6.39}$$

These two problems can be merged into one optimization problem as

$$\begin{aligned} \min_{\bar{\boldsymbol{x}},\ \bar{\boldsymbol{w}}} \quad & \frac{1}{2}\|(\boldsymbol{I}\otimes\boldsymbol{C})\,\bar{\boldsymbol{x}} - \hat{\boldsymbol{b}}\|^2_{\boldsymbol{\Sigma}^{-1}_{\hat{\boldsymbol{b}}}} + \frac{1}{2}\left\|(\boldsymbol{A}\otimes\boldsymbol{I})\,\bar{\boldsymbol{w}} - \bar{\boldsymbol{x}}\right\|^2_{\boldsymbol{\Sigma}^{-1}_{\bar{\boldsymbol{x}}_{\mathrm{ml}}}}, \\ \text{subject to} \quad & \bar{\boldsymbol{w}} \geqslant \boldsymbol{0}. \end{aligned} \tag{6.40}$$

The nonnegative constraint $\bar{\boldsymbol{w}} \geqslant \boldsymbol{0}$ is equivalent to

$$\chi(w_i) = \begin{cases} 0, & w_i \geqslant 0, \\ \infty, & w_i < 0, \end{cases} \tag{6.41}$$

for $i = 1, 2, \cdots, N_v \times N_m$. For the regularization, we want to penalize the total sum of weights. So we add ℓ_1 regularization and use another variable, $\bar{\boldsymbol{z}}$, to facilitate the process.

The integrated optimization is expressed as

$$\min_{\bar{\boldsymbol{x}},\ \bar{\boldsymbol{w}},\ \bar{\boldsymbol{z}}} \quad \frac{1}{2}\|(\boldsymbol{I}\otimes\boldsymbol{C})\,\bar{\boldsymbol{x}}-\hat{\boldsymbol{b}}\|^2_{\boldsymbol{\Sigma}^{-1}_{\hat{\boldsymbol{b}}}}+\frac{1}{2}\,\|(\boldsymbol{A}\otimes\boldsymbol{I})\,\bar{\boldsymbol{z}}-\bar{\boldsymbol{x}}\|^2_{\boldsymbol{\Sigma}^{-1}_{\bar{\boldsymbol{x}}_{\mathrm{ml}}}}+\frac{1}{2}\|\bar{\boldsymbol{z}}-\bar{\boldsymbol{w}}\|_2^2+\beta\|\bar{\boldsymbol{w}}\|_1+\chi(\bar{\boldsymbol{w}}). \quad (6.42)$$

As we can see, the first two terms in problem (6.42) are weighted least squares. Furthermore, the last three terms can be regarded as a soft shrinkage function with a mutation. Given $\bar{\boldsymbol{z}}$, the optimization problem can be described as

$$\min_{\bar{\boldsymbol{w}}} \frac{1}{2}\|\bar{\boldsymbol{w}}-\bar{\boldsymbol{z}}\|_2^2+\beta\|\bar{\boldsymbol{w}}\|_1+\chi(\bar{\boldsymbol{w}}), \quad (6.43)$$

Let the i-th element in $\bar{\boldsymbol{w}}$ be $\bar{w}_i$. If $\bar{w}_i \leqslant 0$, we have $\bar{w}_i = 0$. If $\bar{w}_i > 0$, we differentiate the equation with respect to $\bar{w}_i$ and let the derivative equal 0 for $i = 1, 2, \cdots, N_v \times N_m$. Then we can obtain that

$$\frac{1}{2}\,(2\bar{w}_i - 2\bar{z}_i)+\beta = 0. \quad (6.44)$$

So we have

$$\bar{w}_i = \bar{z}_i - \beta. \quad (6.45)$$

With the bound constraint, each $\bar{w}_i$ should satisfy the condition

$$\bar{w}_i = \max\left(\bar{z}_i-\beta,\ 0\right). \quad (6.46)$$

This function is only defined on the half domain of a regular soft shrinkage function. The plot of this function is shown in Figure 6.1. Therefore, problem (6.43) has an analytical solution

$$\bar{\boldsymbol{w}} = \max\left(\bar{\boldsymbol{z}}-\beta,\ \boldsymbol{0}\right). \quad (6.47)$$

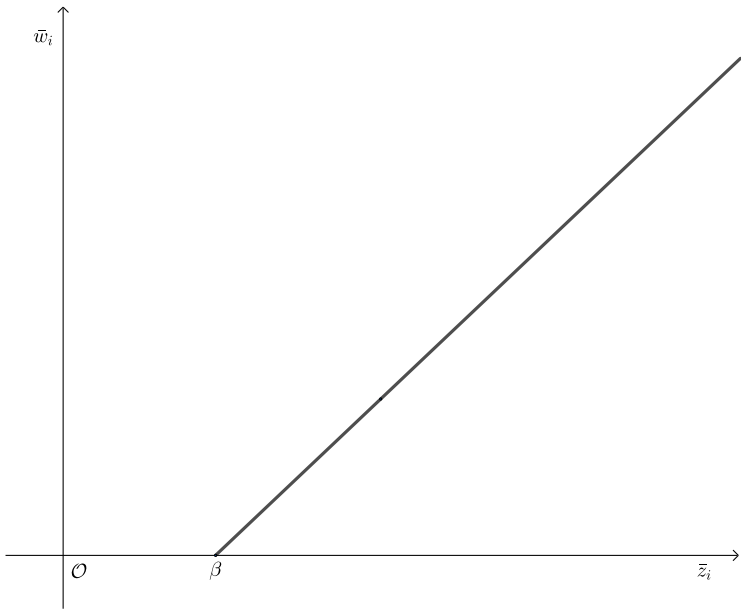

Figure 6.1: The soft shrinkage function with a bound constraint.

The updating framework to solve problem (6.42) is presented as

$$\begin{aligned}
\bar{\boldsymbol{x}}_{k+1} &= \bar{\boldsymbol{x}}_k - \alpha_1 \left[(\boldsymbol{I} \otimes \boldsymbol{C}^T) \boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}}^{-1} (\boldsymbol{I} \otimes \boldsymbol{C}) \bar{\boldsymbol{x}}_{\boldsymbol{k}} - (\boldsymbol{I} \otimes \boldsymbol{C}^T) \boldsymbol{\Sigma}_{\hat{\boldsymbol{b}}}^{-1} \hat{\boldsymbol{b}} \right], \\
\bar{\boldsymbol{z}}_{k+1} &= \bar{\boldsymbol{z}}_k - \alpha_2 \left[(\boldsymbol{A}^T \otimes \boldsymbol{I}) \boldsymbol{\Sigma}_{\hat{\boldsymbol{x}}_{\mathbf{ml}}}^{-1} (\boldsymbol{A} \otimes \boldsymbol{I}) \bar{\boldsymbol{z}}_{\boldsymbol{k}} - (\boldsymbol{A}^T \otimes \boldsymbol{I}) \boldsymbol{\Sigma}_{\hat{\boldsymbol{x}}_{\mathbf{ml}}}^{-1} \bar{\boldsymbol{x}}_{k+1} \right], \\
\bar{\boldsymbol{w}}_{k+1} &= \max \left(\bar{\boldsymbol{z}}_{k+1} - \beta,\ \mathbf{0} \right),
\end{aligned} \tag{6.48}$$

where α_1 and α_2 are selected by line search. This iterative framework is similar to the coordinate descent method, while the result in the first step is reused in the second step and the result obtained from the second step is indispensable for the third step. This framework is simple and easy to implement and it is also clear that we can use the properties of the Kronecker product to avoid saving large matrices.

To conclude, this coupled method might generate a better solution to the optimization problem (6.2). It has the similarity to the coordinate descent method, in which we only need to consider one direction in each update of variables. Furthermore, this method is easy to implement because the iteration in each direction is straightforward. However, we should consider the quality of reconstructed images as well. Similar to the semi-convergence properties [16], the solution obtained with extensive iteration might give us unsatisfactory

results. In this case, we should conduct numerical experiments to compare this method with the two-step method proposed in the previous section.

6.3 Numerical Experiments

For the numerical experiments, we use the same test problem as Chapter 5. The object is composed of two materials, plexiglass and polyvinyl chloride (PVC). The first material map is made up of a circular mask in the center of the image while the second material map includes only small "spikes". In real-life applicaitons, the first material map can be represented by main tissues and these small spikes can indicate calcium in bones. The original images are shown in Figure 6.2. In addition, we use the same parameters as

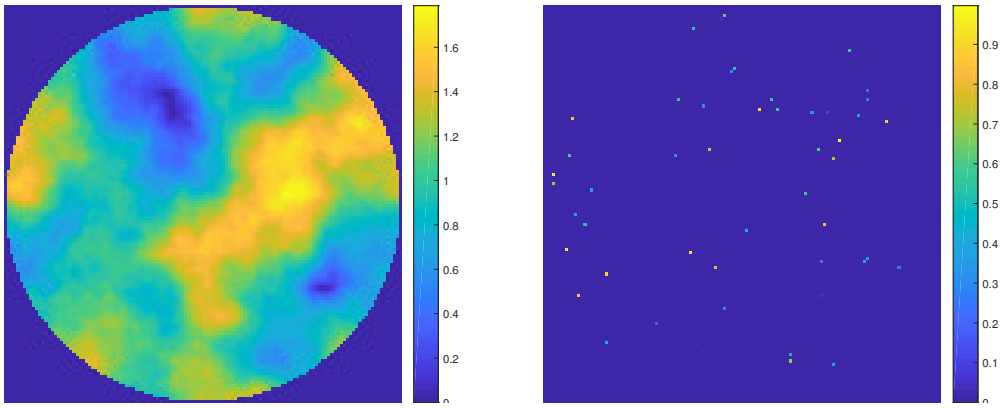

Figure 6.2: The original material maps for plexiglass (Left) and PVC (Right).

Table 4.1 to generate the matrix $\boldsymbol{A}$ and we take 180 projections that are equally distributed from 0 to 360 degrees. The spectrum is built with 120 kV voltage and and the detector is assumed to be photon-counting with five energy windows. These energy windows can detect photon energies of 10 to 34 KeV, 35 to 49 KeV, 50 to 64 KeV, 65 to 79 KeV and 80 to 120 KeV, respectively.

The reconstructed images obtained using the two-step method are shown in Figure 6.3. As we can identify, the images are of high quality in general. The distributions of weights are located in the same positions as the origins. For the first figure, the main profile is similar to the origin but the areas of yellow colors are shrank while the areas of blue color

are expanded. For the second figure, it still has shades but dots are located in the same places as the origin. Compared with the reconstructed images in Chapter 5, the first

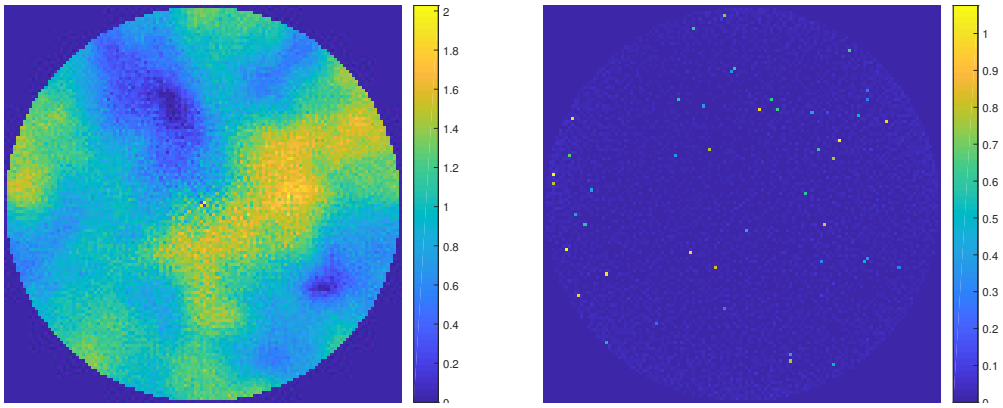

Figure 6.3: The reconstructed material maps for plexiglass (Left) and PVC (Right) using the two-step method.

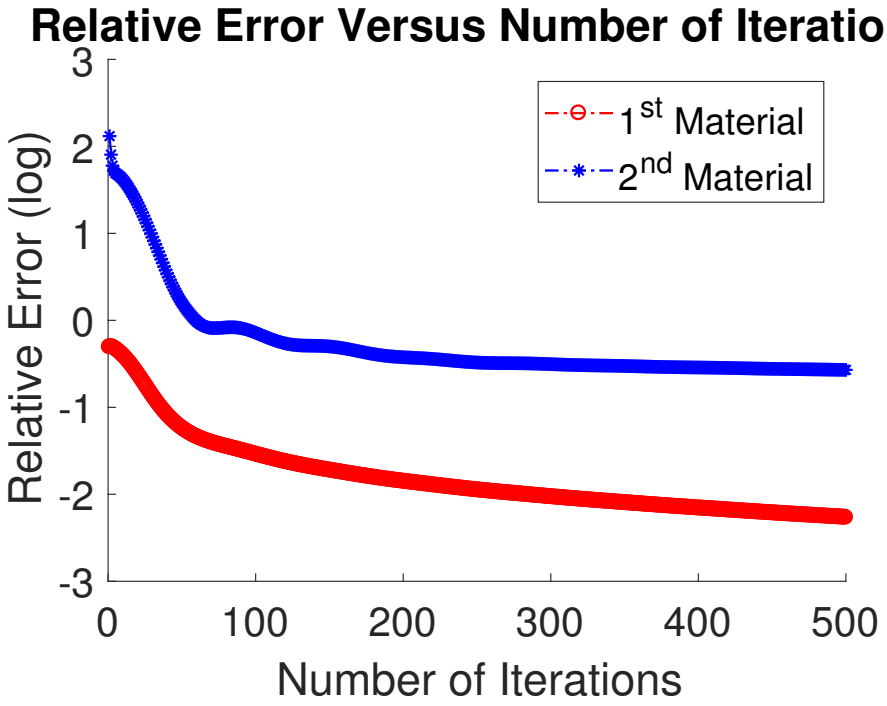

Figure 6.4: The relative errors of two material maps solved by the two-step method.

image has artifacts that are located in the center of the images. Moreover, the edges in the first image are not as smooth as the reconstructed image in Chapter 5. This might result from the ℓ_1 regularization used in this method or early termination of iterations. The

second images displayed in this chapter and Chapter 5 present similar shades and thus the reconstruction qualities are close.

We can also present the convergence properties by showing the plot of relative errors. The relative error figure is shown in Figure 6.4. From Figure 6.4, we can see that the relative error corresponding to the second material drops fast in the beginning but eventually stagnates. The relative error corresponding to the first material drops in a similar way but it arrives to a lower level. It also shows that the first material has a better convergence property than the second material. Moreover, these two curves both indicate convergences. The red curve converges to a lower relative error and the blue curve converges to a higher relative error. The slopes of these two curves in the last several iterations confirm the convergence phenomenon.

To compare with the two-step method, we also show the results obtained using the coupled method. The reconstructed images are shown in Figure 6.5. Compared with Fig-

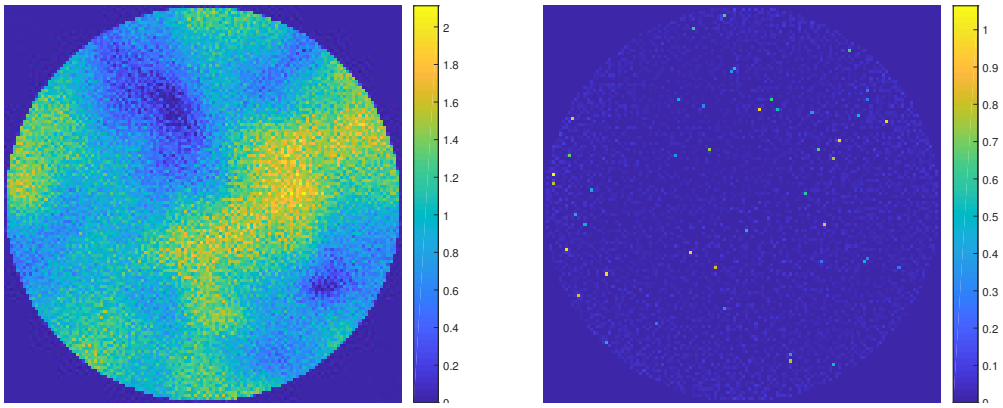

Figure 6.5: The reconstructed material maps for Plexiglass (Left) and PVC (Right) using the coupled method.

ure 6.3, these two images are more blurred and have more shades. For the first image, there are several blue dots scattering around the yellow areas and the yellow areas in the upper right corner and lower bottoms are not clear to identify. For the second material map, the locations of spikes are clear but the shades are more obvious compared with the reconstructed image obtained from the two-step method.

The relative errors corresponding to the iterative process of the coupled method is shown in Figure 6.6. From the slops of curves, we can see that this method converges slower than

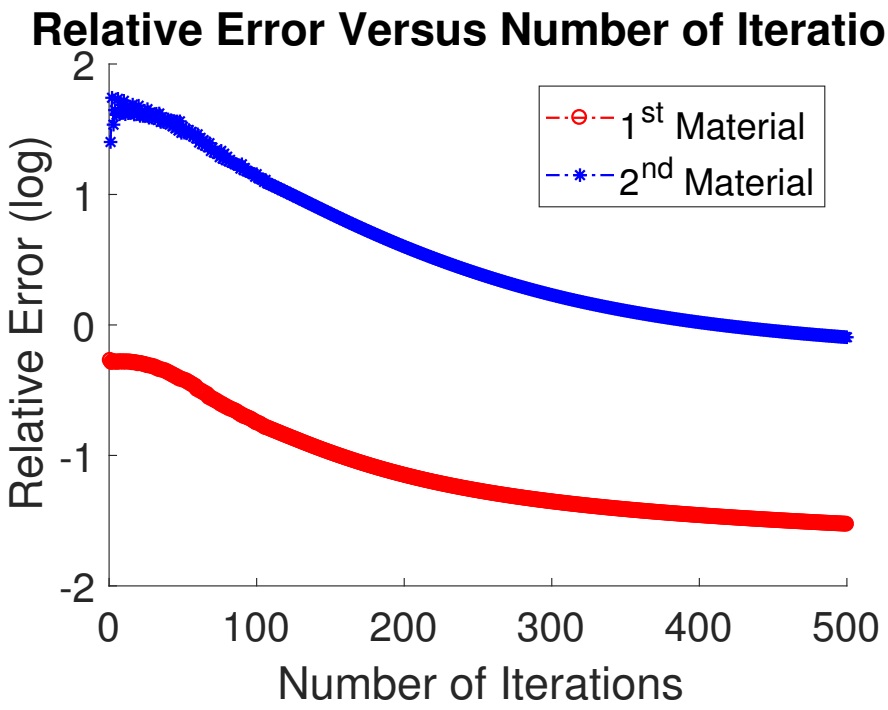

Figure 6.6: The relative errors of two material maps solved by the coupled method.

the previous two-step method and both materials stagnate at higher levels of relative errors. This phenomenon coincides with the observations we have found from the reconstructed images. This coupled method might generate a better solution to the optimization problem (6.2), but it might be a worse solution in terms of image quality. On the other hand, we can update the step alternatingly with this coupled method. In contrast, the updating framework of the two-step method does not have this property.

6.4 Conclusions and Remarks

In this chapter, we propose a two-step method that can solve the spectral CT model in sequence. In the first step, a row-wise model that is based on a weighted least squares term is used as a bridge to the second step. Since each row does not depend on each other, each tiny problem can also be solved in parallel. After we have obtained the results from the first step, we need to quanitfy the noise propagation and set up another least squares problem under bound constraints to compute the final solution. We implement FISTA with

projections to solve this problem to a high accuracy. Instead of using a two-step method, we can build an optimization problem that consists of these two terms and solve it directly. This coupled method is likely to provide us a better solution to the optimization problem. However, based on the numerical results, the two-step method beats the coupled method in terms of image quality, convergence speed and relative errors.

Compared with the method proposed in Chapter 5, this two-step method is more efficient when the product of three parameters, number of projections, range of angles and number of energy bins, is limited. However, for high-resolution images or 3D reconstruction, extensive projections are necessary to guarantee the image quality. Even if we can use parallel processing to reduce this influence, we should also choose different methods under proper circumstances.

Chapter 7

Conclusions and Future Works

Spectral computed tomography problems involve nonlinearity and require more efforts to obtain quantitative information. On the other hand, it can also offer material composition as well as images of higher quality with alleviation of artifacts. Based on different assumptions, spectral computed tomography problems display distinct forms and various math tools are used to compute solutions. For the simple energy discriminating model, we build a nonlinear optimization problem based on a Poisson likelihood estimator, and with this model, a nonlinear interior-point trust region method is introduced to obtain robust reconstruction. For the energy-windowed spectral CT model, we contribute to the optimization frameworks and preconditioners. First, we build a nonlinear least squares problem and solve it with a two-step method, projected line search plus the trust region method. We also propose an adaptive preconditioner to further mitigate the influence of small singular values. With the Gaussian assumption of noise, we transform the energy-windowed spectral CT model into a weighted least squares problem under bound constraints. An efficient preconditioner derived from the corresponding Hessian is presented to reduce the ill-posedness significantly. This preconditioner is inspired from the interlacing of the Kronecker products and diagonal matrices and is built by using a rank-1 approximation. Moreover, we introduce a subproblem to simplify the projection step that is used to keep nonnegativity. Even if we can calculate the solution of the spectral CT model using optimization and precondtioners, we can also consider further separation of the model. In this case, we propose a two-step method using

an auxiliary variable. The first step can be further decomposed into row slices, which can be solved in parallel. After quantifying the noise propagation, we solve the problem corresponding the second step with FISTA.

Even if we only focus on computing the solutions to spectral CT problems, other research areas deserve considerations. For example, the model we use is only based on a single energy source. We can also consider dual-source, or even three or more sources, CT models. The problem of reconstructing images from limited data is even more challenging, and it is important to think about how to find a robust reconstruction under this situation. Furthermore, we can try to combine the state-of-the-art machine learning techniques with spectral CT. Machine learning can be used to conduct data-preprocessing, predict similar features and validate the strength of proposed methods. To conlcude, spectral CT is still an active and promising research area and it can be expected that new inspiring work will continue to be done for years to come.

Bibliography

[1] R. E. Alvarez and A. Macovski, *Energy-selective reconstructions in x-ray computerised tomography*, Physics in Medicine and Biology, 21 (1976), p. 733.

[2] 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.

[3] 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.

[4] E. C. Beckmann, *CT scanning the early days*, The British Journal of Radiology, (2014).

[5] P. Bouguer, *Essai d'optique sur la gradation de la lumière*, chez Claude Jombert, ruë S. Jacques, au coin de la ruë des Mathurins, à l , 1729.

[6] V. M. Bustamante, *Iterative Polyenergetic Digital Tomosynthesis Reconstructions for Breast Cancer Screening*, PhD thesis, Emory University, 2013. https://etd.library.emory.edu/file/view/pid/emory:d6wn4/mejia%20bustamante_dissertation.pdf.

[7] V. M. Bustamante, J. G. Nagy, S. S. Feng, and I. Sechopoulos, *Iterative breast tomosynthesis image reconstruction*, SIAM Journal on Scientific Computing, 35 (2013), pp. S192–S208.

[8] R. Byrd, J. Nocedal, and R. Waltz, *KNITRO: An integrated package for nonlinear optimization*, Large-scale Nonlinear Optimization, (2006), pp. 35–59.

[9] R. H. Byrd, M. E. Hribar, and J. Nocedal, *An interior point algorithm for large-scale nonlinear programming*, SIAM Journal on Optimization, 9 (1999), pp. 877–900.

[10] 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.

[11] J. Chung, J. G. Nagy, and I. Sechopoulos, *Numerical algorithms for polyenergetic digital breast tomosynthesis reconstruction*, SIAM Journal on Imaging Sciences, 3 (2010), pp. 133–152.

[12] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, vol. 18, Prentice-Hall Englewood Cliffs, NJ, 1983.

[13] I. Daubechies, M. Defrise, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, 57 (2004), pp. 1413–1457.

[14] S. R. Deans, *The Radon Transform and Some of Its Applications*, Courier Corporation, MA, 2007.

[15] I. A. Elbakri and J. A. Fessler, *Statistical image reconstruction for polyenergetic x-ray computed tomography*, IEEE Transactions on Medical Imaging, 21 (2002), pp. 89–99.

[16] T. Elfving, P. C. Hansen, and T. Nikazad, *Semi-convergence properties of kaczmarzs method*, Inverse Problems, 30 (2014), p. 055007.

[17] H. Engl, *Discrepancy principles for tikhonov regularization of ill-posed problems leading to optimal convergence rates*, Journal of Optimization Theory and Applications, 52 (1987), pp. 209–215.

[18] C. L. Epstein, *Introduction to the Mathematics of Medical Imaging*, SIAM, PA, 2007.

[19] T. G. Flohr, C. H. McCollough, H. Bruder, M. Petersilka, K. Gruber, C. Süβ, M. Grasruck, K. Stierstorfer, B. Krauss, R. Raupach, et al., *First performance evaluation of a dual-source CT (DSCT) system*, European Radiology, 16 (2006), pp. 256–268.

[20] J. Frikel and E. T. Quinto, *Characterization and reduction of artifacts in limited angle tomography*, Inverse Problems, 29 (2013), p. 125007.

[21] D. Gabay and B. Mercier, *A Dual Algorithm for the Solution of Nonlinear Variational Problems via Finite Element Approximation*, Institut de recherche d'informatique et d'automatique, 1975.

[22] S. Gazzola, P. C. Hansen, and J. G. Nagy, *IR Tools: a matlab package of iterative regularization methods and large-scale test problems*, Numerical Algorithms, (2018), pp. 1–39.

[23] G. H. Golub, M. Heath, and G. Wahba, *Generalized cross-validation as a method for choosing a good ridge parameter*, Technometrics, 21 (1979), pp. 215–223.

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

[25] R. Gordon, R. Bender, and G. T. Herman, *Algebraic reconstruction techniques (art) for three-dimensional electron microscopy and x-ray photography*, Journal of Theoretical Biology, 29 (1970), pp. 471–481.

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

[27] 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.

[28] J. Hadamard, *Sur les problèmes aux dérivées partielles et leur signification physique*, Princeton University Bulletin, (1902), pp. 49–52.

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

[30] P. C. Hansen, J. G. Nagy, and D. P. O'leary, *Deblurring Images: Matrices, Spectra, and Filtering*, vol. 3, SIAM, PA, 2006.

[31] P. C. Hansen and D. P. OLeary, *The use of the l-curve in the regularization of discrete ill-posed problems*, SIAM Journal on Scientific Computing, 14 (1993), pp. 1487–1503.

[32] 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.

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

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

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

[36] S. Karczmarz, *Angenaherte auflosung von systemen linearer glei-chungen*, Bulletin International de l'Acadmie Polonaise des Sciences et des Lettres, (1937), pp. 355–357.

[37] C. T. Kelley, *Iterative Methods for Optimization*, vol. 18, SIAM, PA, 1999.

[38] J. H. Lambert, *Photometria sive de mensura et gradibus luminis, colorum et umbrae*, Klett, 1760.

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

[40] R. Mathias, *The spectral norm of a nonnegative matrix*, Linear Algebra and its Applications, 139 (1990), pp. 269–284.

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

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

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

[44] 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.

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

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

[47] 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.

[48] J. Nocedal and S. Wright, *Numerical Optimization*, Springer Science & Business Media, NY, 2006.

[49] X. Pan, E. Y. Sidky, and M. Vannier, *Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction?*, Inverse problems, 25 (2009), p. 123009.

[50] J. Radon, *1.1 über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten*, Classic Papers in Modern Diagnostic Radiology, 5 (2005), p. 21.

[51] Y. Saad and M. H. Schultz, *Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856–869.

[52] J. Schlomka, E. Roessl, R. Dorscheid, S. Dill, G. Martens, T. Istel, C. Bäumer, C. Herrmann, R. Steadman, G. Zeitler, et al., *Experimental feasibility of multi-energy photon-counting k-edge imaging in pre-clinical computed tomography*, Physics in Medicine & Biology, 53 (2008), p. 4031.

[53] 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.

[54] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM Journal on Numerical Analysis, 20 (1983), pp. 626–637.

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

[56] P. Wolfe, *Convergence conditions for ascent methods*, SIAM Review, 11 (1969), pp. 226–235.

[57] 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.