

Tomographic Reconstruction

Tiantian Song

May 2024

Contents

| | |
|--------------------------------------------------------------------------------|-----------|
| Abstract | 1 |
| Acknowledgement | 2 |
| 1 Introduction | 3 |
| 1.1 A Brief History of Tomography | 3 |
| 1.2 Principle of Tomography | 4 |
| 2 Theories and Methods | 6 |
| 2.1 Lambert-Beer Law | 6 |
| 2.2 Radon Transform | 7 |
| 2.2.1 Sinogram | 8 |
| 2.2.2 Properties | 9 |
| 2.3 Noise | 10 |
| 2.4 Reconstruction Methods | 11 |
| 2.4.1 Filtered Back-Projection | 11 |
| 2.4.2 Bayesian Inversion | 15 |
| 2.4.3 Maximum Likelihood Reconstruction for Emission Tomog- raphy | 17 |
| 2.4.4 Bayesian Reconstruction Based on MLEM Algorithm | 21 |
| 2.5 Regularisation Techniques for Tomography Problems | 23 |
| 2.5.1 Total Variation Regularisation | 23 |
| 2.5.2 Primal-Dual Hybrid Gradient Algorithm | 24 |
| 3 Results and Discussion | 26 |
| 3.1 Overview of Core Imaging Library | 26 |
| 3.1.1 CIL Framework | 26 |
| 3.1.2 Data Containers | 27 |
| 3.1.3 Optimisation Framework | 27 |
| 3.1.4 Functions | 27 |
| 3.1.5 Block Framework | 28 |
| 3.2 Example Studies | 28 |
| 3.2.1 Reconstruction Using FBP and PDHG algorithm | 28 |
| 3.2.2 MLEM Reconstruction Method | 29 |
| 3.2.3 Bayesian Reconstruction Based on MLEM algorithm | 31 |

| | | |
|----------|-----------------------------------------------------------|-----------|
| 4 | Appendix: Computer Code | 33 |
| 4.1 | FBP and PDHG | 33 |
| 4.2 | MLEM algorithm | 34 |
| 4.3 | Bayesian Reconstruction Based on MLEM algorithm | 36 |

Abstract

Tomographic reconstruction is a process used in medical imaging and other fields to produce detailed images of an object's internal structures. This sophisticated technique involves acquiring multiple 2D projections or other imaging data from varied angles around the object and using mathematical algorithms to reconstruct a 3D image. Some examples of tomographic reconstruction types include X-ray computed tomography (CT), positron emission tomography (PET) and transmission electron microscope (TEM) tomography. Tomography reconstruction plays a crucial role in obtaining detailed images for accurate diagnostic purposes. This dissertation provides a review of the history of tomography, discusses fundamental mathematical theories and methods for image reconstruction, and explores regularisation techniques. Lastly, it applies these methods using the Core Imaging Library (CIL) [33] [24].

Keywords: Tomographic Reconstruction, Radon Transform, Filtered Back-Projection, Bayesian Inversion, Maximum Likelihood Reconstruction, Regularisation

Acknowledgement

I would like to thank my supervisor, Sean Holman, for his invaluable guidance, support and patience throughout the entire process of conducting this project. I am thankful for his inspiration in my interests in applied mathematics, and for his assistance in enhancing my academic writing and computer skills. Conducting this project with him has reinforced my passion for research and motivated me to pursue further studies.

I am also thankful to my father's master's and Ph.D. students, Yixiang Ren, Yixuan Jin, Chenyang Zhang and Chengjie Dai, who took the time to teach me the basics of programming, which played an important role in improving the quality of my work.

Lastly, I am grateful to my family for their support, understanding and encouragement in my research journey. Their love, patience and belief in me have been a constant source of strength and motivation.

Chapter 1

Introduction

The dissertation is organised as follows. Chapter 1 provides a review of the history of tomography and the general principle of tomography. In Chapter 2, we give a review of fundamental theories and methods for reconstructing objects from projection data, along with examples of new iterative methods for reconstruction and regularisation techniques. In Chapter 3, we apply some of the methods discussed in Chapter 2 using the Core Imaging Library (CIL), discuss the results and give conclusions.

1.1 A Brief History of Tomography

The term “tomography”, which comes from the Greek word *tomos* (meaning “slice”) and *graphos* (meaning “description”), specifically refers to imaging by acquiring sections or slices of an object [17, Chap. 1]. Tomography is a powerful technique enabling the visualisation of the internal structure of objects or subjects of interest. This is achieved by measuring the transmission of waves that pass through objects, whose intensity is attenuated by the material through which they pass.

The commencement of computed tomography traces back to 1917, when Johann Radon introduced the mathematical framework of the Radon transform [18]. Radon established the foundational principles for the reconstruction of images from their projections after conducting a task of reconstructing a function on the plane from its line integrals.

Progressing into the 1960s, Allan Cormack (Figure 1.1a) and Godfrey Hounsfield (Figure 1.1b) independently promoted the development of the experimental measurements that were crucial for tomographic reconstruction. In 1963, Cormack [6] published a seminal paper outlining the mathematical principles that formed the basis for reconstructing images from X-ray projections. In 1967, Hounsfield built the first emission CT scanner, known as the EMI Mark I at EMI (Electric and Musical Industries), the company where he was employed, which marked the practical realisation of CT imaging. Except for Allan Cormack’s and Godfrey Hounsfield’s contributions to tomography, Kuhl and Roy Edwards, Chiefs of Engineering at the Hospital of the University of Pennsylvania, undertook the construction of two emission CT scanners: the Mark II, finalised in 1964, and the Mark III, completed in 1970 [11].

The development of CT scanners began from when Godfrey Hounsfield invented a clinical CT scanner in 1971 [34], for which a patent was granted in 1972. This remarkable achievement led to Hounsfield and Cormack [7] jointly receiving the Nobel Prize for Physiology or Medicine, acknowledging their groundbreaking contributions to the development of analytical techniques for image reconstruction. The 1970s also witnessed the emergence of magnetic resonance imaging (MRI), a medical imaging

technique that uses magnetic fields to produce images [40]. Concurrently, the development of positron emission tomography (PET) introduced a functional imaging modality that used emission CT together with radioisotope tracers, offering insights into metabolic processes within the human body. At the end of 1974, Mike Phelps, Ed Hoffman and colleagues at Washington University constructed PET III for human studies, which marked the beginning of human PET tomography [31].

The image quality and diagnostic abilities are strongly affected by the scan speed of the CT scanner. The 1980s witnessed a notable advancement in CT scan accuracy and efficiency with the use of electrical energy in the scanner's X-ray tube, which prevented fast and continuous rotation and therefore prevented the losses of image sharpness [26]. Additionally, iterative reconstruction algorithms (e.g.[9]) were introduced during this time. The 1990s saw the emergence of multi-detector CT scanners. The multi-detector row computed tomography (CT) by the major CT vendors, introduced in 1998, enabled faster scan speed, improved spatial resolution and better use of the available X-ray power [10].



(a) Allan Cormack

(b) Godfrey Hounsfield

Figure 1.1: Allan Cormack and Godfrey Hounsfield. These figures are taken from the sources [1] and [22].

Continuing into the 2000s, ongoing developments include advancements in modalities like cone-beam CT [37], dual-energy CT [23], and ultra-high field MRI [28]. These innovations aim to improve image quality, minimise radiation exposure, and expand the range of applications.

1.2 Principle of Tomography

Firstly, detailed images of an object or subject's internal structures are created by collecting and processing data from multiple directions [19, Chap. 3]. Data acquisition can use diverse energy sources, such as X-rays [41], gamma rays [21] and radio waves [35]. Then, the acquired data undergo mathematical transformations. Subsequently, reconstruction algorithms are applied to the transformed data to reconstruct the 2D or 3D image.

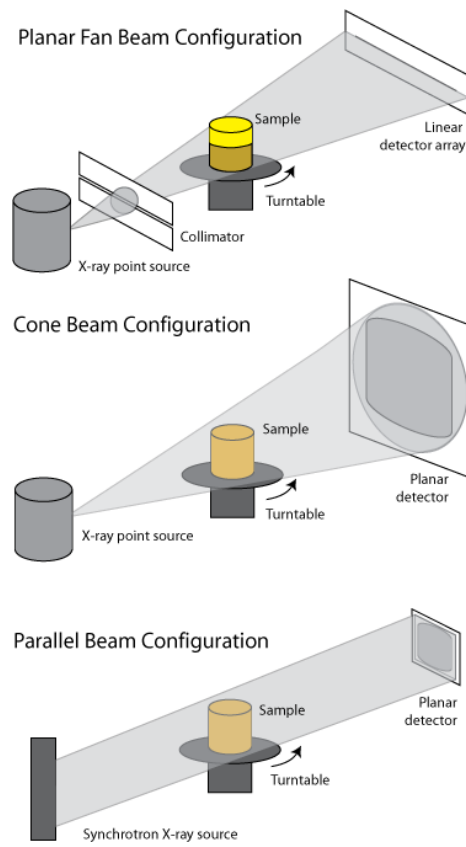


Figure 1.2: X-ray Computed Tomography. This figure is taken from the source [27].

Chapter 2

Theories and Methods

In this chapter, we give detailed descriptions of fundamental theories and methods for image reconstruction. This chapter is organised as follows: in Section 2.1 and 2.2, we provide the mathematical formula for the Lambert-Beer law and a conceptual definition of the Radon transform, and discuss the relationship between them. In Section 2.3, we describe how the noise arises, provide examples of different types of noise and explain how noise affects the data obtained from the Radon transform.

2.1 Lambert-Beer Law

Consider a conventional Cartesian coordinate system (x_1, x_2) . Given an angle θ of the line with respect to the x-axis, define an orthogonal set of basis vectors

$$\theta = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \quad \text{and} \quad \theta^\perp = \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \end{pmatrix}.$$

Consider an axis parameterised by coordinates s rotated by the angle θ with respect to the x_1 -axis, which has a unit direction vector θ . Denote $L_{\theta,s}$ as the line orthogonal to θ with a unit direction vector θ and signed orthogonal distances s to the origin. The expression for this line is as follows

$$L_{\theta,s} = \{x | x \cdot \theta = s\}, \quad (2.1)$$

where

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

and $x \cdot \theta = x_1 \cos(\theta) + x_2 \sin(\theta)$.

Assume that an object is characterised by the image function $f(x_1, x_2)$. The *projection* that expresses line integrals along $L_{\theta,s}$ at fixed angle θ is defined as

$$g(\theta, s) = \int_{L_{\theta,s}} f(x_1, x_2) dl \text{ for } s \in [-1, 1].$$

The Lambert-Beer law describes how the intensity of an X-ray beam decreases exponentially as it passes through a material. Mathematically, it can be expressed as

$$I_{\theta,s} = I_0 e^{-\int_{L_{\theta,s}} \mu(x_1, x_2) dl}, \quad (2.2)$$

where I_0 is the initial intensity of the X-ray beam, I is the remaining beam intensity, μ is the linear attenuation coefficient and l is the parametrised position of a line along which an X-ray propagates [30].

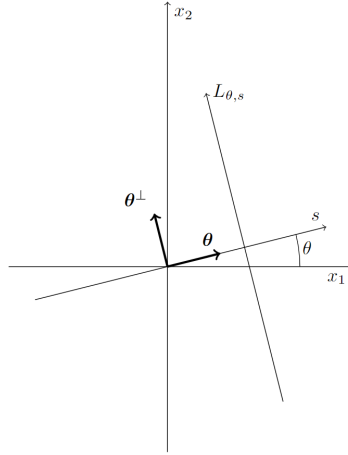


Figure 2.1: The geometry of a line $L_{\theta,s}$. The axis s is rotated by the angle θ with respect to the x_1 -axis, with a unit vector θ . θ^\perp is the unit vector orthogonal to θ . The line $L_{\theta,s}$ is orthogonal to the s -axis. This figure is taken from the source [17, Chap. 5].

2.2 Radon Transform

The Radon transform [17, Chap. 5] is a mathematical operation that represents the integral of a function over straight lines in a particular direction. In X-ray tomography, the Radon transform is used to model the process of collecting data from different angles of an object. The data is then used to reconstruct the image of an object using the techniques such as the filtered back-projection (Section 2.4.1).

Definition 2.2.1 (Radon transform). The *Radon transform* for a function f is defined as the projections at all angles

$$R[f](\theta, s) = g(\theta, s) = \int_{L_{\theta,s}} f(x_1, x_2) dl$$

for $\theta \in [0, 2\pi]$.

In fact, the Radon transform integrates the attenuation values along the paths of X-rays passing through an object, which is governed by the Lambert-Beer law [17, Chap. 5]:

Dividing both sides of Eq. 2.2 by I_0 ,

$$\frac{I_{\theta,s}}{I_0} = e^{-\int_{L_{\theta,s}} \mu(x_1, x_2) dl}. \quad (2.3)$$

Taking the negative logarithm on both sides of Eq. 2.3 yields

$$-\log\left(\frac{I_{\theta,s}}{I_0}\right) = \int_{L_{\theta,s}} \mu(x_1, x_2) dl. \quad (2.4)$$

Eq. 2.4 can be written in Radon transform notation using the identities

$$f(x_1, x_2) = \mu(x_1, x_2) \quad \text{and} \quad R[\mu](\theta, s) = -\log\left(\frac{I_{\theta,s}}{I_0}\right).$$

Figure 2.2 (right) shows an example of the sinogram generated from the Radon transform of the data in Figure 2.2 (left).

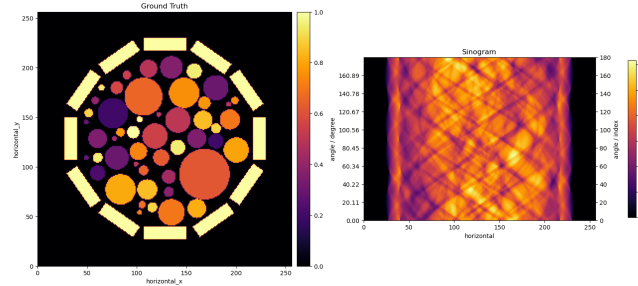


Figure 2.2: The sinogram on the right is obtained from the Radon transform of the data in the left figure. The angle only goes from 0 to π since the second half of the angles contain the same information by symmetry. This figure is taken from the source [24].

2.2.1 Sinogram

The data obtained from the Radon transform is commonly referred to as a sinogram, as sinusoidal waves can often be observed. The example in Figure 2.3 shows disks of different radii, positions and intensities appearing in the sinogram.

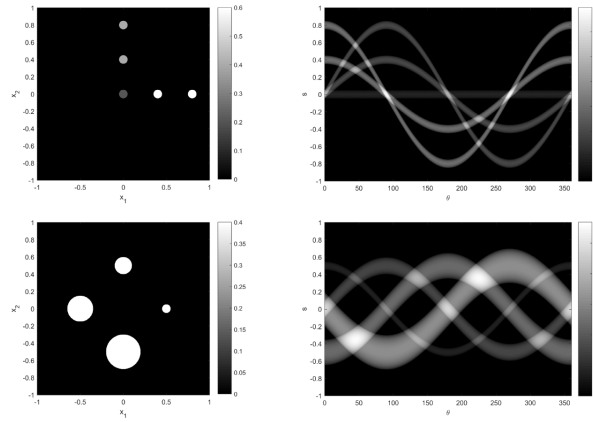


Figure 2.3: In the top left figure, there are five small disks with the same radius but different intensities, distances and angles from the origin. The top right figure displays the sinogram obtained from the Radon transform of the test image in the top left. The sinusoidal waves in the sinogram exhibits different positions, indicating differences in the positions of the disks. The bottom left figure shows four disks with equal intensity and distance from the origin, but with different radii and orientations. The bottom right figure shows the sinogram obtained from the Radon transform of the test image in the bottom left. The sinusoidal waves in the sinogram exhibits varying widths and phases, indicating differences in intensities and positions of the disks. The orientation of this figure is rotated 90 degrees compared to Figure 2.2. This figure is taken from the source [17, Chap. 5].

2.2.2 Properties

The Radon transform has a number of useful properties, such as symmetry, periodicity, linearity, translation, rotation and spatial scaling [17, Chap. 5].

Symmetry and Periodicity

As shown in Section 2.2, the Radon transform of a function $f(x_1, x_2)$ can be written as $g(\theta, s)$. The Radon transform has the symmetry property

$$g(\theta, s) = g(\theta \pm \pi, -s)$$

and has periodicity of 2π , i.e.

$$g(\theta, s) = g(\theta + 2k\pi, s) \quad \text{for } k \text{ integer.}$$

The sinogram in Figure 2.2 exhibits symmetry around the horizontal axis, reflecting the symmetric nature of the Radon transform with respect to rotations by 180 degrees. It also exhibits periodic patterns due to the periodic nature of the Radon transform.

Linearity

Linearity is straightforward. If

$$f(x_1, x_2) = af_1(x_1, x_2) + bf_2(x_1, x_2),$$

then

$$g(\theta, s) = ag_1(\theta, s) + bg_2(\theta, s).$$

Translation

Assume an object is transformed into centre coordinates (x_1^c, x_2^c) , i.e.

$$f_c(x_1, x_2) = f(x_1 - x_1^c, x_2 - x_2^c).$$

Consider a line $L_{\theta, s}$ from Eq. 2.1 that passes through (x_1^c, x_2^c) . The corresponding value of s is

$$s_c(\theta) = x_1^c \cos(\theta) + x_2^c \sin(\theta).$$

s can also be written in another form,

$$s_c(\theta) = r_c \cos(\theta - \phi_c),$$

where (r_c, ϕ_c) are the polar coordinates of the centre, i.e.

$$r_c = \sqrt{(x_1^c)^2 + (x_2^c)^2} \quad \text{and} \quad \phi_c = \arctan\left(\frac{x_2^c}{x_1^c}\right).$$

Therefore, the Radon transform of f_c is

$$g_c(\theta, s) = g(\theta, s - s_c(\theta)).$$

Translation in the sinogram in Figure 2.2 corresponds to shifts along the vertical axis, representing translations of the object in the original image.

Rotation

Assume an object is rotated anticlockwise by an angle of α . Then the polar coordinates of f will be

$$f_\alpha(r, \phi) = f(r, \phi - \alpha).$$

Therefore, the Radon transform of f_α will be

$$g_\alpha(\theta, s) = g(\theta - \alpha, s).$$

Rotation in the sinogram in Figure 2.2 is represented by shifts along the horizontal axis.

Spatial Scaling

Assume an object is scaled from $f(x_1, x_2)$ to

$$f_{ab}(x_1, x_2) = f\left(\frac{x_1}{a}, \frac{x_2}{b}\right),$$

where $a > 0$ and $b > 0$. Then the Radon transform of $f_{ab}(x_1, x_2)$ can be expressed as

$$g_{ab}(\theta, s) = \frac{ab}{\gamma^2(\theta)} g\left(\arctan\left(\frac{b}{a} \tan(\theta)\right), \frac{s}{\gamma(\theta)}\right),$$

where

$$\gamma^2(\theta) = (a \cos(\theta))^2 + (b \sin(\theta))^2.$$

2.3 Noise

In tomography, noise can arise from various sources. Some common sources include photon noise (Poisson noise) due to the statistical fluctuations in the number of photons detected by detectors, detector noise due to imperfections in detectors, and scatter noise caused by X-rays scattering in different directions before reaching the detector. Incorporating noise into a sinogram is a standard procedure in simulating imaging scenarios for diverse experiments and studies. The process involves introducing random variations into the sinogram data and mimicking the impact of noise observed in actual imaging systems [3]. Common methods for adding noise to a sinogram includes Poisson noise, Gaussian noise [29] and Uniform noise [2].

Poisson Noise

Poisson noise, also known as photon noise, is a type of noise that occurs in photon counting and can be modelled by a Poisson process. The transmitted intensity I in the Lambert-Beer law (Section 2.1) can be written as a simple product of the initial intensity I_0 and the survival rate of the intensity p [17, Chap. 4]

$$I = I_0 p \quad \text{with} \quad p = e^{-\int \mu(l) dl}.$$

Each photon is detected with probability p and is independent of other photons. Hence, the number of photons being detected in a certain period of time can be considered a Poisson process, and the expected number of photons being detected should be $I_0 p$. Therefore, I can be considered a random variable following a Poisson distribution with parameter $I_0 p$, i.e. $I \sim \text{Poisson}(I_0 p)$, and the density for I is

$$p(I = k) = e^{-I_0 p} \frac{(I_0 p)^k}{k!}, \quad k = 0, 1, \dots$$

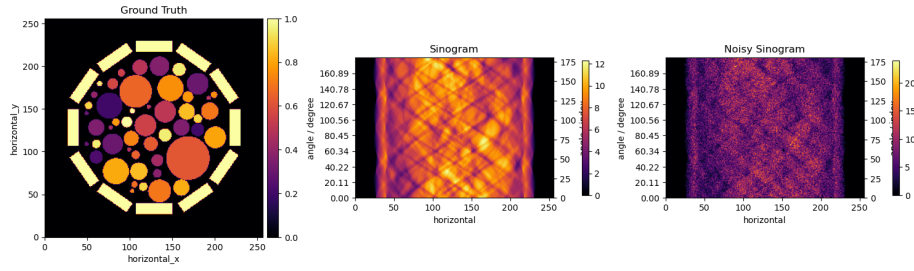


Figure 2.4: The noisy sinogram on the right is obtained by adding Poisson noise to the Radon transform of the data in the left figure. This figure is taken from the source [24].

Poisson-Gaussian approximation

Under the condition that the number of detected photons is large, the Poisson distribution becomes more symmetric and bell-shaped and approaches a Gaussian distribution. The number of detected photons I can be considered as the sum of $I_0 p$ Poisson variables with parameter 1 [17, Chap. 4]. According to the central limit theorem, for large I , I follows a Gaussian distribution with mean and variance both equal to $I_0 p$.

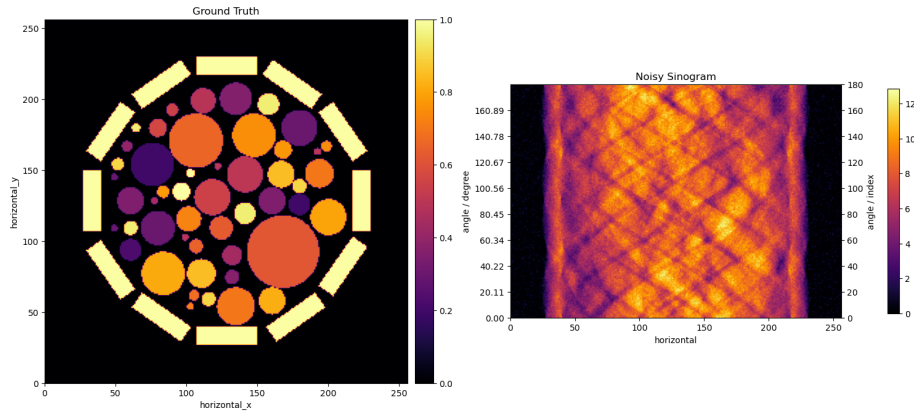


Figure 2.5: The noisy sinogram on the right is obtained by adding Gaussian noise to the Radon transform of the data in the left figure. This figure is taken from the source [24].

2.4 Reconstruction Methods

2.4.1 Filtered Back-Projection

Sections 2.1 and 2.2 demonstrated how an image is mapped to a sinogram through the Radon transform. Intuitively, to reverse this process and reconstruct an image from its sinogram, it is natural to attempt what is known as back-projection [17, Chap. 6].

Definition 2.4.1 (Back-Projection). The back-projection is the integration over the

set of lines which pass through a point. Mathematically, it is an operator $R^\#$ that maps a sinogram to an image through integration over θ

$$R^\#[g](x_1, x_2) = \int_0^{2\pi} g(\theta, x_1 \cos(\theta) + x_2 \sin(\theta)) d\theta.$$

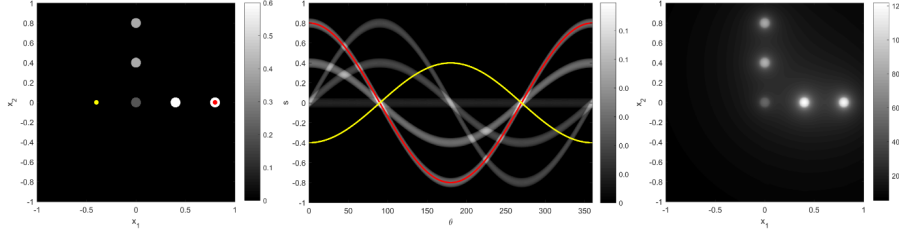


Figure 2.6: Left: Image from Figure 2.3 (top left) with a point marked inside/outside a disk in red/yellow. Middle: The corresponding sinusoidal curves in the sinogram that the back-projection integrates. Right: Back-projection is used to convert the projection data from the sinogram in the middle into a image. It clearly shows the effect of back-projection: it smears out the original image. For an exact recovery of the original image, filtering is also necessary. This figure is taken from the source [17, Chap. 6].

The filtered back-projection involves two main steps: filtering and back-projection. The Fourier transform is used primarily for the filtering step before back-projection. The projection data is transformed into the frequency space using the Fourier transform. After filtering in the frequency space, the filtered data are transformed back to the spatial domain using the inverse Fourier transform.

Fourier transform

Definition 2.4.2 (Fourier transform). The 1D *Fourier transform* [17, Chap. 2] for a function f of the variable x is defined as

$$\hat{f}(\omega) = \sqrt{\frac{1}{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ix\omega} dx.$$

The 2D *Fourier transform* for a function f of the variables x and y is defined as

$$\hat{f}(u, v) = \int \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi(ux+vy)} dx dy.$$

The *inverse Fourier transform* for a function f of the variable ω is defined as

$$\check{f}(x) = \sqrt{\frac{1}{2\pi}} \int_{-\infty}^{\infty} f(\omega) e^{ix\omega} d\omega.$$

Fourier Slice Theorem

Recall from Section 2.1 that $\theta = (\cos(\theta), \sin(\theta))^T$ and $\theta^\perp = (-\sin(\theta), \cos(\theta))^T$.

Theorem 2.4.1 (Fourier Slice Theorem). [17, Chap. 6] Let \hat{f} be the Fourier transform of an image f and $\hat{g}(\theta, r)$ be the Fourier transform of the data $g(\theta, s)$ with regard to s , where $g = R[f]$. The *Fourier Slice Theorem* states that

$$\widehat{R[f]}(\theta, r) = \sqrt{2\pi} \hat{f}(r\theta).$$

Proof. Starting from the definition of 1D Fourier transform

$$\widehat{R[f]}(\theta, \omega) = \sqrt{\frac{1}{2\pi}} \int_{-\infty}^{\infty} g(\theta, s) e^{-is\omega} ds. \quad (2.5)$$

Then apply the definition of Radon transform to Eq. 2.5,

$$\widehat{R[f]}(\theta, \omega) = \sqrt{\frac{1}{2\pi}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(s\boldsymbol{\theta} + l\boldsymbol{\theta}^{\perp}) e^{-is\omega} dl ds. \quad (2.6)$$

Changing the variables from (l, s) to (x_1, x_2) in Eq. 2.6,

$$\widehat{R[f]}(\theta, \omega) = \sqrt{\frac{1}{2\pi}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x) e^{-i\omega\boldsymbol{\theta} \cdot \mathbf{x}} dx_1 dx_2, \quad (2.7)$$

which is exactly the 2D Fourier transform

$$\widehat{R[f]}(\theta, \omega) = \sqrt{2\pi} \hat{f}(\omega\boldsymbol{\theta}). \quad (2.8)$$

□

Theorem 2.4.2 (Filtered Back-Projection). [17, Chap. 6] *The formula for filtered back-projection can be expressed as*

$$\begin{aligned} f(x_1, x_2) &= \frac{1}{4\pi} \int_0^{2\pi} \Lambda[g](\theta, x_1 \cos(\theta) + x_2 \sin(\theta)) d\theta \\ &= \frac{1}{4\pi} R^{\#} \Lambda[g], \end{aligned} \quad (2.9)$$

where $f(x_1, x_2)$ is the reconstructed image, $R^{\#}$ is the back-projection operator in Def 2.4.1, Λ is the filtering operator and $\Lambda[g]$ is the filtered projection data, which can be rewritten as

$$\Lambda[g](\theta, s) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{g}(\theta, r) e^{isr} |r| dr.$$

This formula holds if $g = R[f]$.

Proof. Starting from f satisfying $g = R(f)$,

$$f(x) = F^{-1}F(f)(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{ix\omega} d\omega_1 d\omega_2,$$

where F and F^{-1} are Fourier transform and inverse Fourier transform, respectively. Changing to the polar coordinates (r, θ) with $\omega = r\theta$ yields

$$f(x) = \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\infty} \hat{f}(r\theta) e^{ixr\cdot\theta} r dr d\theta.$$

Applying the Fourier slice theorem,

$$\begin{aligned} f(x) &= \frac{1}{2\pi\sqrt{2\pi}} \int_0^{2\pi} \int_0^{\infty} \hat{g}(\theta, r) e^{ixr\cdot\theta} r dr d\theta \\ &= \frac{1}{4\pi\sqrt{2\pi}} \left(\int_0^{2\pi} \int_0^{\infty} \hat{g}(\theta, r) e^{ixr\cdot\theta} r dr d\theta + \int_0^{2\pi} \int_{-\infty}^0 \hat{g}(\theta + \pi, -r) e^{ixr\cdot\theta} (-r) dr d\theta \right) \\ &= \frac{1}{4\pi\sqrt{2\pi}} \int_0^{2\pi} \int_{-\infty}^{\infty} \hat{g}(\theta, r) e^{ixr\cdot\theta} |r| dr d\theta \\ &= \frac{1}{4\pi} \int_0^{2\pi} \left(\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{g}(\theta, r) e^{ixr\cdot\theta} |r| dr \right) d\theta. \end{aligned} \tag{2.10}$$

The inner integral is exactly the filtered data. Then, the image will be

$$\begin{aligned} f(x_1, x_2) &= \frac{1}{4\pi} \int_0^{2\pi} \Lambda[g](\theta, s) d\theta \\ &= \frac{1}{4\pi} \int_0^{2\pi} \Lambda[g](\theta, x_1 \cos(\theta) + x_2 \sin(\theta)) d\theta \\ &= \frac{1}{4\pi} R^{\#} \Lambda[g], \end{aligned}$$

which is the back-projection of $\Lambda[g]$.

(2.11)

□

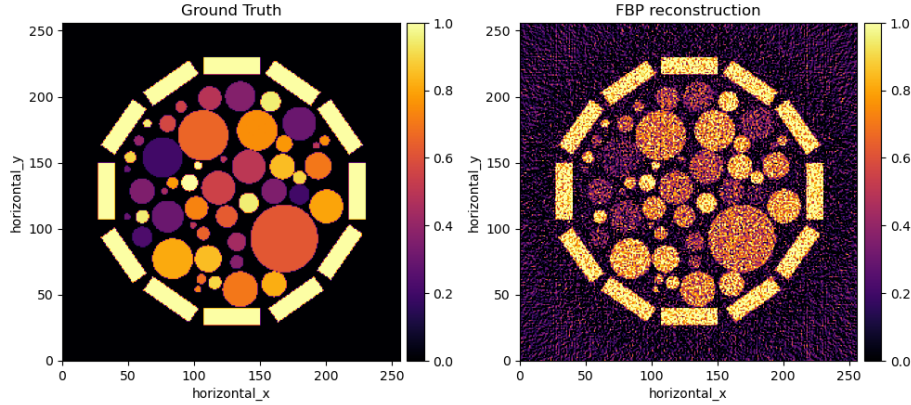


Figure 2.7: Reconstructed image under the filtered Back-Projection method, with added Gaussian noise with mean 0 and variance 0.5. This figure is taken from the source [24].

2.4.2 Bayesian Inversion

The Bayesian inversion methods proposed by [39] incorporate the prior information into the reconstruction process. Whether one can successfully solve inverse problems largely depends on how much one can make use of the prior information about the target. These Bayesian inversion methods address the challenge that arises in some cases, particularly in certain research applications or specialised imaging setups where only a small number of truncated projections from restricted angles are available. Three-dimensional imaging from such data is complex because traditional CT reconstruction techniques, such as filtered back-projection, exhibits poor performance when applied to a small number of irregularly spaced projections, where the angles at which projection data are acquired are not evenly spaced.

The main idea of the statistical inversion problem is to consider the inverse problem as Bayesian inference. Consider linear measurement models with additive Gaussian errors,

$$y = Ax + \epsilon,$$

where $y \in \mathbb{R}^N$, $x \in \mathbb{R}^M$ and $\epsilon \in \mathbb{R}^N$ are vector-valued random variables and A is the deterministic matrix. If ϵ has the Gaussian distribution with mean zero and covariance Γ_{noise} , i.e. $\epsilon \sim N(0, \Gamma_{noise})$, the probability distribution of the noise that can be approximated will be

$$p_{noise}(\epsilon) \sim \exp\left(-\frac{1}{2}\epsilon^T \Gamma_{noise}^{-1} \epsilon\right).$$

Assume that the image x and the noise ϵ are independent. Therefore, their joint probability density function is

$$p(x, \epsilon) = p_{pr}(x)p_{noise}(\epsilon),$$

where $p_{pr}(x)$ is the prior density of the image. The conditional probability of data y given x and ϵ can be written as

$$p(y|x, \epsilon) = \delta(y - Ax - \epsilon),$$

where δ is the Dirac delta function, and its integral over the set \mathbb{R}^N is one. Then, the joint probability distribution function of y and x can be written as

$$p(y, x) = \int_{\mathbb{R}^N} p(y|x, \epsilon) p(x, \epsilon) d\epsilon = p_{pr}(x) p_{noise}(y - Ax). \quad (2.12)$$

Then, using the Bayes theorem, the posterior probability of x , i.e. the conditional probability distribution function of x given y is

$$p(x|y) = \frac{p(y, x)}{p(y)} = \frac{p_{pr}(x) p(y|x)}{p(y)}.$$

From Eq 2.12, the likelihood density $p(y|x)$ is given by

$$p(y|x) = p_{noise}(y - Ax).$$

Therefore, the posterior density is proportional to the product of $p_{pr}(x)$ and $p(y|x)$,

$$p(x|y) \propto p_{pr}(x) p_{noise}(y - Ax).$$

To produce images in the Bayesian approach, there are basically two options. The first tool is the *maximum a posterior* estimate (MAP) in which the image is the maximiser of the posterior density. This is equivalent to regularisation by the prior density and the mathematical problem is optimisation. It is defined as

$$x_{MAP} = \arg \max p(x|y).$$

The second tool is the *conditional mean* estimate (CM) in which a large number of samples are taken from the posterior distribution and then the mean is calculated. It is defined as

$$x_{CM} = \int_{\mathbb{R}^M} x p(x|y) dx.$$

This method is not discussed any further in this dissertation as it's not really currently feasible for imaging applications (it takes a very long computation time).

The most challenging task is to construct the prior. Here are some examples for constructing prior models.

Generic Gaussian Priors

Gaussian white noise and smoothness priors are the most widely used prior models. The Gaussian density functions are in the form of

$$p_{pr}(x) \propto \exp \left(-\frac{1}{2} (x - \bar{x})^T \Gamma^{-1} (x - \bar{x}) \right),$$

where \bar{x} is the mean and Γ is the covariance. The mean of a generic Gaussian prior is typically determined based on the prior knowledge or the estimations of the underlying distribution of the parameter being estimated. The Gaussian white noise is the most common choice for the Tikhonov regularisation problem (Section 2.5.1). It has the following form,

$$p_{pr}(x) \propto \exp \left(-\frac{1}{2\sigma^2} \|x - \bar{x}\|_2^2 \right),$$

where it assumes that the covariance matrix is diagonal. An example of the smoothness prior for a 2D pixel image in a $P \times P = M$ square mesh is

$$p_{pr}(x) \propto \exp \left(-\alpha \sum_{k \in M} \sum_{l \in N_k} |x_k - x_l|^2 \right),$$

where α is the scaling parameter, M is the set of indices of non-boundary pixels, and N_k is the set of index of four nearest pixels for the non-boundary pixel k .

Impulse noise priors

Sometimes, as seen in astronomy for instance, low-contrast images are expected, with only a few outstanding pixels as outliers. In such cases, impulse noise priors would be used. Here are three commonly used examples for impulse noise priors. They are the L^1 prior

$$p_{pr}(x) \propto \exp \left(-\alpha \sum_{k=1}^M |x_k| = \exp(-\alpha \|x\|_1) \right),$$

the maximum entropy prior

$$p_{pr}(x) \propto \exp \left(-\alpha \sum_{k=1}^M x_k \log \left(\frac{x_k}{x_0} \right) \right),$$

and the Cauchy distribution prior

$$p_{pr}(x) \propto \prod_{k=1}^M \frac{1}{1 + \lambda x_k^2}.$$

2.4.3 Maximum Likelihood Reconstruction for Emission Tomography

The maximum likelihood reconstruction method for emission tomography involves estimating the parameters of interest, such as the number of photon counts, by maximizing the likelihood function given the observed data. The likelihood function typically models the statistical relationship between the observed data (e.g. photon counts) and the underlying activity distribution.

A mathematical model for emission tomography was proposed by [38], which aims to reconstruct an unknown emission density λ . All calculations for this model are based on [38]. The emission is generated by a radioactive isotope in an object. Each emission moves and is detected by a detector tube, or is missed. The emission density λ needs to be estimated from the total number $n^*(d)$ of emissions detected by the d th detector unit, with $d = 1, \dots, D$, where D represents the total number of detector tubes.

For convenience, the emission density λ is discretised into B boxes. Denote each box by b and the number of unobserved counts in each box b by $n(b)$. Assume a nearly perfect Poisson model (Section 2.3 Poisson noise) in which $n(b)$ follows a Poisson distribution with unknown means $\lambda(b)$,

$$p(n(b) = k) = e^{-\lambda(b)} \frac{\lambda(b)^k}{k!}, \quad k = 0, 1, \dots.$$

Let $n(b, d)$ be the variable representing the number of emissions in box b that are detected in tube d , and let

$$p(b, d) = p(\text{detected in } d | \text{emitted in } b)$$

be the probability of each emission in box b that is detected in tube d . Then, $n(b, d)$ follows a Poisson distribution with mean $\lambda(b, d) = \lambda(b)p(b, d)$. Denote the number of observed counts in the d th detector tube by

$$n^*(d) = \sum_{b=1}^B n(b, d), \tag{2.13}$$

and note that the number of (unobserved) emissions in the b th box can be expressed as

$$n(b) = \sum_{d=1}^D n(b, d). \quad (2.14)$$

Let $y = \{y_d\}$ be the complete collection of counts. Clearly, the observed data $y_d = \sum_b n(b, d)$. Following the Poisson noise model in Section 2.3, the photons emitted from b detected by bin d forms a Poisson process. Therefore, y are independent Poisson random variables, with

$$y_d \sim \text{Poisson} \left(\sum_b p(b, d) \lambda(b) \right).$$

The likelihood function of the emission density λ is

$$L(\lambda) = p(n^*(d) | \lambda) = \sum_A \prod_{\substack{b=1, \dots, B \\ d=1, \dots, D}} e^{-\lambda(b, d)} \frac{\lambda(b, d)^{n(b, d)}}{n(b, d)!}, \quad (2.15)$$

where the sum is over the set A of $n(b, d)$'s with

$$A = \left\{ n(b, d) | n^*(d) = \sum_{b=1}^B n(b, d) \right\},$$

restricted to $n^*(d)$.

Taking the partial derivative of Eq. 2.15 with respect to $\lambda(b_0)$,

$$\frac{\partial L(\lambda)}{\partial \lambda(b_0)} = \sum_A \left\{ \prod_{\substack{b=1, \dots, B \\ d=1, \dots, D}} e^{-\lambda(b, d)} \frac{\lambda(b, d)^{n(b, d)}}{n(b, d)!} \right\} \left\{ -1 + \frac{1}{\lambda(b_0)} \sum_{d=1}^D n(b_0, d) \right\}. \quad (2.16)$$

Equating Eq. 2.16 to zero,

$$\left\{ \prod_{\substack{b=1, \dots, B \\ d=1, \dots, D}} e^{-\lambda(b, d)} \frac{\lambda(b, d)^{n(b, d)}}{n(b, d)!} \right\} \left\{ -1 + \frac{1}{\lambda(b_0)} \sum_{d=1}^D n(b_0, d) \right\} = 0.$$

Therefore, the maximum likelihood estimate $\hat{\lambda}(b_0)$ of $n^*(d)$ is

$$\hat{\lambda}(b_0) = \sum_{d=1}^D n(b_0, d) = n(b). \quad (2.17)$$

The following steps that computes the expected value of $n(b, d)$ correspond precisely to the expectation step in the EM algorithm. The conditional probability of $n(b, d)$ given y_d is

$$\begin{aligned} p(n(b, d) | y_d) &= \frac{p(n(b, d) \text{ and } y_d)}{p(y_d)} \\ &= \frac{\left(\frac{e^{-p(b, d) \lambda(b)} (p(b, d) \lambda(b))^{n(b, d)}}{n(b, d)!} \right) \left(\frac{e^{(-\sum_{b' \neq b} p(b', d) \lambda'(b))} (\sum_{b' \neq b} p(b', d) \lambda'(b))^{y_d - n(b, d)}}{(y_d - n(b, d))!} \right)}{\frac{e^{(-\sum_{b'} p(b', d) \lambda'(b))} (\sum_{b'} p(b', d) \lambda'(b))^{y_d}}{y_d!}} \\ &= \frac{y_d!}{n(b, d)! (y_d - n(b, d))!} \frac{(p(b, d) \lambda(b))^{n(b, d)} (\sum_{b' \neq b} p(b', d) \lambda'(b))^{y_d - n(b, d)}}{(\sum_{b'} p(b', d) \lambda'(b))^{y_d}}. \end{aligned} \quad (2.18)$$

Then, the conditional expectation of $n(b, d)$ given y_d is

$$\begin{aligned}
E[n(b, d)|y_d] &= \sum_{n(b, d)=0}^{y_d} p(n(b, d)|y_d) n(b, d) \\
&= \sum_{n(b, d)=0}^{y_d} n(b, d) \frac{y_d!}{n(b, d)!(y_d - n(b, d))!} \frac{(p(b, d)\lambda(b))^{n(b, d)} \left(\sum_{b' \neq b} p(b', d)\lambda'(b)\right)^{y_d - n(b, d)}}{\left(\sum_{b'} p(b', d)\lambda'(b)\right)^{y_d}} \\
&= \frac{1}{\left(\sum_{b'} p(b', d)\lambda'(b)\right)^{y_d}} y_d p(b, d)\lambda(b) \sum_{n(b, d)=0}^{y_d} \left(\frac{(y_d - 1)!(p(b, d)\lambda(b))^{n(b, d)-1}}{(n(b, d) - 1)!(y_d - 1 - (n(b, d) - 1))!} \right. \\
&\quad \left. \left(\sum_{b' \neq b} p(b', d)\lambda'(b) \right)^{y_d - 1 - (n(b, d) - 1)} \right) \\
&= \frac{1}{\left(\sum_{b'} p(b', d)\lambda'(b)\right)^{y_d}} y_d p(b, d)\lambda(b) \left(\sum_{b'} p(b', d)\lambda'(b) \right)^{y_d - 1} \\
&= \frac{y_d p(b, d)\lambda(b)}{\sum_{b'} p(b', d)\lambda'(b)}.
\end{aligned}$$

Therefore, $n(b, d)$ can be estimated as

$$n(b, d) = \frac{y_d p(b, d)\lambda(b)}{\sum_{b'} p(b', d)\lambda'(b)}. \quad (2.19)$$

Theorem 2.4.3. *The log-likelihood function of Eq. 2.15*

$$l(\lambda) = \log(L(\lambda)) = \sum_A \sum_{\substack{b=1, \dots, B \\ d=1, \dots, D}} \left(-\lambda(b, d) + \log \left(\frac{\lambda(b, d)^{n(b, d)}}{n(b, d)!} \right) \right)$$

is concave with respect to λ .

Proof. The partial derivative of $l(\lambda)$ with respect to $\lambda(b_0)$ is

$$\frac{\partial l(\lambda)}{\partial \lambda(b_0)} = -1 + \sum_{d=1}^D \frac{n^*(d)p(b_0, d)}{\sum_{b'=1}^B \lambda(b')p(b', d)}. \quad (2.20)$$

The conditional expectation of $n(b_0)$ is

$$E[n(b_0)|n^*, \lambda] = \frac{1}{P(n^*|\lambda)} \sum_A \prod_{\substack{b=1, \dots, B \\ d=1, \dots, D}} e^{-\lambda(b, d)} \frac{\lambda(b, d)^{n(b, d)}}{n(b, d)!} n(b_0).$$

Using Eq. 2.16 and Eq. 2.14,

$$\begin{aligned}
\frac{\partial l(\lambda)}{\partial \lambda(b_0)} &= \frac{1}{L(\lambda)} \frac{\partial L(\lambda)}{\partial \lambda(b_0)} \\
&= \frac{1}{P(n^*|\lambda)} \left\{ -P(n^*|\lambda) + \frac{P(n^*|\lambda)}{\lambda(b_0)} E[n(b_0)|n^*, \lambda] \right\} \\
&= -1 + \frac{1}{\lambda(b_0)} \sum_{d=1}^D E[n(b_0, d)|n^*, \lambda].
\end{aligned} \quad (2.21)$$

Because of Eq. 2.13, and the fact that $n(b, d)$ are independent, the d th term in Eq. 2.21 is

$$E[n(b_0, d)|n^*, \lambda] = E[n(b_0, d)|n^*(d), \lambda] = \frac{n^*(d)\lambda(b_0, d)}{\sum_{b'=1}^B \lambda(b', d)}. \quad (2.22)$$

Eq. 2.22 and Eq. 2.21 yields Eq. 2.20. From Eq. 2.20,

$$\frac{\partial^2 l(\lambda)}{\partial \lambda(b_0) \partial \lambda(b_1)} = - \sum_{d=1}^D \frac{n^*(d)p(b_0, d)p(b_1, d)}{[\sum_{b'=1}^B \lambda(b')p(b', d)]^2}.$$

Therefore, for any $z(1), \dots, z(B)$,

$$\sum_{b_0=1}^B \sum_{b_1=1}^B z(b_0)z(b_1) \frac{\partial^2 l(\lambda)}{\partial \lambda(b_0) \partial \lambda(b_1)} = - \sum_{d=1}^D n^*(d)c^2(d)$$

where

$$c(d) = \frac{\sum_{b=1}^B z(b)p(b, d)}{\sum_{b'=1}^B \lambda(b')p(b', d)}.$$

Since $n^*(d)c^2(d) \geq 0$, the quadratic equation is negative semi-definite, and so $l(\lambda)$ is concave. \square

Since $l(\lambda)$ is concave, it follows from [43, Theorem 2.19(e)] that the KT conditions [42] are sufficient conditions for $\hat{\lambda}$ to be a maximiser of $l(\lambda)$. In this case, for each $b = 1, \dots, B$,

$$0 = \lambda(b) \frac{\partial l(\lambda)}{\partial \lambda(b)} \Big|_{\hat{\lambda}} = -\hat{\lambda}(b) + \sum_{d=1}^D \frac{n^*(d)\hat{\lambda}(b)p(b, d)}{\sum_{b'=1}^B \hat{\lambda}(b')p(b', d)} \quad (2.23)$$

and

$$\frac{\partial l(\lambda)}{\partial \lambda(b)} \Big|_{\hat{\lambda}} \leq 0 \quad \text{if } \hat{\lambda}(b) = 0.$$

The EM algorithm [8] can be used to choose an estimate $\hat{\lambda}$ that maximises the likelihood $L(\lambda)$, or equivalently, maximises the log-likelihood $l(\lambda) = \log(L(\lambda))$,

$$\hat{\lambda}^{new}(b) = \frac{\hat{\lambda}^{old}(b)}{\sum_{d=1}^D p(b, d)} \sum_{d=1}^D \frac{n^*(d)p(b, d)}{\sum_{b'=1}^B \hat{\lambda}^{old}(b')p(b', d)}, \quad b = 1, \dots, B, \quad (2.24)$$

where $\hat{\lambda}^{old}(b')$ and $\hat{\lambda}^{new}(b)$ denote the current estimate of $\lambda(b)$ and the new estimate, respectively. Because of the expression for the right hand-side of Eq. 2.23 (derived from Eq. 2.20), $\hat{\lambda}(b)$ can be the estimate for the emission counts. To be consistent, $\lambda(b)$ should be estimated by $\hat{\lambda}(b)$, which is the maximisation step

$$\hat{\lambda}^{new}(b) = E[n(b)|\hat{\lambda}^{old}, n^*], \quad b = 1, \dots, B. \quad (2.25)$$

As shown in Eq. 2.21, the right hand-side of Eq. 2.25 is the right hand-side of Eq. 2.24. Therefore, Eq. 2.24 is an EM algorithm.

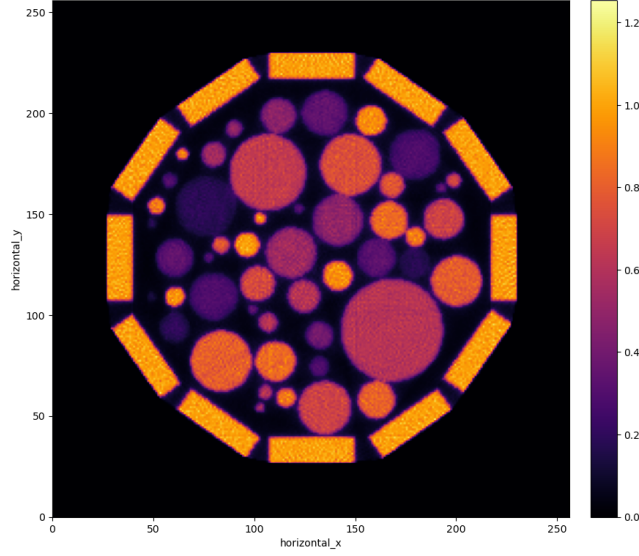


Figure 2.8: Reconstructed image under MLEM algorithm after 100 iterations. The ground truth is the same as Figure 2.7.

2.4.4 Bayesian Reconstruction Based on MLEM Algorithm

The Bayesian reconstruction method [14] that is going to be discussed in this section is based on the MLEM algorithm in Section 2.4.3. The photon counts are modelled in the same way as in the MLEM model; however, this method additionally models the prior information and aims at maximising the posterior distribution.

Assuming reconstruction over a three-dimensional space, with the space subdivided into B pixels. Denote $\lambda(b)$ as the isotope concentration in pixel b , $b = 1, 2, \dots, B$ and denote λ as the whole array $\{\lambda(b)\}$. Assume that the emissions from pixel b is random. Denote d as the d th detector bin, where $d = 1, 2, \dots, D$. Let $y = \{y_d\}$ be the complete collection of counts. The conditional probability of y given λ is

$$p(y|\lambda) = \prod_d \frac{\exp(-\sum w_{sr}\lambda(b)) (w_{sr}\lambda(b))^{y_d}}{y_d!},$$

where w_{sr} is the weight, which will be modelled later. Following [12] and [13], the prior information can be modelled as

$$p(\lambda) \propto \exp(-\beta V(\lambda)) = \exp\left(-\beta \sum_{s,r} w_{sr} \phi\left(\frac{\lambda(s) - \lambda(r)}{\delta}\right)\right), \quad (2.26)$$

where β and δ are parameters. w_{sr} is defined as: $w_{sr} = 1$ if pixel s and r are orthogonal nearest neighbours, $w_{sr} = \sqrt{\frac{1}{2}}$ if pixel s and r are diagonal neighbours, and $w_{sr} = 0$ otherwise. The function ϕ needs to be non-negative and symmetric about 0. In [12] and [13], $\phi(u) = \frac{1}{1+u^2}$, and [14] uses $\phi(u) = c_1 \log \cosh(c_2)$ and take $c_1 = \frac{27}{128}$ and $c_2 = \frac{16}{3\sqrt{3}}$.

This Bayesian reconstruction aims to choose a λ to maximise the posterior probability, or alternatively, the log posterior probability

$$\begin{aligned} \log p(\lambda|y) &= \log p(y|\lambda) + \log p(\lambda) + C \quad \text{where } C \text{ is a constant} \\ &= \sum_d \left(y_d \log \left(\sum_b w_{sr} \lambda(b) \right) - \sum_b w_{sr} \lambda(b) \right) - \beta V(\lambda) + C, \end{aligned} \quad (2.27)$$

where $V(\lambda)$ is given by Eq. 2.26. Since $\log p(\lambda|y)$ is not a quadratic function, some iteration algorithm is needed. It is obvious that traditional Newton's method or gradient method are not applicable because this problem is of large size. Therefore, EM algorithm [8] is more practicable. EM algorithm is a generic algorithm that is used for maximising the likelihood or posterior distribution in the case that some data is missing. In this problem, it is difficult to observe $n(b, d)$, the number of photons detected by the d th detector bin emitted from pixel b .

The expectation (E) step in Section 2.4.3 Eq. 2.17 is the same for each iteration. The maximisation (M) step is to maximise $\log p(\lambda|y)$, or alternatively,

$$\log p(\lambda|n) = \sum_{d,b} (n(b, d) \log(w_{sr} \lambda(b)) - w_{sr} \lambda(b)) - \beta V(\lambda) + C \quad \text{where } C \text{ is a constant.} \quad (2.28)$$

By differentiating Eq. 2.28, the equation that needs to be solved is

$$\frac{\sum n(b, d)}{\lambda(b)} - \sum w_{sr} - \beta \frac{\partial}{\partial \lambda(b)} V(\lambda) = 0. \quad (2.29)$$

[15] proposed a “one-step-late” (OSL) approximation for this posterior probability problem to evaluate the partial derivatives of $V(\lambda)$ in Eq. 2.29 at the current estimate $\hat{\lambda}^{old}$. Hence, the updating equation for estimating the new estimate $\hat{\lambda}^{new}$ from the old $\hat{\lambda}^{old}$ is

$$\hat{\lambda}_b^{new} = \frac{n(b, d)}{\sum w_{sr} + \beta \frac{\partial}{\partial \lambda(b)} V(\lambda)|_{\hat{\lambda}^{old}}}.$$

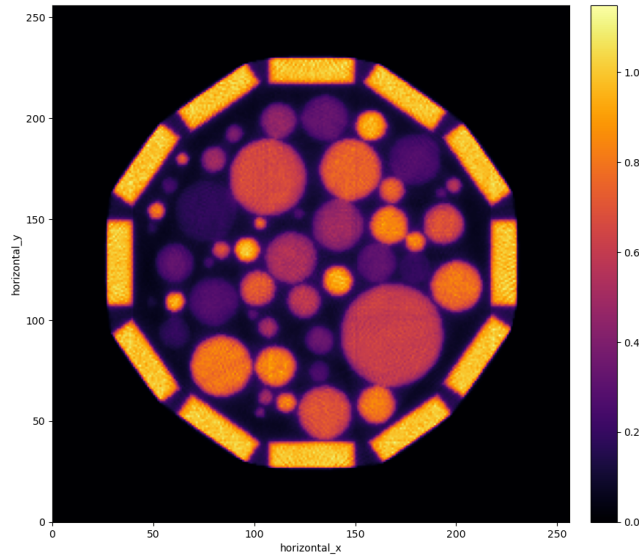


Figure 2.9: Reconstructed image under Bayesian reconstruction when choosing $\phi(u) = \frac{1}{1+u^{-2}}$, after 50 iterations.

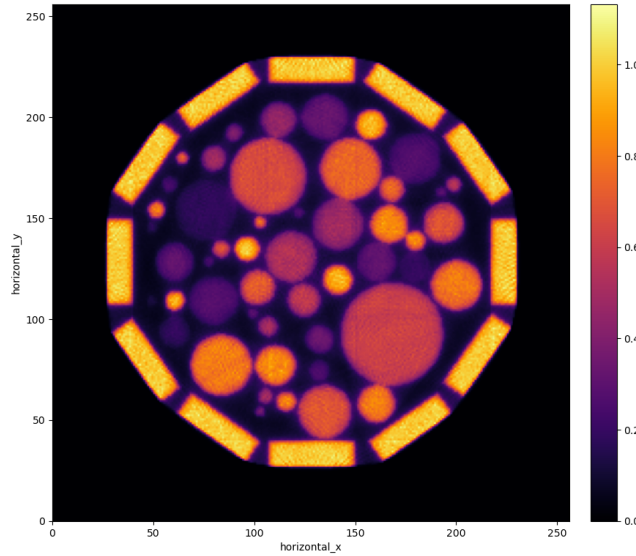


Figure 2.10: Reconstructed image under Bayesian reconstruction when choosing $\phi(u) = c_1 \log \cosh(c_2)$ where $c_1 = \frac{27}{128}$ and $c_2 = \frac{16}{3\sqrt{3}}$, after 50 iterations.

2.5 Regularisation Techniques for Tomography Problems

Tomographic reconstruction is generally considered an ill-posed inverse problem. An ill-posed problem may not have a unique solution. In an ill-posed problem, small changes in the input data can result in large changes in the output. Additionally, there might not be solutions satisfying the given conditions [25]. In CT problems, small errors in the acquired data can lead to significant distortions in the reconstructed image. In some cases, the number of projections is not sufficient compared to the number of pixels in an image, which is called underdetermined problems. This also leads to a non-unique set of solutions [20]. It has been observed that reconstructions exhibit sensitivity to measurement noise [17, Chap. 2]. Consequently, satisfactory results cannot be achieved by simply solving the system of linear equations. To address this, regularisation techniques are needed to stabilise the solutions [17, Chap. 12].

2.5.1 Total Variation Regularisation

A numerical algorithm for removing noise from images by minimising the total variation $TV(f)$ was proposed by [36],

$$TV(f) = \int |\nabla f(x)| dx, \quad \text{for } f \text{ continuous,}$$

where $\nabla f(x)$ represents the gradient of the image.

The formulation considered is the Tikhonov problem [16], which aims to find the image f that maximises

$$\hat{f} = \arg \min_f \int \frac{1}{2} \|b - Af\|_2^2 + \alpha(x) |\nabla f(x)| dx, \quad (2.30)$$

where A is the imaging system operator, b is the measured data and $\|\cdot\|_2$ is the L_2 norm. When $\alpha(x) = \alpha$ is a constant, Eq. 2.30 is simply

$$\hat{f} = \arg \min_f \frac{1}{2} \|b - Af\|_2^2 + \alpha TV(f). \quad (2.31)$$

The Tikhonov regularisation problem can be formulated using Bayesian inversion (Section 2.4.2) as follows

$$\hat{f} = \arg \min_f (-\log p(y|f) - \log p(f)), \quad (2.32)$$

where $p(y|f)$ is the likelihood model, representing the probability of observing the data given the image f and corresponding to b in Eq. 2.31, and $p(f)$ is the prior model, representing the prior information about the image f and corresponding to Af in Eq. 2.31.

If the image f is discrete, the total variation for f [4] is defined as

$$TV(f) = \sum_{i,j} |\nabla f_{i,j}|,$$

where $\nabla f_{i,j}$ is the gradient of the image at the pixel (i, j) .

In the Bayesian inversion (Section 2.4.2), total variation regularisation is used as a prior model to encourage the reconstructed image to be smooth.

This iterative reconstruction method based on solving specific convex optimisation problem has shown great success in improving reconstruction quality in some challenging cases, for example the low-dose X-ray computed tomography [44].

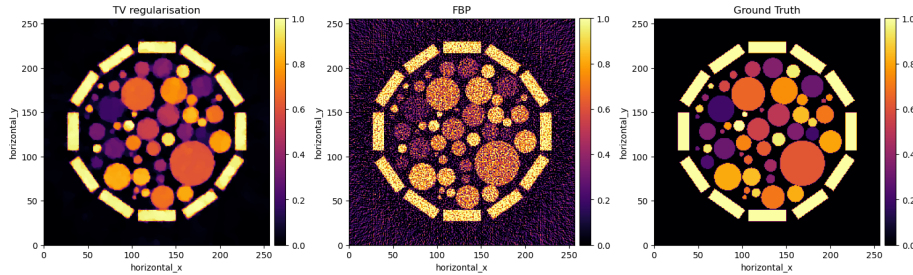


Figure 2.11: TV regularisation for the image data. The left figure is a reconstruction from noisy data using Eq. 2.30 and PDHG algorithm, which will be discussed later (Section 2.5.2). The middle figure is a reconstruction from the same data using the FBP method discussed earlier (Section 2.4.1). This figure is taken from the source [24].

2.5.2 Primal-Dual Hybrid Gradient Algorithm

The Primal-Dual Hybrid Gradient (PDHG) algorithm [5] is an iterative optimisation approach for the resolution of convex optimisation problems characterised by a specific structure. This algorithm is used in the scenarios where the objective involves minimising a sum of convex functions. It is useful for solving optimisation problems, particularly those involving total variation (TV) regularisation (Section 2.5.1). L1 regularisation [32] is also one of the classes of optimisation problems that can be solved by PDHG, which aims to find an image f that minimises

$$\hat{f} = \arg \min_f \frac{1}{2} \|b - Af\|_2^2 + \alpha \|f\|_1,$$

where A is the imaging system operator, b is the measured data, and $\|\cdot\|_1$ is the L_1 norm.

PDHG starts with an initial estimate for the primal value and the dual value. It then updates both variables using gradient descent steps. The iterative updates continue until the convergence criteria are met. In each iteration, the dual update step involves a projection operation that enforces constraints based on the TV regularisation.

Chapter 3

Results and Discussion

3.1 Overview of Core Imaging Library

In this chapter, we study several examples based on the Core Imaging Library (CIL) provided by [33] and [24]. The Core Imaging Library (CIL) is a versatile open-source Python framework for tomographic imaging, especially for reconstruction of challenging imaging data. In recent years, there have been numerous reconstruction methods for imaging, particularly total variation (TV) regularisation, which has been successfully applied to improving reconstruction quality. However, there is a lack of good reconstruction software to process this highly professional and time-consuming work. Existing reconstruction software usually focuses on optimising traditional data sets with high quality. It relies on the filtered back-projection (FBP) method, which does not adequately support incomplete, highly noisy and non-standard data sets. CIL solves this problem by combining the world's best tomography and optimisation software in an easy to use and configurable Python library.

3.1.1 CIL Framework

We will now introduce the CIL modules and packages that we use to study the examples.

AcquisitionGeometry

AcquisitionGeometry refers to the specific configuration and parameters of the data acquisition process. This configuration contains various aspects of constructing frameworks for parallel or cone beam 2D/3D data acquisition systems. We can obtain the data using commands such as *create_Parallel2D*, *create_Parallel3D*, *create_Cone2D* and *create_Cone3D*. We can arrange the detectors using commands such as *detector_position*, *detector_direction_x*, *detector_direction_y*, measure angles using commands such as *set_angles*, and adjust the positioning of the object being imaged relative to the detectors using commands such as *rotation_axis_position* and *rotation_axis_direction*. Essentially, it defines how the data is acquired during the imaging process.

ImageGeometry

The *ImageGeometry* class holds meta data describing the reconstruction volume of interest. The volume is centred on the position of the rotation axis defined in

AcquisitionGeometry, with the z-direction aligned with the rotation axis direction.

3.1.2 Data Containers

The *cil.framework.DataContainer* class is a generic class to hold data in a numpy array. The *cil.framework.AcquisitionData* class is a *DataContainer* that holds 2D or 3D sinogram data. The *cil.framework.ImageData* class is a *DataContainer* that holds 2D or 3D data.

3.1.3 Optimisation Framework

PDHG

The *cil.optimisation.algorithms.PDHG* class represents the Primal-Dual Hybrid Gradient (PDHG) algorithm (Section 2.5.2), with parameters such as functions f , g , the step size σ , the initial point $initial$, and the maximum number of iterations $max_iteration$.

3.1.4 Functions

L2 Norm Squared

The *cil.optimisation.functions.L2NormSquared* class calculates the squared L2 norm of the data. The formula for the squared L2 norm of a vector x is given by

$$F(x) = \|x\|_2^2 = \sum_i x_i^2,$$

$$\text{or } F(x) = \|x - b\|_2^2 \text{ if centred at } b.$$

The following code calculates the L2 norm of the error in Figure 2.11 using the PDHG regularisation algorithm in CIL.

```
gaussian_mean = 0
gaussian_variance = np.linspace(0,0.5,5)
L2Norm = []

for gaussian_vars in gaussian_variance:
    n12 = np.random.normal(gaussian_mean, gaussian_vars, size = ag.shape)

    noisy_sino2 = ag.allocate()
    noisy_sino2.fill(n12 + sino.array)
    noisy_sino2.array[noisy_sino.array<0]=0

    alpha = 0.1
    f11 = 0.5 * L2NormSquared(b=noisy_sino2)
    f21 = alpha * MixedL21Norm()
    f1 = BlockFunction(f11, f21)

    g1 = IndicatorBox(lower=0)

    normK = K.norm()
    sigma = 1./normK
    tau = 1./normK
```

```

pdhg1 = PDHG(f = f1, g = g1, operator = K, sigma = sigma, tau = tau,
             max_iteration = 200,
             update_objective_interval = 50)
pdhg1.run(verbose=2)

L2squared = L2NormSquared(b = phantom)
L2 = L2squared(pdhg1.solution)**(1/2)
L2Norm.append(L2)

```

Mixed L21 Norm

The *cil.optimisation.functions.MixedL21Norm* class calculates the mixed L21 norm of the data. The formula for the mixed L21 norm of a vector \mathbf{x} is given by

$$F(\mathbf{x}) = \|\mathbf{x}\|_{2,1} = \sum_{i=1}^n |x_i|_2 = \sum_{i=1}^n \sqrt{(x_1)^2 + (x_2)^2 + \dots},$$

where \mathbf{x} is a *BlockDataContainer*, i.e., $\mathbf{x} = (x_1, x_2, \dots)$.

3.1.5 Block Framework

The block framework is a modular approach which provides generic strategies for implementing reconstruction algorithms. It contains *BlockDataContainer*, *BlockFunction*, and *BlockOperator*.

The *cil.framework.BlockDataContainer* class is a data structure that holds *Data container*. We can do basic algebra between *BlockDataContainers* using different functions in this class. Examples of the block functions include the convex conjugate at the vector \mathbf{x}^*

$$F^*(\mathbf{x}^*) = \sum_{i=1}^m f_i^*(x_i^*),$$

and the gradient at the vector \mathbf{x}

$$F'(\mathbf{x}) = [f'_1(x_1), \dots, f'_m(x_m)].$$

3.2 Example Studies

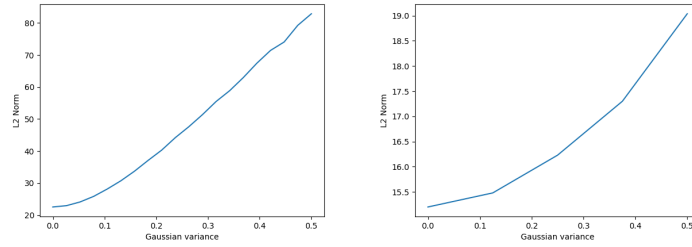
The data that we are going to study is the CIL demo provided by [33]. We use different reconstruction methods to study this example.

3.2.1 Reconstruction Using FBP and PDHG algorithm

In this section, we focus on the FBP method (Section 2.4.1) and the PDHG algorithm (Section 2.5.2). When we run the FBP algorithm, we obtain the reconstructed image shown in Figure 2.7, where noise is noticeable across the image. When we set the maximum number of iterations to 200 and run the PDHG algorithm, we obtain the reconstructed image in Figure 2.11 (left). Compared to the FBP algorithm, the image generated by the PDHG algorithm exhibits significantly less noise, although some noise remains visible at the edges of the circles in the middle.

When we change the simulated Gaussian variance for the sinogram and run the FBP method and the PDHG algorithm, we can compare the errors of these two methods to determine which one yields better reconstruction results for the given data. In this example, we choose 20 equally spaced values of Gaussian variance between 0 and 0.5 for the FBP method and 5 values of Gaussian variance within the same interval for

the PDHG algorithm because running the PDHG algorithm in CIL takes a long time. We can compute the L2 norm of the error for both methods, as mentioned in Section 3.1.4, and obtain the relationship between L2 norm and Gaussian variance by plotting them, as illustrated in Figure 3.1 and 3.2. It can be observed that there is a roughly linear relationship between the L2 norm and Gaussian variance. It is also obvious that the PDHG algorithm yields smaller error and it may be the better method to reconstruct the data.



(a) Plot of the L2 norm against Gaussian variance under the FBP method. (b) Plot of the L2 norm against Gaussian variance under the PDHG algorithm.

Figure 3.1: Plot of the L2 norm against Gaussian variance under the FBP method and the PDHG algorithm.

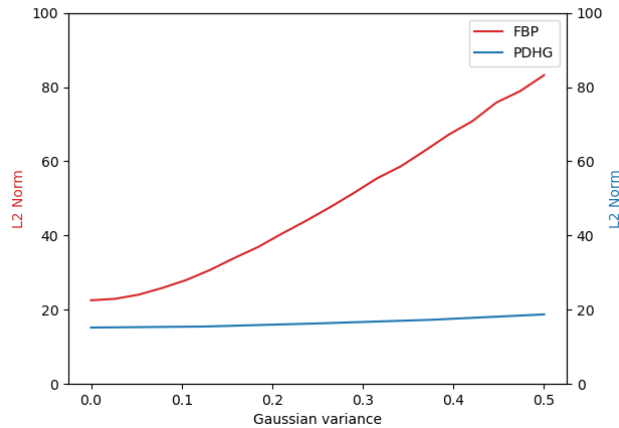


Figure 3.2: Plot of the L2 norm against Gaussian variance under the FBP method and the PDHG algorithm, combined in the same plot.

3.2.2 MLEM Reconstruction Method

In this section, we focus on the MLEM algorithm as described in Section 2.4.3. When we set the number of iterations to 100, we obtain the reconstructed image as shown in Figure 2.8. It can be observed that there is some noise throughout the image, as indicated in the middle of the circles and rectangles.

We can compute the L2 norm and KL divergence of the error for each number

of iterations. Additionally, we can plot the L2 norm and KL divergence against the number of iterations, as shown in Figure 3.3a and 3.3b, respectively. It can be observed that the L2 norm decreases with an increase in the number of iterations, while the KL divergence firstly decreases, reaches the minimum at about 50 iterations, and then increases as the number of iterations increases. Typically, the L2 norm and KL divergence both initially decrease and then increase as the number of iterations increases. This phenomenon is referred to as semi-convergence. Therefore, it may be necessary to introduce noise to this MLEM algorithm to observe if the L2 norm follows the expected pattern.

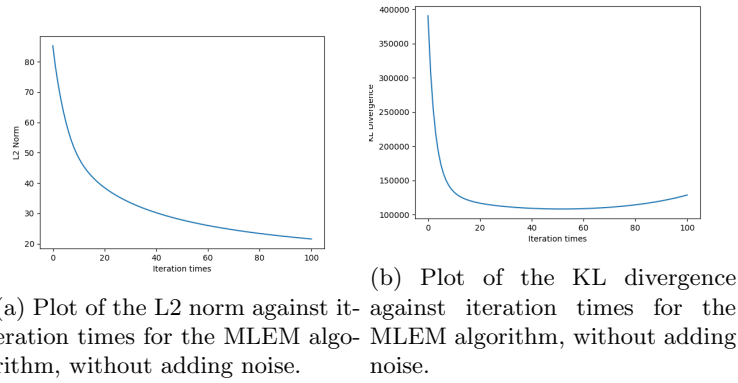


Figure 3.3: Plot of the L2 norm and KL divergence against iteration times for the MLEM algorithm, without adding noise.

When we add Gaussian noise with mean 0 and variance 0.5 to this MLEM algorithm, we observe that the L2 norm initially decreases, reaches the minimum at about 55 iterations, and then increases as the number of iterations increases. Together with the relationship between the KL divergence and the number of iterations discussed earlier, we can choose the optimal number of iterations for this MLEM algorithm between 50 and 55.

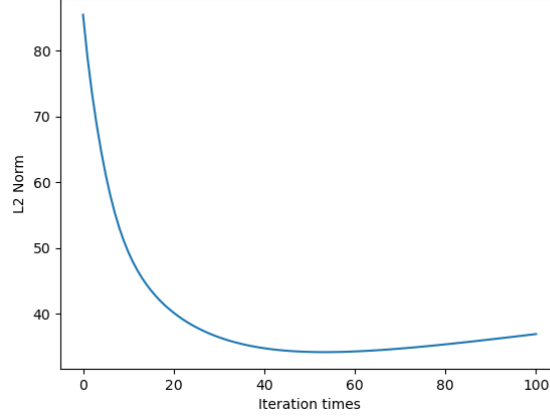


Figure 3.4: Plot of the L2 norm against iteration times for the MLEM algorithm, with added Gaussian noise with mean 0 and variance 0.5.

3.2.3 Bayesian Reconstruction Based on MLEM algorithm

In this section, we implement the Bayesian reconstruction method based on MLEM algorithm (Section 2.4.4) in CIL. When we choose $\phi(u) = \frac{1}{1+u-2}$ and $\phi(u) = c_1 \log \cosh(c_2)$ where $c_1 = \frac{27}{128}$ and $c_2 = \frac{16}{3\sqrt{3}}$, and set the number of iterations to 50, we obtain the reconstructed image as shown in Figure 2.9 and 2.10, respectively. Compared to the reconstructed image under the MLEM algorithm, the Bayesian reconstruction exhibits better reconstruction performance. The image exhibits less noise, although some noise remains visible in the middle of the circles and rectangles. We can also obtain the KL divergence of the error corresponding to each number of iterations, and plot the KL divergence against the number of iterations, as shown in Figure 3.5 and 3.6, respectively. We can observe that the KL divergence decrease as the number of iterations increases, and the rate of decrease approaches zero after 30 iterations. Therefore, we can choose the optimal number of iterations to be 30.

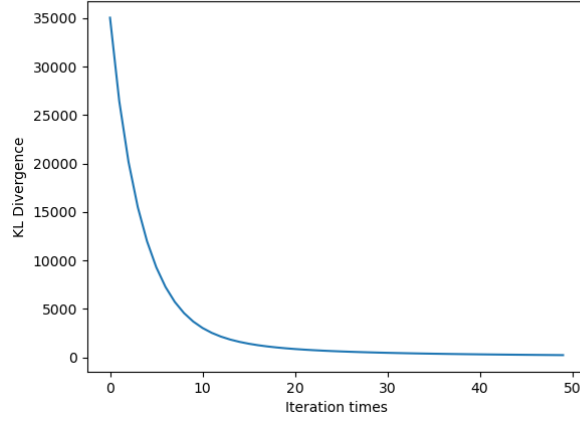


Figure 3.5: Plot of the KL divergence against the number of iterations under the Bayesian reconstruction method, when choosing $\phi(u) = \frac{1}{1+u^{-2}}$. There is no semi-convergence when we use a prior.

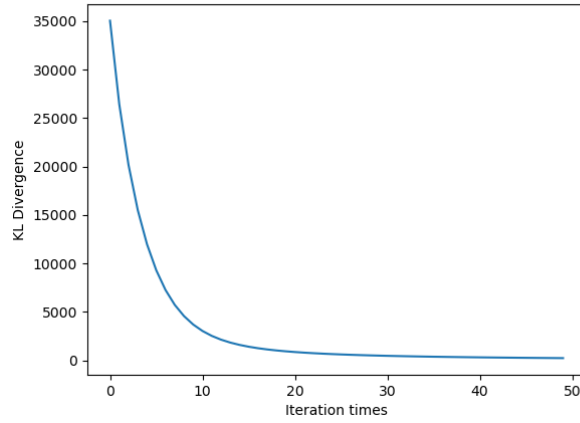


Figure 3.6: Plot of the KL divergence against the number of iterations under the Bayesian reconstruction method, when choosing $\phi(u) = c_1 \log \cosh(c_2)$ where $c_1 = \frac{27}{128}$ and $c_2 = \frac{16}{3\sqrt{3}}$.

Chapter 4

Appendix: Computer Code

4.1 FBP and PDHG

```
gaussian_mean = 0
gaussian_variance = np.linspace(0, 0.5, 20)
L2Norm = []

for gaussian_vars in gaussian_variance:
    n11 = np.random.normal(gaussian_mean, gaussian_vars, size = ag.shape)

    noisy_sino1 = ag.allocate()
    noisy_sino1.fill(n11 + sino.array)
    noisy_sino1.array[noisy_sino.array<0]=0

    fbp_recon1 = FBP(ig, ag, device = 'cpu')(noisy_sino1)

    L2squared = L2NormSquared(b = phantom)
    L2 = L2squared(fbp_recon1)**(1/2)
    L2Norm.append(L2)

fig, ax1 = plt.subplots()

color = 'tab:red'
ax1.set_xlabel("Gaussian variance")
ax1.set_ylabel("L2 Norm", color = color)
lns1 = ax1.plot(gaussian_variance, L2Norm, color = color, label = "FBP")

gaussian_mean = 0
gaussian_variance = np.linspace(0,0.5,5)
L2Norm = []

for gaussian_vars in gaussian_variance:
    n12 = np.random.normal(gaussian_mean, gaussian_vars, size = ag.shape)

    noisy_sino2 = ag.allocate()
    noisy_sino2.fill(n12 + sino.array)
```

```

noisy_sino2.array[noisy_sino.array<0]=0

alpha = 0.1
f11 = 0.5 * L2NormSquared(b=noisy_sino2)
f21 = alpha * MixedL21Norm()
f1 = BlockFunction(f11, f21)

g1 = IndicatorBox(lower=0)

normK = K.norm()
sigma = 1./normK
tau = 1./normK

pdhg1 = PDHG(f = f1, g = g1, operator = K, sigma = sigma, tau = tau,
             max_iteration = 200,
             update_objective_interval = 50)
pdhg1.run(verbose=2)

L2squared = L2NormSquared(b = phantom)
L2 = L2squared(pdhg1.solution)**(1/2)
L2Norm.append(L2)

ax2 = ax1.twinx()

a, b = 0, 100
ax1.set_ylim(a,b)
ax2.set_ylim(a,b)

color = 'tab:blue'
ax2.set_ylabel("L2 Norm", color = color)
lns2 = ax2.plot(gaussian_variance, L2Norm, color = color, label = "PDHG")

lns = lns1+lns2
labs = [l.get_label() for l in lns]
ax1.legend(lns, labs, loc = 0)

plt.savefig('l2gaussianvar.png')
plt.show

```

4.2 MLEM algorithm

```

x0 = ig.allocate(value = 1)
ones = ag.allocate(value = 1)
m = A.adjoint(ones)
for i in range(4):
    d = A.direct(x0)
    n = sino/d
    p = A.adjoint(n)
    x1 = p*x0
    xnew = x1/m
    x0 = xnew

ml5 = show2D(xnew, cmap = 'inferno')

```

```

ml5.save('ml5.png')

L2Norm = []

x00 = ig.allocate(value = 1)
ones1 = ag.allocate(value = 1)
m1 = A.adjoint(ones1)

for i in range(101):
    d1 = A.direct(x00)
    n1 = sino/d1
    p1 = A.adjoint(n1)
    x11 = p1*x00
    xnew1 = x11/m1
    x00 = xnew1

    L2squared = L2NormSquared(b = phantom)
    L2 = L2squared(xnew1)**(1/2)
    L2Norm.append(L2)

plt.plot(range(101), L2Norm)
plt.xlabel("Iteration times")
plt.ylabel("L2 Norm")
plt.savefig('l2.png')
plt.show()

KL = []

x000 = ig.allocate(value = 1)
ones11 = ag.allocate(value = 1)
m11 = A.adjoint(ones11)

for i in range(101):
    d11 = A.direct(x00)
    n11 = sino/d11
    p11 = A.adjoint(n11)
    x11 = p11*x000
    xnew11 = x11/m11
    x000 = xnew11

    b = sino.as_array()
    a = (A.direct(xnew11)).as_array()

    div = sp.kl_div(b, a)
    div_sum = np.sum(div)

```

```

KL.append(div_sum)

plt.plot(range(101), KL)
plt.xlabel("Iteration times")
plt.ylabel("KL Divergence")
plt.savefig('kl.png')
plt.show()

L2Norm = []

x00 = ig.allocate(value = 1)
ones1 = ag.allocate(value = 1)
m1 = A.adjoint(ones1)

for i in range(101):
    d1 = A.direct(x00)
    n1noise = noisy_sino/d1
    p1noise = A.adjoint(n1noise)
    x11noise = p1noise*x00
    xnew1noise = x11noise/m1
    x00 = xnew1noise

L2squared = L2NormSquared(b = phantom)
L2 = L2squared(xnew1noise)**(1/2)
L2Norm.append(L2)

plt.plot(range(101), L2Norm)
plt.xlabel("Iteration times")
plt.ylabel("L2 Norm")
plt.savefig('l2noise.png')
plt.show()

```

4.3 Bayesian Reconstruction Based on MLEM algorithm

```

# phi(x) = 1/(1 + x^(-2))

def phi(x):
    return 1/(1+x**2)

def phiprime(x):
    return 2*x/(1+x**2)**2

def vprime(x, delta):
    aaa = x.as_array()
    pad = np.pad(aaa, ((1,1),(1,1)))
    left = pad[1:-1,0:-2]

```

```

right = pad[1:-1,2:]
upper = pad[0:-2,1:-1]
lower = pad[2:,1:-1]
upperleft = pad[0:-2,0:-2]
upperright = pad[0:-2,2:]
lowerleft = pad[2:,0:-2]
lowerright = pad[2:,2:]
t1 = phiprime((aaa - left)/delta)*1/delta
t2 = phiprime((aaa - right)/delta)*1/delta
t3 = phiprime((aaa - upper)/delta)*1/delta
t4 = phiprime((aaa - lower)/delta)*1/delta
t5 = 1/2**(1/2)*phiprime((aaa - upperleft)/delta)*1/delta
t6 = 1/2**(1/2)*phiprime((aaa - upperright)/delta)*1/delta
t7 = 1/2**(1/2)*phiprime((aaa - lowerleft)/delta)*1/delta
t8 = 1/2**(1/2)*phiprime((aaa - lowerright)/delta)*1/delta

return t1 + t2 + t3 + t4 +t5 + t6 + t7 + t8

KL = []

x0 = ig.allocate(value = 1)
ones = ag.allocate(value = 1)
m = A.adjoint(ones)
beta = 1000
for i in range(50):
    d = A.direct(x0)
    n = sino/d
    p = A.adjoint(n)
    x1 = p*x0
    xnew = x1/(m + beta*2*vprime(x0, 1000))
    x0 = xnew

    b = sino.as_array()
    a = (A.direct(xnew)).as_array()

    div = sp.kl_div(b, a)
    div_sum = np.sum(div)
    KL.append(div_sum)

plt.plot(range(50), KL)
plt.xlabel("Iteration times")
plt.ylabel("KL Divergence")
plt.savefig('mlkl1.png')
plt.show()

bayesian1 = show2D(xnew, cmap = 'inferno')
bayesian1.save('bayesian1.png')

# phi(x) = 27/128*log(cosh(16/(3*3**(1/2))x))

```



```

def phi(x):
    return 27/128*np.log(np.cosh(16/(3*3**(1/2))*x))

def phiprime(x):
    return 3/8*3**(1/2)*np.tanh(16*x/3*3**(1/2))

KL = []

x0 = ig.allocate(value = 1)
ones = ag.allocate(value = 1)
m = A.adjoint(ones)
beta = 1000
for i in range(50):
    d = A.direct(x0)
    n = sino/d
    p = A.adjoint(n)
    x1 = p*x0
    xnew = x1/(m + beta*2*vprime(x0, 1000))
    x0 = xnew

    b = sino.as_array()
    a = (A.direct(xnew)).as_array()

    div = sp.kl_div(b, a)
    div_sum = np.sum(div)
    KL.append(div_sum)

plt.plot(range(50), KL)
plt.xlabel("Iteration times")
plt.ylabel("KL Divergence")
plt.savefig('mlkl2.png')
plt.show()

bayesian2 = show2D(xnew, cmap = 'inferno')
bayesian2.save('bayesian2.png')

```

Bibliography

- [1] ALYSSAAVILA321. Allan macleod cormack, 2020. <https://beyondthesinglestory.wordpress.com/2020/11/09/allan-macleod-cormack/> [Accessed: 31/01/2024].
- [2] C. Baskin, N. Liss, E. Schwartz, E. Zheltonozhskii, R. Giryes, A. M. Bronstein, and A. Mendelson. Uniq: Uniform noise injection for non-uniform quantization of neural networks. *ACM Transactions on Computer Systems (TOCS)*, 37(1-4):1–15, 2021.
- [3] R. Boellaard, N. C. Krak, O. S. Hoekstra, and A. A. Lammertsma. Effects of noise, image resolution, and roi definition on the accuracy of standard uptake values: a simulation study. *Journal of Nuclear Medicine*, 45(9):1519–1527, 2004.
- [4] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20:89–97, 2004.
- [5] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40:120–145, 2011.
- [6] A. M. Cormack. Representation of a function by its line integrals, with some radiological applications. *Journal of applied physics*, 34(9):2722–2727, 1963.
- [7] A. M. Cormack and G. N. Hounsfield. The nobel prize in physiology or medicine 1979. *Nobelprize.org*, 9, 2011.
- [8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.
- [9] P. P. B. Eggermont, G. T. Herman, and A. Lent. Iterative algorithms for large partitioned linear systems, with applications to image reconstruction. *Linear algebra and its applications*, 40:37–67, 1981.
- [10] T. G. Flohr, S. Schaller, K. Stierstorfer, H. Bruder, B. M. Ohnesorge, and U. J. Schoepf. Multi-detector row ct systems and image-reconstruction techniques. *Radiology*, 235(3):756–773, 2005.
- [11] G. Friedland and B. Thurber. The birth of CT. *AJR. American journal of roentgenology*, 167(6):1365–1370, 1996.
- [12] D. Geman and S. Geman. Bayesian image analysis. In *Disordered systems and biological organization*, pages 301–319. Springer, 1986.
- [13] S. Geman. Statistical methods for tomographic image restoration. *Bull. Internat. Statist. Inst.*, 52:5–21, 1987.

- [14] P. J. Green. Bayesian reconstructions from emission tomography data using a modified em algorithm. *IEEE transactions on medical imaging*, 9(1):84–93, 1990.
- [15] P. J. Green. On use of the em algorithm for penalized likelihood estimation. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 52(3):443–452, 1990.
- [16] C. Groetsch. The theory of tikhonov regularization for fredholm equations. *104p, Boston Pitman Publication*, 1984.
- [17] P. C. Hansen, J. Jørgensen, and W. R. Lionheart. *Computed tomography: algorithms, insight, and just enough theory*. SIAM, 2021.
- [18] S. Helgason and S. Helgason. *The radon transform*, volume 2. Springer, 1999.
- [19] J. Hsieh. Computed tomography: principles, design, artifacts, and recent advances. 2003.
- [20] C. M. Hyun, S. H. Baek, M. Lee, S. M. Lee, and J. K. Seo. Deep learning-based solvability of underdetermined inverse problems in medical imaging. *Medical Image Analysis*, 69:101967, 2021.
- [21] G. Johansen. Gamma-ray tomography. In *Industrial Tomography*, pages 197–222. Elsevier, 2015.
- [22] A. Johnson and M. P. Hartung. Godfrey hounsfield, 2023. <https://litfl.com/godfrey-hounsfield/> [Accessed: 31/01/2024].
- [23] T. R. Johnson. Dual-energy CT: general principles. *American Journal of Roentgenology*, 199(5-supplement):S3–S8, 2012.
- [24] J. S. Jørgensen, E. Ametova, G. Burca, G. Fardell, E. Papoutsellis, E. Pasca, K. Thielemans, M. Turner, R. Warr, W. R. Lionheart, et al. Core imaging library-part i: a versatile python framework for tomographic imaging. *Philosophical Transactions of the Royal Society A*, 379(2204):20200192, 2021.
- [25] S. I. Kabanikhin. Definitions and examples of inverse and ill-posed problems. 2008.
- [26] W. A. Kalender. X-ray computed tomography. *Physics in medicine & Biology*, 51(13):R29, 2006.
- [27] R. Ketcham. X-ray computed tomography (ct), 2023. https://serc.carleton.edu/research_education/geochemsheets/techniques/CT.html [Accessed: 31/01/2024].
- [28] M. E. Ladd, P. Bachert, M. Meyerspeer, E. Moser, A. M. Nagel, D. G. Norris, S. Schmitter, O. Speck, S. Straub, and M. Zaiss. Pros and cons of ultra-high-field mri/mrs for human application. *Progress in nuclear magnetic resonance spectroscopy*, 109:1–50, 2018.
- [29] F. Luisier, T. Blu, and M. Unser. Image denoising in mixed poisson–gaussian noise. *IEEE Transactions on image processing*, 20(3):696–708, 2010.
- [30] V. Mosorov. The lambert-beer law in time domain form and its application. *Applied Radiation and Isotopes*, 128:1–5, 2017.
- [31] R. Nutt. The history of positron emission tomography. *Molecular Imaging & Biology*, 4(1):11–26, 2002.

- [32] N. Parikh, S. Boyd, et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014.
- [33] E. Pasca, J. S. Jørgensen, E. Papoutsellis, E. Ametova, G. Fardell, K. Thielemans, L. Murgatroyd, M. Duff, and H. Robarts. Core Imaging Library (CIL) (v23.1.0), 2023. <https://doi.org/10.5281/zenodo.8398330> [Accessed: 31/01/2024].
- [34] V. Petrik, V. Apok, J. A. Britton, B. A. Bell, and M. C. Papadopoulos. Godfrey hounsfield and the dawn of computed tomography. *Neurosurgery*, 58(4):780–787, 2006.
- [35] S. Pryse. Radio tomography: A new experimental technique. *Surveys in Geophysics*, 24:1–38, 2003.
- [36] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.
- [37] W. C. Scarfe and A. G. Farman. What is cone-beam ct and how does it work? *Dental Clinics of North America*, 52(4):707–730, 2008.
- [38] L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Transactions on Medical Imaging*, 1(2):113–122, 1982.
- [39] S. Siltanen, V. Kolehmainen, S. Järvenpää, J. Kaipio, P. Koistinen, M. Lassas, J. Pirttilä, and E. Somersalo. Statistical inversion for medical x-ray tomography with few radiographs: I. general theory. *Physics in Medicine & Biology*, 48(10):1437, 2003.
- [40] M. T. Vlaardingerbroek and J. A. Boer. *Magnetic resonance imaging: theory and practice*. Springer Science & Business Media, 2013.
- [41] P. J. Withers, C. Bouman, S. Carmignato, V. Cnudde, D. Grimaldi, C. K. Hagen, E. Maire, M. Manley, A. Du Plessis, and S. R. Stock. X-ray computed tomography. *Nature Reviews Methods Primers*, 1(1):18, 2021.
- [42] W. I. Zangwill. A decomposable nonlinear programming approach. *Operations Research*, 15(6):1068–1087, 1967.
- [43] W. I. Zangwill. Convergence conditions for nonlinear programming algorithms. *Management Science*, 16(1):1–13, 1969.
- [44] H. Zhang, J. Wang, D. Zeng, X. Tao, and J. Ma. Regularization strategies in statistical image reconstruction of low-dose x-ray ct: A review. *Medical physics*, 45(10):e886–e907, 2018.