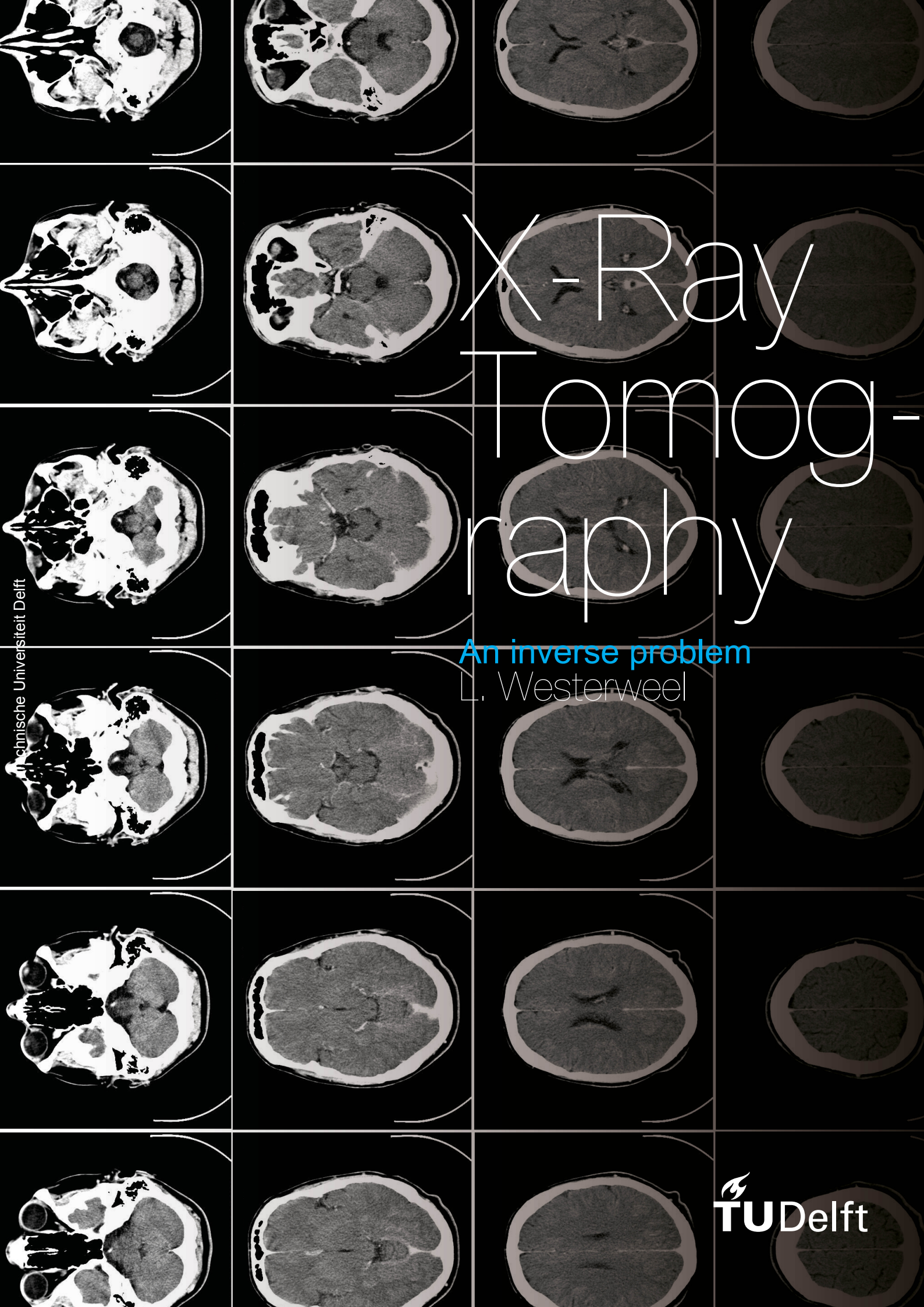


X-Ray Tomography

An inverse problem

L. Westerweel

Technische Universiteit Delft

TU Delft

X-Ray Tomography

An inverse problem

by

L. Westerweel

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended publicly on July 12, 2023 at 15:00.

Student number: 4792548
Project duration: April, 2023 – July, 2023
Thesis committee: Dr. H.N. Kekkonen, TU Delft, supervisor
Prof. dr. ir. M. B. van Gijzen, TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Layman's Abstract

A CT scan is a widely used medical imaging tool used by doctors all around the world. CT stands for computed tomography and it gives a view of the inside of a patient. In this way, a doctor can diagnose defects to internal organs or brain in a non-invasive way. This medical imaging technique involves a lot of mathematics. Since the measurement does not give a direct image, it has to be derived from the data. From this indirect measurement one has to extract only that information that is needed to make the CT image, but this is not self-evident. The measurement may contain noise caused by the machine. This additional information can make the CT image very blurry such that the subject is not recognizable in the image. This unwanted problem can be overcome by using different mathematical tools. These can enhance the CT image such that it is more accurate and that a doctor is able to make a correct diagnosis based on the image. This thesis describes several of these mathematical tools. They are derived and applied to CT measurements to make CT images. These results analyzed and compared to each other.

Abstract

This thesis aims at introducing the reader to the mathematical concepts behind the imaging technique X-ray tomography, also known as a CT scan. It includes the derivation of the filtered and unfiltered backprojection reconstruction methods for noise-free data. It was concluded that for noisy data, X-ray tomography is an ill-posed, and therefore unstable, inverse problem that needs regularization in order to produce adequate reconstructions. The methods of truncated singular value decomposition regularization and Tikhonov regularization are derived, analyzed and compared using simulated as well as real-life data. It was found that both methods can produce stable reconstructions, but that Tikhonov regularization is less sensitive to parameter choice.

Contents

1

Introduction

Medical imaging is a fundamental part of modern medicine. It is hard to imagine not being able to take an X-ray image of your broken arm or have an ultrasound when pregnant. However, this field of medicine is quite recent and started with the invention of X-rays by Wilhelm Röntgen in 1895 [1]. This invention and the application for all fields of medicine spread like wildfire among doctors.Not even two months after Röntgen took the first X-ray image of his wife's hand, the first X-ray assisted surgery took place. The years after, further specialised and more complicated techniques entered the scope of medical imaging. Following the emergence of computers, X-ray (computed) tomography was established in the 1970s. Often abbreviated with a CT scan, this type of scan works by combining multiple 2D X-ray scans taken from different angles to render a 3D view of the inside of a patient. These more detailed scans have a strong foundation in mathematics and in order to produce reconstructions lots of mathematical tools have to be used.

Figure 1.1: *Hand mit Ringen*, print of Wilhelm Röntgen's first X-ray, of his wife's hand, taken on 22 December 1895.

The objective of this thesis is to give an introduction to the mathematical fundamentals of X-ray tomography as an inverse problem, by showing different reconstruction methods and their mathematical derivations. The main topics in the thesis are based on chapters 1-5 of the book by Mueller and Siltanen [2]. The reconstruction methods in this thesis include filtered and unfiltered backprojection for perfect noise-free data, and minimum norm least-squares, truncated singular value decomposition regularization and Tikhonov regularization for noisy data. The described methods are shown and tested by making reconstructions of simulated data and real life data using MATLAB. The simulated data arises from the Shepp-Logan phantom, which is a commonly used standard test image that serves as the model of a human head, where one can determine the resolution themselves [3]. The real life data used contains CT measurements of a walnut of 82×82 pixels [4]. The produced reconstructed images will be analyzed on visual quality and error.

The thesis will be presented in the following structure. Chapter 2 provides an introduction to X-rays and the physical properties relevant to X-ray tomography. It includes the derivation of the fundamental attenuation law, which is the foundation of conventional reconstruction methods. In chapter 3 the case of continuous X-ray tomography is discussed in a situation where we have perfect noise-free data. The object that is measured is described by a continuous function and measurements can be done in a continuous way. The Radon transform is used to obtain a reconstruction of measurement called backprojection. A filtered version of this reconstruction, called filtered backprojection, is obtained with the additional use of the Fourier transform. The more applicable case of discrete X-ray tomography is then presented in chapter 4, from where noisy data is considered. Discrete measurements are taken, and the reconstruction is based on pixels with constant values. The chapter introduces X-ray tomography

as a discrete linear ill-posed inverse problem. A reconstruction is done based on the minimum norm solution and the pseudoinverse, after which the effects of ill-posedness are visibly shown. Finally, the concept of inverse crime is introduced and it is explained how one can avoid inverse crime when using simulated data. Chapter 5 contains two regularization methods to overcome instability. The first method is truncated singular value decomposition regularization, and the second method is Tikhonov regularization. Regularized reconstructions of the two methods for both the data sets are discussed. Finally, the two methods are compared. Lastly, the main takeaways and discussion points can be found in chapter 6.

2

X-rays

2.1. Introduction

When studying X-ray tomography it is important to understand some physical concepts and properties associated with X-rays as electromagnetic radiation. In this section the fundamental attenuation law will be derived, that will later be used in several reconstruction methods. In section 2.2 I will explain what electromagnetic radiation is and where X-rays fall on this spectrum. Sections 2.3 and 2.4 will go deeper into the physics behind taking an X-ray image, but these sections will not be used extensively further in the report. Finally section 2.5 describes the concept of attenuation and the fundamental attenuation law will be derived. The content of this chapter is based on the book by Prince and Links [5]

2.2. Electromagnetic Radiation

X-rays are a form of electromagnetic radiation. Other types of electromagnetic radiation include radio waves, microwaves, visible light and UV rays. Electromagnetic radiation can be considered a wave as well as moving particles.

- When viewing electromagnetic radiation as a wave, this wave is called an electromagnetic wave, where the frequency determines the type of electromagnetic radiation. Frequencies between $1.0 \times 10^5 - 3.0 \times 10^{10}$ Hz result in radio waves, while frequencies between $4.6 \times 10^{14} - 7.5 \times 10^{10}$ Hz make up visible light. X-rays used in medical imaging usually have frequencies between $3.0 \times 10^{18} - 3.0 \times 10^{19}$ Hz. The wavelength is given by

$$\lambda = \frac{c}{v},$$

  where c is the speed of light, and v is the frequency of the wave.

- When viewing electromagnetic radiation as a moving particle, it can be viewed as 'packets' of particles called photons. These photons have zero charge and zero mass, but carry energy. The energy of a photon is given by

$$E = hv, \tag{2.1}$$

  where h is Planck's constant and v is the frequency of the radiation in Hz, as mentioned above. The energy of a photon is given in unit of electron volts (eV), where $1\text{eV} = 1.6 \times 10^{-19}\text{J}$.

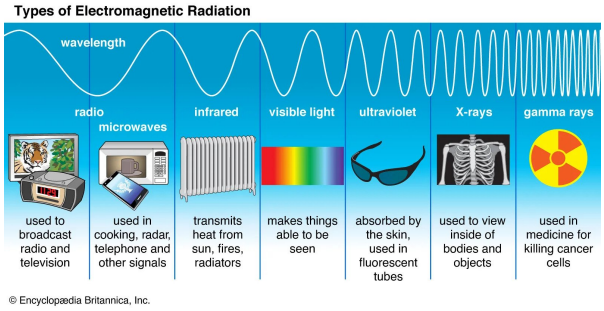


Figure 2.1: The different types of electromagnetic radiation, `https://www.britannica.com/science/electromagnetic-spectrum`.

2.3. Electron Binding Energy and Ionization

Electromagnetic radiation can interact with the material it is travelling through in multiple ways, including ionization. When an atom is ionized, an electron is ejected from the atom, leaving a free electron and an ion, which is an atom with positive charge. To understand how ionization affects an atom, one must understand the structure of an atom.

An atom consists of a core, called the nucleus, and electrons orbiting the nucleus. The nucleus is made up of neutrons and protons.

- Protons have positive charge and the number of protons defines which element the atom is.
- Neutrons are electrically neutral and there are approximately as many neutrons present in the nucleus as protons.
- Electrons have a negative charge and orbit the nucleus.

A whole atom has neutral charge, so there are as many positively charged protons as negatively charged electrons. The electrons orbit the nucleus in different orbits. The number of electrons that can fit in orbit n, is $2n^2$. This means that the first orbit has at most 2 electrons, the second has at most 8, the third at most18 and so on. In general, electrons first fill a lower orbit before moving on to the next higher orbit.

The energy of an atom plus the energy of a free electron is more than when the electron binds in an orbit of the atom. So when a free electron binds to an atom, there is remaining energy, called the electron binding energy, measured in electron volts (eV). The electron binding energy depends on the atom to which the electron binds and in which orbit the electron ends up. The electron binding energy decreases when the orbit number increases. However, it is sufficient to take an average electron binding energy in a given element. Metals have a high electron binding energy. For example, lead has an electron binding energy of about 1 keV, while air has an electron binding energy of about 34 eV. The electron binding energy of a single electron in a hydrogen atom is 13.6 eV.

If radiation, such as electromagnetic radiation, transfers energy to an electron in an atom that is greater than the electron binding energy, the electron is ejected. It becomes a free electron. This process is called ionization. Generally, radiation with energy greater than 13.6 eV is considered ionizing. Since high frequencies result in higher amounts of energies, only high-frequency electromagnetic radiation is ionizing. X-rays and gamma rays, for example, are ionizing, while visible light is not.

2.4. The Photoelectric Effect and Compton Scatter

When a photon interacts with an atom, a free electron is ejected, usually from the first orbit. The photon will be absorbed as energy and the ejected electron is called a photo-electron. This is called the photoelectric effect. Sometimes the missing electron in the orbit is filled up by an electron from a higher orbit. This produces radiation that can be harmful. It is therefore important to study this phenomenon, but this will not be included in this paper. For further reference, see

In Compton scatter the photon ejects an electron from one of the outer orbits of the atom. The photon is not completely absorbed, but loses energy. Due to the collision, the trajectory of the photon changes. The loss in energy depends on the scatter angle. The photon no longer travels in a straight line, which can affect the measurements and reconstructions of the measurement.

2.5. Attenuation

Attenuation is the reduction of the intensity of an X-ray beam as it travels through a medium. Here we consider an X-ray beam a short burst of X-rays. An X-ray beam has a strength, which is important to determine for the later reconstruction of the measurement. We will only consider narrow beam geometry, opposed to broad beam geometry, as it can be viewed as an accurate assumption from an imaging perspective.

When measuring the strength of an X-ray beam, one can consider mono-energetic or poly-energetic photons. If all photons in the beam have the same energy, the X-ray source is mono-energetic. If photons have different energies, the source is called poly-energetic. In practical X-ray imaging a source is always poly-energetic, due to the way the X-ray beam is produced. However, since the poly-energetic case is more complicated and can be based on the mono-energetic case, in the following we will only consider the mono-energetic case.

Consider the mono-energetic case. The photon fluence rate describes the number of photons N per unit area A in a fixed interval Δt, defined by

$$\phi = \frac{N}{A\Delta t}.$$

The intensity of an X-ray beam is given by

$$I = E\phi,$$

where E is the energy of a photon, given by equation (2.1). If an X-ray beam of N photons travels to a detector through a vacuum, the detector should measure again N photons. If now a thin slab is placed between the source and the detector, we expect the detector to measure $N' \leq N$ photons. This loss in photons and thus energy is the basic concept of attenuation. The number of lost photons is proportional to both N and Δx [5], or mathematically

$$\Delta N = -\mu N \Delta x,$$

where μ is called the linear attenuation coefficient. It can be rewritten as

$$\mu = \frac{-\Delta N/N}{\Delta x}. \tag{2.2}$$

Letting the slab become very thin we get the differential equation

$$\frac{dN}{N} = -\mu dx,$$

which gives

$$N = N_0 e^{-\mu \Delta x}, \tag{2.3}$$

where N_0 is the number of photons emitted from the beam at $x = 0$. This equation is called the fundamental attenuation law. In the mono-energetic case the intensity of the beam is a multiple of the number of photons, so we can also write the intensity as

$$I = I_0 e^{-\mu \Delta x}.$$

Now suppose that our slab is not homogeneous, but that the attenuation coefficient depends on the position x within the slab. This means that we should solve

$$\frac{dN}{N} = \mu(x) dx.$$

Integration gives that the number of photons at position x is given by

$$N(x) = N_0 e^{-\int_0^x \mu(x')dx'}.$$

Again, since we are considering the mono-energetic case, the same holds for the intensity of the beam.

$$I(x) = I_0 e^{-\int_0^x \mu(x')dx'},$$

or equivalently

$$\log(I_0) - \log(I(x)) = \int_0^x \mu(x')dx'.$$

This equation is called the integral form of the fundamental attenuation law, equation (2.3).

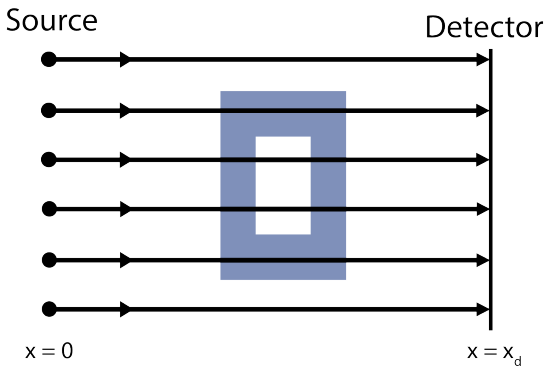


Figure 2.2: Visual representation of the fundamental attenuation law. The rectangle serves as the attenuation coefficient $\mu(x, y)$. The beams travel from $x = 0$ to $x = x_d$.

Suppose that we measure the beam intensity at a detector for a fixed x_d, say I_d. Then we have that

$$\log(I_0) - \log(I_d) = \int_0^{x_d} \mu(x)dx. \tag{2.4}$$

This means that the difference in the log of the intensity is determined by the integral over the linear attenuation coefficient. Since I_0 and I_d are known, we also know what the value of the integral is.
Note that the attenuation coefficient is a function in $\mathbb{R}^2$, so a point is given by (x, y). However, expression (2.4) only takes the integral over x, so we get

$$\log(I_0) - \log(I_d) = \int_{x_0}^{x_d} \mu(x, y)dx. \tag{2.5}$$

This expression relates something that can be measured, the log difference in intensity, with something that we would like to reconstruct in X-ray tomography, namely the attenuation coefficient $\mu(x, y)$. This equation will form the basis of the X-ray tomography reconstruction methods dicussed further in this report.

3

Continuous X-ray Tomography

3.1. Introduction

In X-ray tomography we want to reconstruct the inside of an object, based on measurements in difference in intensity of an X-ray beam that travels through the object. This section will consider the reconstruction methods of unfiltered and filtered backprojection for perfect, noise-free data. The contents of this chapter are based on chapter 2 of the book by Mueller and Siltanen [2]. In section 3.2 the Radon transform will be introduced and it is shown how this transform relates to the fundamental attenuation law derived in Chapter 2. Section 3.3 described the first reconstruction method of unfiltered backprojection, for noise-free data. Next, section 3.4 describes the Fourier transform that will be used in filtered backprojection. The Fourier transform and Radon transform shown to be related in section 3.5 by the central slice theorem. Finally, section 3.6 combines the content of the previous sections for the reconstruction method of filtered backprojection for noise-free data.

3.2. Radon Transform

In this section I will describe the definition of the Radon transform and how this relates to our derived equation for the difference in beam intensity, equation (2.4).

We interpret the $\theta \in \mathbb{R}$ as an angle. We denote the unit vector with angle θ with respect to the x-axis by $\vec{\theta} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \in \mathbb{R}^2$. Also, take $\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^2$.

Definition 3.2.1 (Radon transform)**.** *The Radon transform of the function $f(x, y)$ depends on the angular parameter θ and on the linear parameter $s \in \mathbb{R}$ in the following way:*

$$\Re f(s, \vec{\theta}) = \int_{\boldsymbol{x}\cdot\vec{\theta}=s} f(\boldsymbol{x}) d\boldsymbol{x}. \tag{3.1}$$

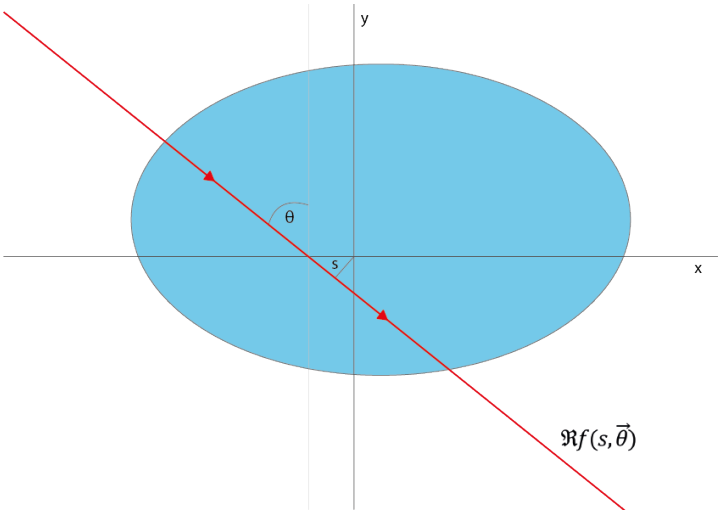


Figure 3.1: A visual representation of the radon transform. The radon transform is the line integral over the red line of the blue surface ($f(x)$), with angle θ and radius s.

A visual representation of the radon transform can be found in figure 3.1. An alternative notation for the Radon transform, using that

$$x(z) = z\sin\theta + s\cos\theta \tag{3.2}$$
$$y(z) = -z\cos\theta + s\sin\theta, \tag{3.3}$$

along the line $L(s,\theta) = \{(x,y)|x\cos\theta + y\sin\theta = s\}$ for some $z \in \mathbb{R}$, is given by:

$$\begin{aligned}\Re f(s,\theta) &= \int_{-\infty}^{\infty} f(x(z),y(z))dz \\ &= \int_{-\infty}^{\infty} f[(z\sin\theta + s\cos\theta),(-z\cos\theta + s\sin\theta)]dz.\end{aligned}$$

The Radon transform is a line integral of function $f(x,y)$ along the line with angle θ and radius s. We can compare the Radon transform of a function with derived equation (2.5). We can see that the difference in beam intensity along a line due to attenuation, is the same as the expression of the Radon transform of that attenuation coefficient. This can also be represented mathematically. If we take $\theta = \frac{\pi}{2}$, so that the line we integrate over in the Radon transform is a line parallel to the x-axis, we get the following expressions for x and y:

$$\begin{aligned}x(z) &= z\sin\frac{\pi}{2} + s\cos\frac{\pi}{2} = z \\ y(z) &= -z\cos\frac{\pi}{2} + s\sin\frac{\pi}{2} = s\end{aligned}$$

Using these expressions in the expression of the Radon transform of attenuation coefficient $\mu(x,y)$ we get:

$$\begin{aligned}\Re\mu(s,\vec{\theta}) &= \int_{\boldsymbol{x}\cdot\vec{\theta}=s} \mu(\boldsymbol{x})d\boldsymbol{x}, \\ &= \int_{-\infty}^{\infty} \mu(x(z),y(z))dz, \\ &= \int_{-\infty}^{\infty} \mu(z,s)dz, \\ &= \int_{-\infty}^{\infty} \mu(x,y)dx, \\ &= \int_{0}^{x_d} \mu(x,y)dx, \\ &= \log(I_0) - \log(I_d).\end{aligned} \tag{3.4}$$

We used that it is known that the X-ray beam travels from the source at $x_0 = 0$ to the detector at x_d. The Radon transform of the linear attenuation coefficient equals the log-difference in beam intensity after the ray traveled through the object. Since we know the value of $\log(I_0) - \log(I_d)$ through measurements, we also know the value of $\Re\mu(s, \vec{\theta})$ for a certain angle θ and radius s.

3.3. Backprojection

The difference in beam intensity of the X-ray after it has traveled through an object is not uniquely determined. Lots of attenuation coefficients can result in the same value of $\log(I_0) - \log(I_d)$ for a certain angle of measurement. A solution to this problem of non-uniqueness is to consider the measurement from different angles θ and combine the results.

Let $\Re\mu(s, \theta)$ be the Radon transform of the linear attenuation coefficient μ and take a fixed angle θ_0, see figure 3.2

$$\Re\mu(s, \theta_0) = \int_{\boldsymbol{x}\cdot\overrightarrow{\theta_0}=s} \mu(\boldsymbol{x})d\boldsymbol{x}.$$

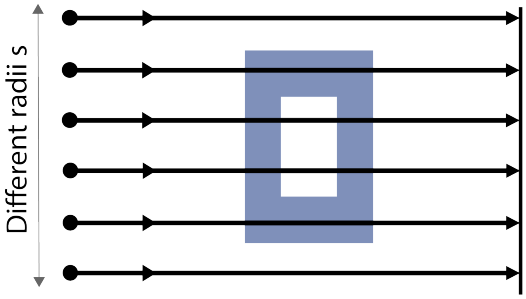
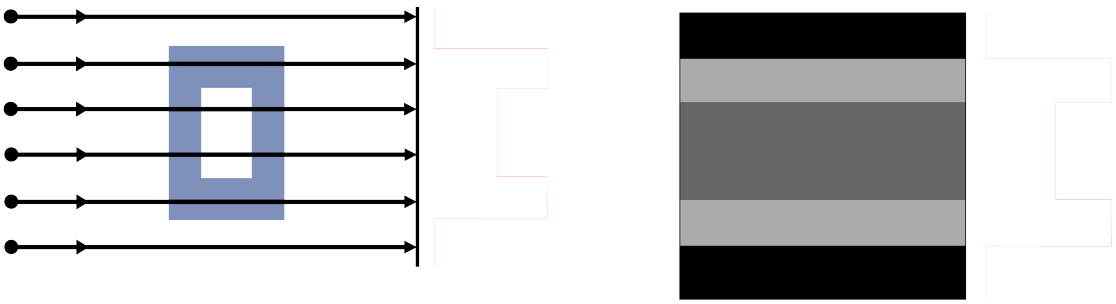


Figure 3.2: The Radon transform line integral for a fixed angle θ_0 but multiple radii s though a rectangle with a hole. The darker colours indicate lower valued measurements, and the white colours indicate higher value measurements.

This corresponds to the measurement of difference in beam intensity from an angle θ_0, as described in section 3.2. We can reason that if the value $\Re\mu(s, \theta_0)$ is large for a certain value s, then μ must be large somewhere along the line $\boldsymbol{x} \cdot \overrightarrow{\theta_0} = s$. We can fix s at s_0 and assign every point on the line $L(s_0, \theta_0) = \{(x, y) | x\cos(\theta_0) + y\sin(\theta_0) = s_0\}$ the value of $\Re\mu(s_0, \theta_0)$. This means that the points $(x(z), y(z))$, as given in equation (3.2) get the value $\Re\mu(s_0, \theta_0)$ for all z. If this is done for all values of s we get the backprojection image for angle θ_0:

$$b_{\theta_0}(x, y) = \Re\mu(s, \theta_0).$$

The measurement with a fixed angle of a rectangle with a hole inside can be seen in figure 3.3a. The backprojection image of this measurement is given in figure 3.3b.



(a) Measurements for a fixed angle and multiple radii s. The red line represents measurement $\Re\mu(s, \theta_0)$.

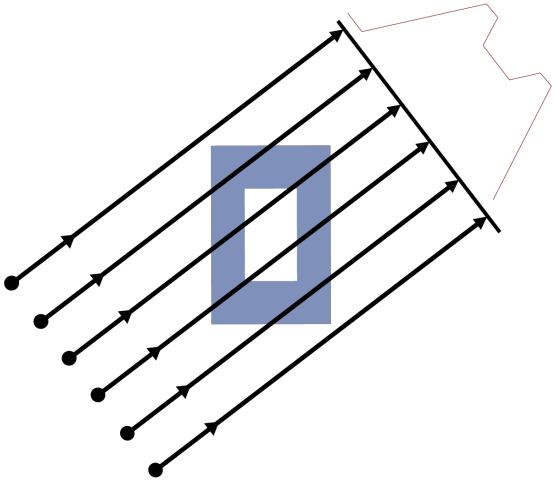
(b) The backprojection image for angle θ_0.

Figure 3.3: The measurement for a fixed angle θ_0 and it's corresponding backprojection image.

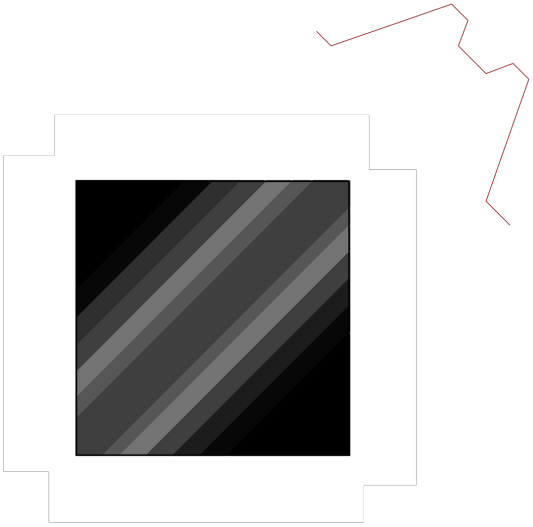
If we do this for all angles θ from 0 to π and overlay all obtained backprojection images by adding them up by integral, we get the backprojection summation image

$$\begin{aligned}\mu_b(x,y) &= \int_0^\pi b_\theta(x,y)d\theta \\ &= \int_0^\pi \Re\mu(s_0,\theta)d\theta \\ &= \int_0^\pi \log\frac{I_0(s_0,\vec{\theta})}{I_d(s_0,\vec{\theta})}d\theta. \end{aligned} \tag{3.5}$$

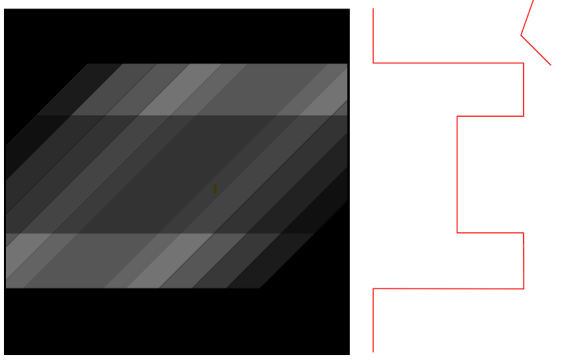
In figure 3.4 one can see a second measurement for a different angle of the same rectangle with a hole. The backprojection image of this second angled is combined with the backprojection image from figure 3.3 to form a combined backprojection summation image.



(a) Measurement for a different fixed angle and multiple radii s. The red line represents the measurement $\Re\mu(s,\theta_0)$.



(b) The backprojection image for a fixed angle.



(c) The combined backprojection summation images of two backprojection summation images.

Figure 3.4: The backprojection image for one fixed angle and the combined backprojection summation images for two different angles.

In figure 3.5 four different measurements of the rectangle with hole are combined to form a back-projection summation image. In this simple reconstruction we can, although vaguely, recognize the rectangle and the hole inside.

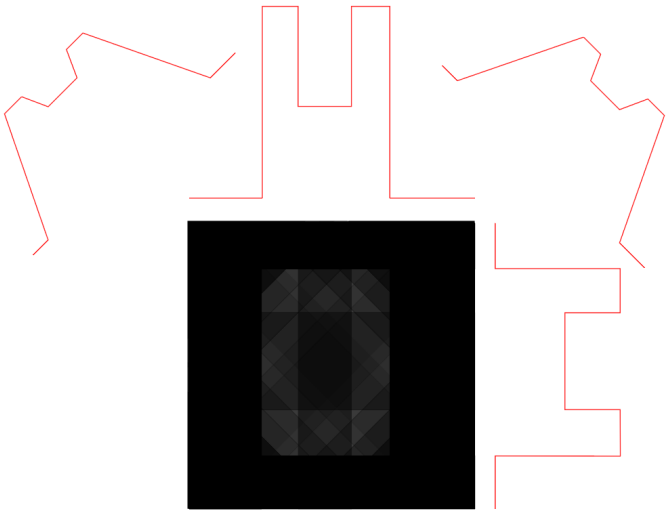
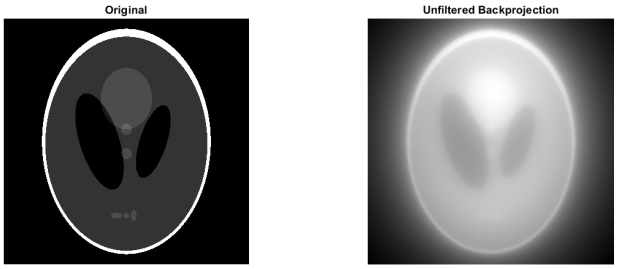


Figure 3.5: The backprojection summation image of the rectangle with a hole (see figure 3.2) for four different measurement angles.

Here $\mu_b(x, y)$ is a reconstruction of the original attenuation function $\mu(x, y)$. In figure 3.6b one can see the backprojection summation image of the Shepp-Logan phantom, compared to the original. One can see that the backprojection image is quite blurry, which might cause a problem when analyzing a patient's brain. This blurriness is the result of the way unfiltered backprojection works. It takes the measurements from one angle and spreads it out evenly in the reconstruction. If two points lie on a line with the same angle as the measurement angle, then these two points will receive the same reconstructed value for that angle, even if one point lies inside the object and the other not. As a result it is difficult to have clear sharp borders, as points that lie close to the object will also cumulatively receive a reconstruction value, while it should be zero.



(a) Original Shepp-Logan phantom. (b) Backprojection summation image.

Figure 3.6: The original Shepp-Logan phantom compared to the backprojection summation image.

3.4. Fourier Transform

The backprojection summation image found in equation (3.5) produces a reconstruction of the attenuation of the measured object, but the image is quite blurry. A way to get rid of this blurriness is to apply mathematical filters to the measured (noise-free) data. This can be done by a Fourier transform. In this section I will give the definition of a Fourier transform and some of its properties, and explain the process of how it can be applied in our case.

Definition 3.4.1 (Fourier transform)**.** *The Fourier transform of a function f defined on $\mathbb{R}^n$ is given by*

$$\mathfrak{F}f(\boldsymbol{x})(\boldsymbol{\xi}) = \hat{f}(\boldsymbol{\xi}) = \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} f(\boldsymbol{x}) e^{-i\boldsymbol{x}\cdot\boldsymbol{\xi}} d\boldsymbol{x}.$$

Then the Fourier transfomrs in $\mathbb{R}$ and $\mathbb{R}^2$ are expressed as

$$\text{In } \mathbb{R} : \mathfrak{F}f(x)(\xi) = \hat{f}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ix\xi} dx$$
$$\text{In } \mathbb{R}^2 : \mathfrak{F}f(x,y)(\xi_1,\xi_2) = \hat{f}(x,y) = \frac{1}{2\pi} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{-i(x\xi_1+y\xi_2)} dydx$$

Definition 3.4.2 (One-dimensional Fourier transform, scalar parameter)**.** *The one-dimensional Fourier transform of a function in the scalar parameter, $h(t,\vec{\theta})$ is given by*

$$\mathfrak{F}h(t,\vec{\theta})(s) = \tilde{h}(s,\vec{\theta}) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} h(t,\vec{\theta}) e^{-its} dt.$$

The Fourier transform decomposes a function f into complex exponentials. The transform gives the amplitude that corresponds to sine and cosine waves with amplitude ξ. By decomposing the function f into complex exponentials with frequencies ξ it is easier to filter out frequencies that cause the blurriness. By removing the blurring frequencies from the data, the reconstruction will be of a sharper, less blurry quality.

One can also describe the inverse Fourier transform, or the Fourier inversion formula.

Definition 3.4.3 (Fourier Inversion Formula)**.** *Let $f(\boldsymbol{x})$ be a function in $\mathbb{R}^n$. The Fourier inversion formula is given by*

$$\mathfrak{F}^{-1}f(\boldsymbol{\xi})(\boldsymbol{x}) = \hat{f}^{-1}(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} f(\boldsymbol{\xi}) e^{i\boldsymbol{x}\cdot\boldsymbol{\xi}} d\boldsymbol{\xi}.$$

Theorem 3.4.1 (Fourier Inversion Theorem)**.** *Let $\hat{f}(\boldsymbol{\xi})$ be the Fourier transform of a function $f(\boldsymbol{x})$ in $\mathbb{R}^n$, given by Definition 3.4.1. Then the Fourier inversion formula of a Fourier transform is again the original function $f(\boldsymbol{x})$,*

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} \hat{f}(\boldsymbol{\xi}) e^{i\boldsymbol{x}\cdot\boldsymbol{\xi}} d\boldsymbol{\xi},$$

that is

$$\mathfrak{F}^{-1}[\mathfrak{F}f(\boldsymbol{x})(\boldsymbol{\xi})](\boldsymbol{x}) = \mathfrak{F}^{-1}\hat{f}(\boldsymbol{\xi})(\boldsymbol{x}) = f(\boldsymbol{x}).$$

A proof of this theorem can be found in the text by Wong and Yam [6].

3.5. Central-slice Theorem

The Radon and Fourier transform are connected, known as the central-slice theorem. This theorem states that the 1D Fourier transform of the Radon transform of a function f equals a slice from the 2D Fourier transform of the same function f. The 1D Fourier transform equals a line passing through the origin of the 2D Fourier transform of the object the angle corresponding to the Radon transform.

I will first state the central-slice theorem and give a proof. After that I will derive an expression to reconstruct the attenuation coefficient based only on measurable information.

Theorem 3.5.1 (Central-slice Theorem)**.** *Let f be an absolutely integrable function defined on the whole real line. For any real number r and unit vector $\vec{\theta}$, we have the identity*

$$\widehat{\mathfrak{R}f}(r,\vec{\theta}) = \hat{f}(r\vec{\theta}).$$

Proof. Let f be an absolutely integrable function defined in the whole real line. Then from Definition 3.4.2 we get

$$\begin{aligned}
\widetilde{\mathfrak{R}f}(r,\vec{\theta}) &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\mathfrak{R}f(s,\vec{\theta})e^{-isr}ds,\\
&= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\left[\int_{\boldsymbol{x}\cdot\vec{\theta}=s} f(\boldsymbol{x})d\boldsymbol{x}\right]e^{-isr}ds,\\
&= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(x(z),y(z))dz\right]e^{-isr}ds,\\
&= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x(z),y(z))e^{-isr}dzds,
\end{aligned}$$

where we used the definition of the Radon transform, equation 3.1. We can now change the integration variables z and s to x and y by a variable transformation. We have to calculate the Jacobian:

$$\begin{aligned}
x = z\sin\theta + s\cos\theta \Rightarrow \frac{\partial x}{\partial z} = \sin\theta, \frac{\partial x}{\partial s} = \cos\theta,\\
y = -z\cos\theta + s\sin\theta \Rightarrow \frac{\partial y}{\partial z} = -\cos\theta, \frac{\partial y}{\partial s} = \sin\theta,
\end{aligned}$$

$$J = \left|(\sin\theta)^2 - (-\cos\theta)^2\right| = 1.$$

This change of variables gives

$$\begin{aligned}
\widetilde{\mathfrak{R}f}(r,\vec{\theta}) &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x(z),y(z))e^{-isr}dzds\\
&= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-i(x\cos\theta+y\sin\theta)r}dydx\\
&= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-i(xr\cos\theta+yr\sin\theta)}dxdy\\
&= \hat{f}(r\cos\theta, r\sin\theta)\\
&= \hat{f}(r\vec{\theta}).
\end{aligned}$$

□

Thus the central slice theorem says that if we put a Fourier transform on our Radon transform, which equals our measurement data in case of the attenuation coefficient, this is the same as the line $r\vec{\theta}$ in the 2D Fourier transform of the attenuation coefficient.

It is useful to obtain an expression of $f(x,y)$, or $\mu(x,y)$ in case of the attenuation coefficient, in terms of the measurable quantities including a filter by means of a Fourier transform. Note that the measurement data is perfect, meaning that it does not include noise. This is a way to write $f(x,y)$ given perfect data.

Theorem 3.5.2 (Radon inversion formula)**.** *If f is an absolutely integrable function defined on the real line and $\hat{f}$ is absolutely integrable, then*

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{2\pi}}\int_0^{\pi}\int_{-\infty}^{\infty}\widetilde{\mathfrak{R}f}(s,\vec{\theta})e^{is\boldsymbol{x}\cdot\vec{\theta}}|s|dsd\theta. \tag{3.6}$$

Before we prove this theorem, I state two lemmas and their proofs.

Lemma 3.5.3. *Denote the Radon transform of a function $f(x,y)$ in $\mathbb{R}^2$ as $\mathfrak{R}f(s,\vec{\theta})$, then*

$$\mathfrak{R}f(s,\vec{\theta}) = \mathfrak{R}f(-s,-\vec{\theta}).$$

Proof. The Radon transform of a function $f(x,y)$, which is denoted by $\mathcal{R}f(s,\vec{\theta})$, can be written as

$$\Re f(s,\vec{\theta}) = \int_{\boldsymbol{x}\cdot\vec{\theta}=s} f(\boldsymbol{x})d\boldsymbol{x} = \int_{-\infty}^{\infty} f(x(z),y(z))dz, \tag{3.7}$$

where $x(z) = z\sin\theta + s\cos\theta$ and $y(z) = -z\cos\theta + s\sin\theta$. Now again take the Radon transform, but with $-s$ and $-\vec{\theta}$,

$$\Re f(-s,-\vec{\theta}) = \int_{\boldsymbol{x}\cdot-\vec{\theta}=-s} f(\boldsymbol{x})d\boldsymbol{x} = \int_{-\infty}^{\infty} f(\bar{x}(z),\bar{y}(z))dz, \tag{3.8}$$

where

$$\bar{x}(z) = z(-\sin\theta) + (-s)(-\cos\theta) = -z\sin\theta + s\cos\theta$$
$$\bar{y}(z) = -z(-\cos\theta) + (-s)(-\sin\theta) = z\cos\theta + s\sin\theta.$$

Note that

$$\bar{x}(-z) = -(-z)\sin\theta + s\cos\theta = z\sin\theta + s\cos\theta = x(z)$$
$$\bar{y}(-z) = (-z)\cos\theta + s\sin\theta = -z\cos\theta + s\sin\theta = y(z).$$

A substitution of $z = -u$ into equation (3.8), using the properties mentioned above, we get

$$\begin{aligned}
\Re f(-s,-\vec{\theta}) &= \int_{\boldsymbol{x}\cdot-\vec{\theta}-s} f(\boldsymbol{x})d\boldsymbol{x} \\
&= \int_{-\infty}^{\infty} f(\bar{x}(z),\bar{y}(z))dz \\
&= \int_{\infty}^{-\infty} f(\bar{x}(-u),\bar{y}(-u))(-1)du \\
&= \int_{-\infty}^{\infty} f(x(u),y(u))du \\
&= \Re f(s,\vec{\theta}),
\end{aligned}$$

as equation (3.7). Thus we conclude that

$$\Re f(s,\vec{\theta}) = \Re f(-s,-\vec{\theta}).$$

□

Lemma 3.5.4. *Denote the Radon transform of a function $f(x,y)$ in $\mathbb{R}^2$ as $\Re f(s,\vec{\theta})$, and the Fourier transform of this radon transform as $\widetilde{\Re f}(s,\vec{\theta})$. Then*

$$\widetilde{\Re f}(-s,-\vec{\theta}) = \widetilde{\Re f}(s,\vec{\theta}).$$

Proof. Using Lemma 3.5.3 and applying a change of variables with $u = -t$, and thus $du = -dt$, we get

$$\begin{aligned}
\widetilde{\Re f}(-s,-\vec{\theta}) &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \Re f(t,-\vec{\theta})e^{-it(-s)}dt \\
&= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \Re f(t,-\vec{\theta})e^{-i(-t)s}dt \\
&= \frac{1}{\sqrt{2\pi}}\int_{\infty}^{-\infty} \Re f(-u,-\vec{\theta})e^{-ius}(-1)du \\
&= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \Re f(-u,-\vec{\theta})e^{-ius}du \\
&= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \Re f(u,\vec{\theta})e^{-ius}du \\
&= \widetilde{\Re f}(s,\vec{\theta}).
\end{aligned}$$

□

Now we can prove Theorem 3.5.2.

Proof Theorem 3.5.2. By the Fourier inversion theorem, Theorem 3.4.1 we know that for $\boldsymbol{x} \in \mathbb{R}^2$

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} \hat{f}(\boldsymbol{\xi}) e^{i\boldsymbol{x}\cdot\boldsymbol{\xi}} d\boldsymbol{\xi},$$

where $\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$ and $\boldsymbol{\xi} = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}$. We can now do a change of variables, from $\boldsymbol{\xi}$ to polar coordinates s and θ, by using that $\boldsymbol{\xi} = (s\cos\theta, s\sin\theta)$. This gives

$$f(x,y) = \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\infty} \hat{f}(s\vec{\theta}) e^{is\boldsymbol{x}\cdot\vec{\theta}} s ds d\theta.$$

We can now recognize the term $\hat{f}(s\vec{\theta})$ from Theorem 3.5.1, the central slice theorem and can therefore replace the term by the 1D Fourier transform of the Radon transform of f, $\widetilde{\mathfrak{R}f}(s,\vec{\theta})$.

$$f(x,y) = \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\infty} \widetilde{\mathfrak{R}f}(s,\vec{\theta}) e^{is\boldsymbol{x}\cdot\vec{\theta}} s ds d\theta.$$

Although this equation is correct, in X-ray computed tomography we usually take $\theta \in [0,\pi]$, so we would also like to have this in our integral expression for f. We can split the integral into a part where we integrate θ from 0 to π, and into a part where we integrate θ from π to 2π. Note that $\sin(\theta - 2\pi) = \sin(\theta)$ and $\cos(\theta - 2\pi) = \cos(\theta)$, so we can replace the second part by an integral that integrates θ from $-\pi$ to 0. Now we can apply Lemma 3.5.4 and use the even property.

$$\begin{aligned}
f(x,y) &= \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\infty} \widetilde{\mathfrak{R}f}(s,\vec{\theta}) e^{is\boldsymbol{x}\cdot\vec{\theta}} s ds d\theta, \\
&= \frac{1}{2\pi} \int_0^{\pi} \int_0^{\infty} \widetilde{\mathfrak{R}f}(s,\vec{\theta}) e^{is\boldsymbol{x}\cdot\vec{\theta}} s ds d\theta + \frac{1}{2\pi} \int_{\pi}^{2\pi} \int_0^{\infty} \widetilde{\mathfrak{R}f}(s,\vec{\theta}) e^{is\boldsymbol{x}\cdot\vec{\theta}} s ds d\theta \\
&= \frac{1}{2\pi} \int_0^{\pi} \int_0^{\infty} \widetilde{\mathfrak{R}f}(s,\vec{\theta}) e^{is\boldsymbol{x}\cdot\vec{\theta}} s ds d\theta + \frac{1}{2\pi} \int_{-\pi}^{0} \int_0^{\infty} \widetilde{\mathfrak{R}f}(s,\vec{\theta}) e^{is\boldsymbol{x}\cdot\vec{\theta}} s ds d\theta \\
&= \frac{1}{2\pi} \int_0^{\pi} \int_0^{\infty} \widetilde{\mathfrak{R}f}(s,\vec{\theta}) e^{is\boldsymbol{x}\cdot\vec{\theta}} s ds d\theta + \frac{1}{2\pi} \int_0^{\pi} \int_{-\infty}^{0} \widetilde{\mathfrak{R}f}(-rs,-\vec{\theta}) e^{-is\boldsymbol{x}\cdot-\vec{\theta}} (-s) ds d\theta \\
&= \frac{1}{2\pi} \int_0^{\pi} \int_0^{\infty} \widetilde{\mathfrak{R}f}(s,\vec{\theta}) e^{is\boldsymbol{x}\cdot\vec{\theta}} s ds d\theta + \frac{1}{2\pi} \int_0^{\pi} \int_{-\infty}^{0} \widetilde{\mathfrak{R}f}(s,\vec{\theta}) e^{is\boldsymbol{x}\cdot\vec{\theta}} (-s) ds d\theta \\
&= \frac{1}{2\pi} \int_0^{\pi} \int_{-\infty}^{\infty} \widetilde{\mathfrak{R}f}(s,\vec{\theta}) e^{is\boldsymbol{x}\cdot\vec{\theta}} |s| ds d\theta. \qquad (3.9)
\end{aligned}$$

□

From the above we have established a filter, which we will denote by

$$\mathfrak{G}\mathfrak{R}f(t,\vec{\theta}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \widetilde{\mathfrak{R}f}(s,\vec{\theta}) e^{ist} |s| ds. \qquad (3.10)$$

One can see that low frequencies are oppressed by the term $|s|$, while high frequencies are amplified. Since we integrate using polar coordinates, the information for s small, is measured more than information for s large. In the reconstruction method of backprojection in section 3.3 we add up all the measured values by means of an integral, so the middle of the reconstructed image is sharper than the outer edges of the reconstructed image. Equation (3.10) gives a filter that, by oppressing and amplifying certain frequencies, ensures that the reconstructed image is of the same sharpness throughout. Note that we consider perfect, noiseless data. In case of noisy data, the noise for high frequencies will also be amplified.

3.6. Filtered Backprojection

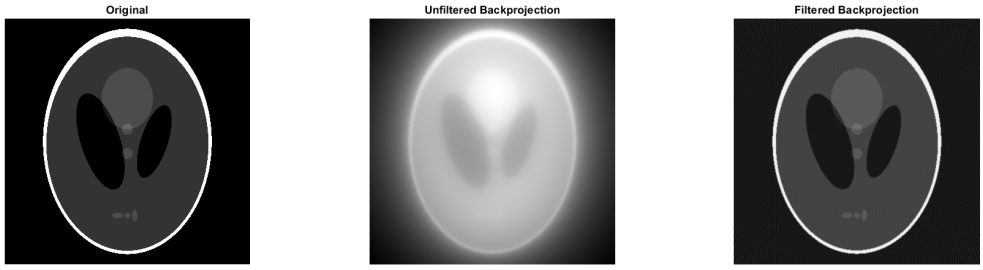
Equation (3.9) gives us a filtered reconstruction of a function $f(x, y)$. If we now apply this to the linear attenuation coefficient $\mu(x, y)$ we get the filtered backprojection summation image

$$\mu_{fb}(x, y) = \frac{1}{\sqrt{2\pi}} \int_0^{\pi} \int_{-\infty}^{\infty} \widetilde{\mathfrak{R}\mu}(s, \vec{\theta}) e^{is\boldsymbol{x}\cdot\vec{\theta}} |s| ds d\theta. \tag{3.11}$$

From equation (3.4), we know that $\mathfrak{R}\mu(s, \vec{\theta})$ is an expression for our measurement result $\log \frac{I_0(s,\vec{\theta})}{I_d(s,\vec{\theta})}$. Replacing in equation (3.11) gives the filtered backprojection summation image in terms of the measurement:

$$\mu_{fb}(x, y) = \frac{1}{\sqrt{2\pi}} \int_0^{\pi} \int_{-\infty}^{\infty} \widetilde{\log}\Big(\frac{I_0(s, \vec{\theta})}{I_d(s, \vec{\theta})}\Big) e^{is\boldsymbol{x}\cdot\vec{\theta}} |s| ds d\theta.$$

In figure 3.7 one can see the difference in the unfiltered backprojection summation image compared to the filtered one. The filtered backprojection gives a perfect reconstruction, as the data contains no noise.



(a) Original Shepp-Logan phantom. (b) Unfiltered backprojection. (c) Filtered backprojection.

Figure 3.7: The original Shepp-Logan phantom compared to the unfiltered and filtered backprojection summation images.

4

X-ray Tomography as a Discrete Linear Inverse Problem

4.1. Introduction

In this chapter I will derive X-ray tomography as a discrete linear inverse problem where the data is assumed to contain noise. The problem will be derived by introducing linear inverse problems in section 4.2 and ill-posedness of those problems in section 4.3. Next, the general method of naive inversion is considered, together with the downsides of this method. In section 4.5 singular value decomposition is explained, as this is needed in section 4.6, where the least-squares solution, commonly used for naive reconstruction, is derived. Finally, section 4.7 covers inverse crime that can occur when solving linear inverse problems with simulated data. Solutions to X-ray tomography with noisy data as a linear inverse problem will be covered in chapter 5.

4.2. Linear Inverse Problems

An inverse problem is always related to a direct problem. In a direct problem we have a given cause and are want to predict the effect. In an inverse problem we know the effect and the event, but are interested in predicting the cause.

As an example, suppose we take a photograph of a building. In this direct problem, the building itself is the cause, which we call μ. The taking a picture is the event, which is given by A. The resulting image of the building is the effect, m. If we know all the details about the building, and know the mechanism with which the camera takes a picture, we can determine what the image will look like. That is, if we have A and μ, we can calculate m by $m = A\mu$. Now suppose that we take the picture, but the lens was not completely focused. Then the image will be out of focus and thus blurry. Then the event of taking a picture is given by a new $\tilde{A}$. If we know how unfocused the lens was, we can determine the blurriness of the image, without having to see the blurry image. We known the cause, the building, and know the event, the unfocused lens taking a picture, and can thus determine the effect, which is the blurry image. We know $\tilde{A}$ and μ, and can thus calculate m, by $m = \tilde{A}\mu$. We will now consider the inverse problem corresponding to the direct problem of the unfocused image of the building. If we know of the event, the unfocused lens taking a picture, and know the effect, the blurry image of a building, can we determine the cause, which is the building? So if we have m and $\tilde{A}$, can we recover μ by $m = \tilde{A}\mu$?

X-ray computed tomography can also be viewed as a linear inverse problem. In this case we have an event, the CT scan which is modelled by A, that transforms a cause, for example the brain of a patient μ, into an effect, the CT image of the brain m. The direct problem is

> Given the details of a patient's brain and the CT machine, produce measurements resulting in the CT image of the brain.

We know all about the cause and the event, and can thus theoretically determine the effect. Now consider the inverse problem:

Given the CT machine and the measurements, reconstruct the patient's brain.

As this inverse problem results in a minimally-invasive way to look into a patient's brain to determine possible injuries of illness, it is a very important inverse problem to consider.

We will now consider a mathematical expression of the direct and inverse problem for X-ray tomography. I will not consider the continuous case, but will directly work with the discrete case, since realistic X-ray computed tomography can only be done in the discrete case.

Cause: Attenuation coefficient First we want to find a mathematical expression for the cause. In the case of X-ray tomography this is what we want to measure, for example the tissue density in a patient's brain. A patient's brain is a 3-dimensional object, but we will consider only 2D slices. We say that this tissue density is related to the attenuation coefficient $\mu(x, y)$, a function in $\mathbb{R}^2$. As said before, we cannot realistically reconstruct $\mu(x, y)$ as a continuous function, due to limitation of the CT scanning machine and the computational reconstruction. Therefore, we need to determine a grid for the relevant domain of $\mu(x, y)$. We need to divide this area into pixels and as an effect the value of $\mu(x, y)$ is constant within that pixel. The more pixels one chooses, the more details the reconstruction can show. The detail in the image also depends on the number of measurements taken. The more pixels one chooses, the more computing power is needed to make the reconstruction. We have to split the relevant area within the domain of $\mu(x, y)$ in $\mathbb{R}^2$ into pixels. We will divide the x-axis into b_x intervals, and do the same for the y-axis, dividing it into b_y intervals. It is convenient to make the grid symmetric, thus taking $b_x = b_y$. For convenience, the grids used in this report are symmetric. The total number of pixel is given by $B = b_x \times b_y$. As a result of this discretization the attenuation coefficient $\mu(x, y)$ is constant within a pixel, and we denote this value as μ_b, where $1 \leq b \leq B$.

Thus the cause in our X-ray tomography problem is given by $\boldsymbol{\mu}$ [1], which is a vector in $\mathbb{R}^B$, where the B is determined by the chosen size of the grid. The entries of the vector $\boldsymbol{\mu}$ are μ_b, the constant value of the attenuation coefficient within pixel b.

Effect: Measurements Next we need to mathematically express the effect, which is the measurements as a result of the CT scan. In the case of continuous X-ray tomography we measure by sending X-ray beams through the patient's head and measure the difference in intensity. We do this for beams with different angles θ and radii s. The discretization of the angles and radii is determined by the CT machine taking the measurements. Let θ be sampled with equidistant steps over the half circle:

$$\theta_j = \frac{j-1}{J}\pi, \text{ where } 1 \leq j \leq J.$$

J is the total number of steps. Let s be sampled with with equidistant steps over an interval from $-S$ to S:

$$s_\nu = -S + 2S\frac{\nu - 1}{N}, \text{ where } 1 \leq \nu \leq N.$$

N is the total number of steps. If we measure from J different angles, and for each angle measure from N different distances to the origin, there are a total of $K = J \times N$ measurements taken. Since we take K measurements, the resulting mathematical expression is a vector of K entries, one for each measurement.

Thus the effect in our X-ray tomography is given by $\boldsymbol{m}$, which is a vector in $\mathbb{R}^K$, where K is determined by the chosen number of measurements.

[1] The bold notation $\boldsymbol{\mu}$ refers to the discretized version of the continuous attenuation coefficient, denoted with a non-bold $\mu(x, y)$.

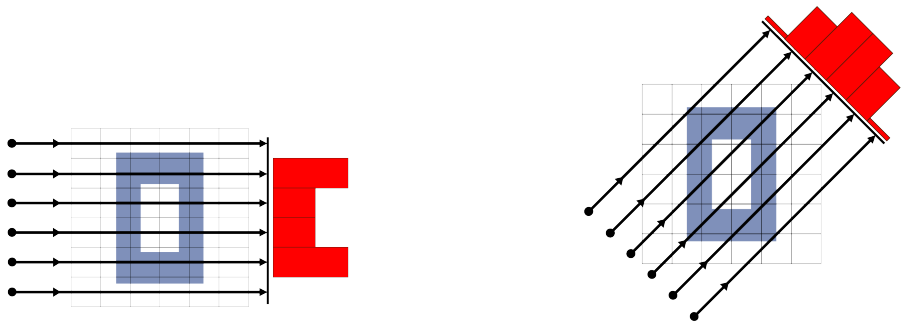


Figure 4.1: Two discrete measurements of a rectangle with a hole in the middle.

Event: CT scan Next we need to mathematically express the event, which is the modelling of the CT machine taking the measurements. As described before, the event will transform the object into a measurement. Our object is the continuous attenuation coefficient $\mu(x, y)$. The mapping A is also continuous, but naturally discrete due to the measurement device. In order to compute reconstructions using a computer, A, just as $\mu(x, y)$, has to be further discretised. The measurement is given by the $K \times 1$ vector $\boldsymbol{m}$. If we want to find a mathematical expression that transforms $\boldsymbol{\mu}$ into $\boldsymbol{m}$, we need a $K \times B$ matrix A. Each row of this matrix will give the values with which the attenuation coefficient in each pixel is transformed. The sum of all these transformations will result in the measurement. But how do we determine the entries of this matrix A? In the continuous case, we determined the log difference in intensity by summing over the attenuation coefficient times an infinitesimally small distance, resulting in the integral. In the discrete case, we can do the same, but will not let the distance become very small. We thus consider the attenuation coefficient times the distance, summing over all distance intervals, the pixels in this case. So, for one measurement, the entries of the row of the matrix A are given by the distance that beam travels through a pixel. If that row is multiplied by the values in $\boldsymbol{\mu}$, we get the sum of the distance in each pixel, times the attenuation coefficient in that pixel, which is exactly what we wanted. Mathematically this is given by

$$m_{s_\nu,\theta_j} = \sum_{b=1}^{B} a_{b,(\theta_j,s_\nu)}\mu_b.$$

Thus the event in our X-ray tomography case is given by the $K \times B$ matrix A, where the entries are determined by the distance an X-ray beam with angle θ_j and radius s_ν travels through pixel b.

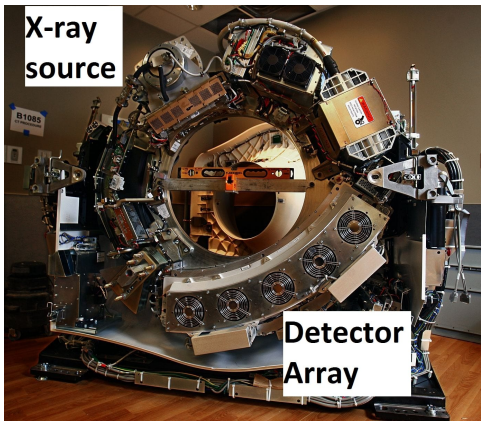


Figure 4.2: The inside of a tomography machine, `https://oncologymedicalphysics.com/ct-design-and-operation/`.

Error For each measurement there is an error. This can either be caused by the calibration of the machine, or physical events such as Compton scatter. Since we have an error term for each measure-

ment, the error vector containing the individual errors will be a vector in $\mathbb{R}^K$.

X-ray tomography in matrix form Combining the results from above we get the following discrete mathematical form of the problem of X-ray tomography.

$$\boldsymbol{m} = A\boldsymbol{\mu} + \boldsymbol{\epsilon}, \tag{4.1}$$

or in complete matrix form

$$\begin{bmatrix} m_{1,1} \\ m_{1,2} \\ m_{1,3} \\ \vdots \\ m_{J,1} \\ \vdots \\ m_{J,N-1} \\ m_{J,N} \end{bmatrix} = \begin{bmatrix} a_{1,(1,1)} & a_{2,(1,1)} & a_{3,(1,1)} & \cdots & a_{B-1,(1,1)} & a_{B,(1,1)} \\ a_{1,(1,2)} & a_{2,(1,2)} & a_{3,(1,2)} & \cdots & a_{B-1,(1,2)} & a_{B,(1,2)} \\ a_{1,(1,3)} & a_{2,(1,3)} & a_{3,(1,3)} & \cdots & a_{B-1,(1,3)} & a_{B,(1,3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{1,(J,1)} & a_{2,(J,1)} & a_{3,(J,1)} & \cdots & a_{B-1,(J,1)} & a_{B,(J,1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{1,(J,N-1)} & a_{2,(J,N-1)} & a_{3,(J,N-1)} & \cdots & a_{B-1,(J,N-1)} & a_{B,(J,N-1)} \\ a_{1,(J,N)} & a_{2,(J,N)} & a_{3,(J,N)} & \cdots & a_{B-1,(J,N)} & a_{B,(J,N)} \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \vdots \\ \mu_{B-1} \\ \mu_B \end{bmatrix} + \begin{bmatrix} \epsilon_{1,1} \\ \epsilon_{1,2} \\ \epsilon_{1,3} \\ \vdots \\ \epsilon_{J,1} \\ \vdots \\ \epsilon_{J,N-1} \\ \epsilon_{J,N} \end{bmatrix}.$$

Remember that the inverse problem of X-ray tomography is given by

> Given the CT machine and the measurements, reconstruct the patient's brain.

Now that we have determined a mathematical expression for our problem, given by equation (4.1), we can mathematically write down the inverse problem as:

> Given $\boldsymbol{m}$ by $\boldsymbol{m} = A\boldsymbol{\mu} + \boldsymbol{\epsilon}$, determine $\boldsymbol{\mu}$.

The MATLAB code by Mueller and Siltanen [7] can be used to compute the matrix A. The number of pixels B can be chosen, and the number of measurement angles is set to $\sqrt{B}$. The number of lines per measurement, the radii, is determined by the radon function of MATLAB and cannot be chosen. In figure 4.3 one can see a visual representations of the computed matrices A for 4, 8, 16 and 32 pixels. In the captions the sizes of the matrices are given. One can see that even for a relatively small pixel grid of 32×32 the matrix becomes very large. For 64×64 or more pixels the computation of A becomes computationally demanding.

4.3. Well-posed and ill-posed problems

Remember our derived model for the linear discrete inverse problem of X-ray tomography, where we had

$$\boldsymbol{m} = A\boldsymbol{\mu} + \boldsymbol{\epsilon}.$$

- $\boldsymbol{m} \in \mathbb{R}^K$ is the vector of measurements. K is the number of measurements.
- $\boldsymbol{\mu} \in \mathbb{R}^B$ is the vector of constant attenuation coefficients in pixels of our measurement area. B is the number of pixels, determined by $b_x \times b_y$.
- $A \in \mathbb{R}^{K \times B}$ is the matrix that transforms the attenuation coefficients into measurements.
- $\boldsymbol{\epsilon} \in \mathbb{R}^K$ is the vector containing the error per measurement.

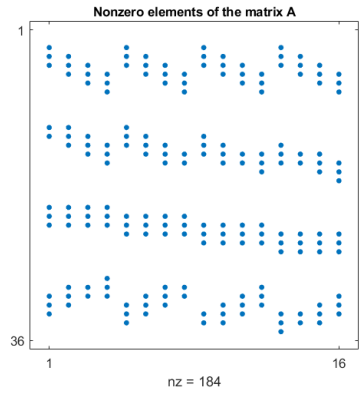
Before we talk about well-posed and ill-posed problems, we need to write down some more information about the matrix A. It is a linear operator that maps from its domain, $\mathcal{D}(A)$ in the model space $\mathbb{R}^B$, to the image $A(\mathcal{D}(A))$ in the data space $\mathbb{R}^K$. A visual representation can be seen in image.

$$A : \mathcal{D}(A) \subset \mathbb{R}^B \to A(\mathcal{D}(A)) \subset \mathbb{R}^K$$

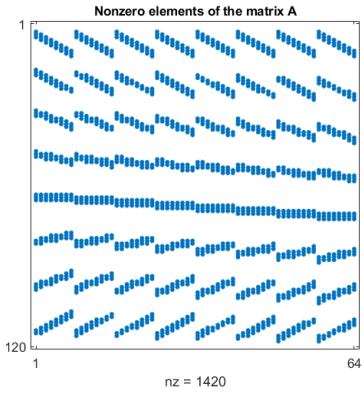
Although the error term is random, it might be possible to estimate an upper bound, so

$$\|\epsilon\|_{\mathbb{R}^K} \leq \delta,$$

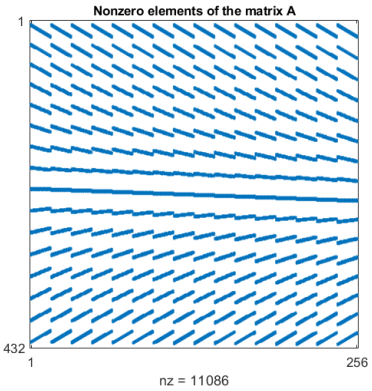
where δ is the upper bound. This upper bound can be based on conclusions from calibration tests of the CT machine. When working with real-life data it is best to not to make any assumptions on the



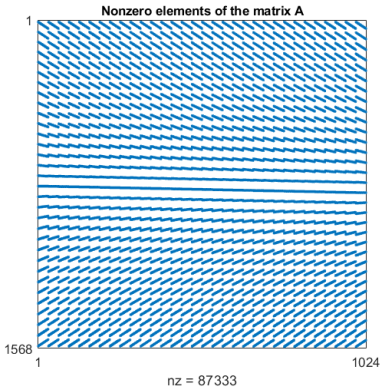
(a) $4 \times 4 = 16$ pixels and $6 \times 9 = 36$ measurements.



(b) $8 \times 8 = 64$ pixels and $8 \times 15 = 120$ measurements.



(c) $16 \times 16 = 256$ pixels and $16 \times 27 = 432$ measurements.



(d) $32 \times 32 = 1024$ pixels and $32 \times 49 = 1568$ measurements.

Figure 4.3: The nonzero elements of matrix A for different numbers of pixels (columns) and measurements (rows). 'Nz' indicates the number of nonzero elements.

level of noise incorporated in the measurement without basing them on tests. In this section, the upper bound for the error is used to mathematically and visually explain that noise can cause ill-posedness, particularly instability, regardless of the size of the noise.

Due to the error and the upper bound for the error, we know that the measurement vector $\boldsymbol{m}$ lies within $B(A\boldsymbol{\mu})_\delta$, the sphere or circle in $\mathbb{R}^K$ with center point $A\boldsymbol{\mu}$ and radius δ, as can be seen in figure 4.4. If we consider this argument the other way around, we know that the real value $A\boldsymbol{\mu}$ lies within $B(\boldsymbol{m})_\delta$, the sphere or circle in $\mathbb{R}^K$ with center point $\boldsymbol{m}$ and radius δ, since the maximum distance between $\boldsymbol{m}$ and $A\boldsymbol{\mu}$ is at most δ.

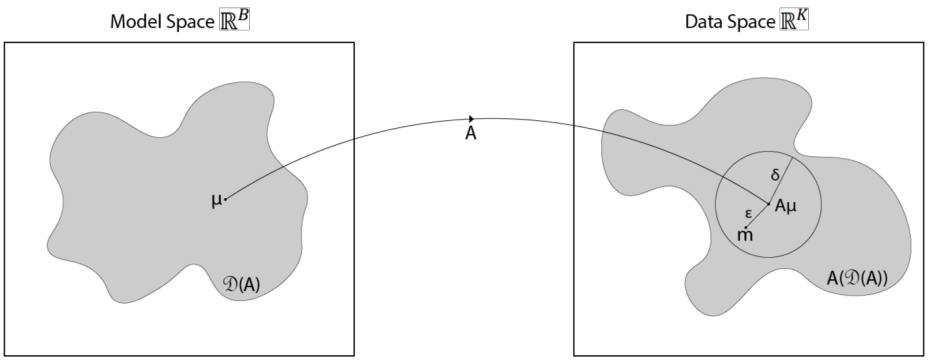


Figure 4.4: Visual representation of the linear operator A.

Now that we know more about the matrix A and the error, we can define when a problem is well-posed or ill-posed.

Definition 4.3.1 (Well-posed)**.** *A solution method is called well-posed if the following three conditions are satisfied:*

- ***H1** Existence: There should be at least one solution.*
- ***H2** Uniqueness: There should be at most one solution.*
- ***H3** Stability: The solution must depend continuously on data.*

If one of these conditions is not satisfied, the problem or solution method is called ill-posed.

In practice, equation (4.1) is ill-posed, mainly because of instability of A. Note that even though A^{-1} can exist, it could be that the eigenvalues are so small, that a small addition of noise can lead to a totally different reconstruction. This effect can be seen in figure 4.7. As this is undesirable for a reconstruction, it is important to study ill-posedness closely.

I will show how discrete X-ray tomography can fail to fulfill the conditions in Definition 4.3.1.

H1: A solution does not exist. By definition, $A\boldsymbol{\mu} \in A(\mathcal{D}(A))$, since $\boldsymbol{\mu} \in \mathcal{D}(A)$, but it could be that

$$\boldsymbol{m} = A\boldsymbol{\mu} + \boldsymbol{\epsilon} \notin A(\mathcal{D}(A)).$$

This means that there is no solution to the inverse problem. $A^{-1}\boldsymbol{m}$ is not defined, since $\boldsymbol{m} \notin A(\mathcal{D}(A))$, see figure 4.5a. Note that if an element in $\mathbb{R}^K$ is not in Range(A), it is in Coker(A). Thus when checking whether a solution exists, one can check whether Coker$(A) = A(\mathcal{D}(A))$ is empty.

For X-ray tomography this means that the reconstruction will contain empty spots for the parts where $A^{-1}\boldsymbol{m}$ is not defined.

H2: There is no unique solution. This condition fails if there are two elements $\boldsymbol{\mu}, \boldsymbol{\lambda} \in \mathcal{D}(A)$ such that

$$A\boldsymbol{\mu} = A\boldsymbol{\lambda},$$

which means that the solution is not unique. This can occur when there exists a $\boldsymbol{\mu}_0 \in$ Ker(A) such that $\boldsymbol{\mu}_0 \neq \mathbf{0}$. Then for $\boldsymbol{m} \in$ Range(A) we have $A(A^{-1}(\boldsymbol{m})) = \boldsymbol{m} = A(A^{-1}(\boldsymbol{m}) + \boldsymbol{\mu}_0)$. Both $A^{-1}(\boldsymbol{m}) \in \mathbb{R}^B$ and $A^{-1}(\boldsymbol{m}) + \boldsymbol{\mu}_0 \in \mathbb{R}^B$ give the same value $\boldsymbol{m} \in \mathbb{R}^K$, so the solution is not unique. See figure 4.5b. Thus when checking whether the solution is unique, one can check whether Ker(A) only contains $\mathbf{0}$.

For X-ray tomography this means that if we want to reconstruct from the measurement, the chosen values could be off (by $\boldsymbol{\mu}_0$), which makes the reconstruction less accurate for those parts.

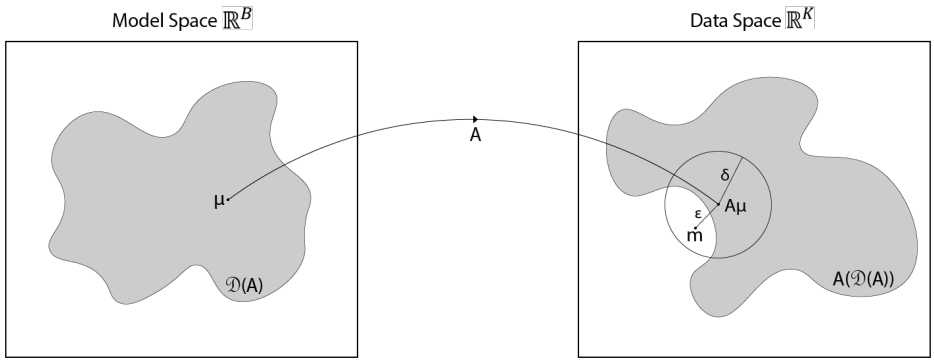
H3: The solution is not stable. A solution is stable when the solution depends on the data continuously. Simply said, two points that lie close in $\mathbb{R}^K$, must also lie close in $\mathbb{R}^B$ and thus a small perturbation in $\mathbb{R}^K$ should not result in a drastically different reconstruction in $\mathbb{R}^B$. See figure 4.5c. A problem can be unstable when the matrix A has small eigenvalues. Then even a small amount of noise added to the measurement can lead to a totally different reconstruction, which is undesirable. A tool to see how big the difference between the eigenvalues is, is called the condition number.

Definition 4.3.2 (Condition number)**.** *The condition number of an invertible matrix A is given by*

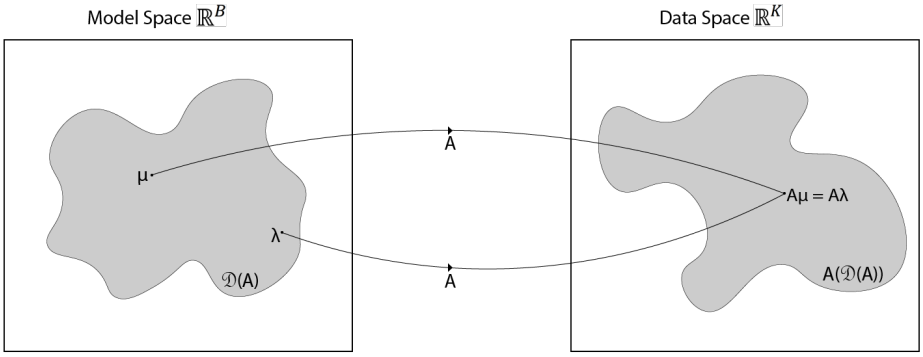
$$\mathit{Cond}(A) := \frac{d_1}{d_K},$$

where d_1 is the largest singular value and $d_K > 0$ is the smallest singular value.

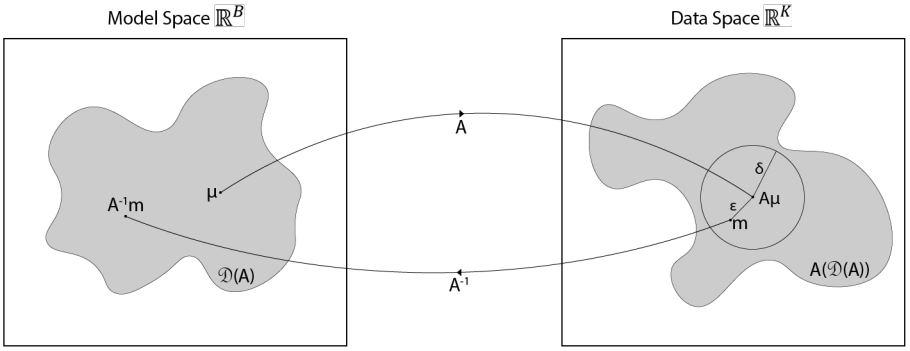
If there is a large difference in size of the singular values, Cond(A) will be very big. So using the condition number, we can make an estimate on the stability of the matrix A. Note that when a matrix is not invertible and thus has a singular value zero, then the condition number is not defined.



(a) Visual representation of a failure of H1, there is no solution.



(b) Visual representation of failure of H2, there is no unique solution.



(c) Visual representation of failure of H3, the solution is unstable.

Figure 4.5: Visual representation of ill-posed solution methods

4.4. Naive reconstruction

We will now consider a method to solve this inverse problem, called naive inversion. As the name might suggest, this method might seem intuitive at first, but comes with some big problems. I will first explain the solution method and will then state some of its relevant problems. I will do this only considering the X-ray tomography case.

If A is an invertible matrix, we can recover $\boldsymbol{\mu}$ in the following way, called naive reconstruction:

$$\begin{aligned} \boldsymbol{m} &= A\boldsymbol{\mu} + \boldsymbol{\epsilon}, \\ A^{-1}\boldsymbol{m} &= A^{-1}A\boldsymbol{\mu} + A^{-1}\boldsymbol{\epsilon}, \\ A^{-1}\boldsymbol{m} &= \boldsymbol{\mu} + A^{-1}\boldsymbol{\epsilon}, \\ \boldsymbol{\mu} &= A^{-1}\boldsymbol{m} - A^{-1}\boldsymbol{\epsilon}. \end{aligned} \tag{4.2}$$

When A is not invertible, one would look for the least-squares solution, which is obtained by minimising $\|A\boldsymbol{\mu} - \boldsymbol{m}\|^2$. More on the least-squares solution can be read in section 4.6

Error In equation (4.2) one can see that we not only apply the inverse A to the measurement, but also the vector containing all errors. It could be that $\left\|A^{-1}\right\|$ is very large, so even though $\|\boldsymbol{\epsilon}\|$ might not be large, the total term $\left\|A^{-1}\boldsymbol{\epsilon}\right\|$ can be large. Due to this instability, the reconstruction might not be accurate.

Inverse In the naive reconstruction we are taking the inverse of the matrix A. Of course, this inverse does not always exist. Remember that we can only take the inverse of a square matrix, so we can only take the inverse when the number of pixels of the measurement area is the same as the number of measurements we are doing. Also, if the rows or columns of A are not linearly dependent or if 0 is an eigenvalue, the inverse does not exist. As was already stated above, even when the inverse of A exists, it can be very unstable due to very small eigenvalues. A little bit of noise can already result in a totally different reconstruction.

4.5. Singular Value Decomposition

Before considering the general method used for naive inversion, called the minimum norm least-squares solution, we need to take a closer look at the unstable matrix A. We will consider a well-known method of extracting information from a matrix, called singular value decomposition. Using this method we can draw some more conclusions on the ill-posedness of the matrix. The definitions and theorems used in this section are based on the text by Yanai, Takeuchi and Takane [8].

Consider the following matrices

$$\begin{aligned} U_{[r]} &= [\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_r], \boldsymbol{u}_1, \dots, \boldsymbol{u}_r \in \mathbb{R}^K, \\ V_{[r]} &= [\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_r], \boldsymbol{v}_1, \dots, \boldsymbol{v}_r \in \mathbb{R}^B, \\ \Delta_{[r]} &= \begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_r \end{bmatrix}. \end{aligned}$$

Theorem 4.5.1 (Matrix Decomposition)**.** *A $K \times B$ matrix A of rank r can be decomposed as*

$$\begin{aligned} A &= d_1\boldsymbol{u}_1\boldsymbol{v}_1^T + d_2\boldsymbol{u}_2\boldsymbol{v}_2^T + \cdots + d_r\boldsymbol{u}_r\boldsymbol{v}_r^T, \\ &= U_{[r]}\Delta_{[r]}V_{[r]}^T, \end{aligned}$$

where $d_j, j = 1, \dots, r$ are nonzero singular values of A^TA.

Definition 4.5.1 (Compact Singular Value Decomposition)**.** *The matrix decomposition (as given in theorem 4.5.1), is called the compact singular value decomposition of the matrix A, where d_j indicates the j-th largest singular value of A.*

Now let $U_{[0]}$ be a $B \times (B-r)$ columnwise orthogonal matrix, that is also orthogonal to $U_{[r]}$. Similarly, let $V_{[0]}$ be a $K \times (K-r)$ columnwise orthogonal matrix, that is also orthogonal to $V_{[r]}$.

Definition 4.5.2 (Complete Singular Value Decomposition)**.** *A complete form of singular value decomposition of the $K \times B$ matrix A is expressed as*

$$A = UDV^T, \tag{4.3}$$

where we define

$$U = [U_{[r]}, U_{[0]}], \tag{4.4}$$

$$V = [V_{[r]}, V_{[0]}], \tag{4.5}$$

$$D = \begin{bmatrix} \Delta_{[r]} & 0 \\ 0 & 0 \end{bmatrix}, \tag{4.6}$$

using $U_{[r]}$, $V_{[r]}$ and $\Delta_{[r]}$ as defined in Definition 4.5.1. U and V are both fully orthogonal, giving $U^TU = UU^T = I_B$ and $V^TV = VV^T = I_K$.

To understand SVD more intuitively, consider the following. The columns of both U and V, as well as the singular values in D, are arranged in a hierarchical manner, indicating that $\boldsymbol{u}_1$, $\boldsymbol{v}_1$ and d_1 are somehow more important than $\boldsymbol{u}_2$, $\boldsymbol{v}_2$ and d_2. The SVD can be seen as a way to decompose a matrix and arrange the terms in a such way that the first term explains most of the variation seen in the rows/-columns of A. Thus if we want to express A in one term, the term $d_1\boldsymbol{u}_1\boldsymbol{v}_1^T$ gives the best approximation out of all combinations. The term $d_2\boldsymbol{u}_2\boldsymbol{v}_2^T$ is the second best approximation, and so on.

Using singular value decomposition we can also determine that a matrix with a singular value equal to zero is not invertible. Let A be a matrix of arbitrary size. Then Definition 4.5.2 says that we can write $A = UDV^T$, where U, D, V are as defined in Definition 4.5.2. If we take the inverse of A, we should also take the inverse of UDV^T. Using that U and V are fully orthogonal we get

$$\begin{aligned} A^{-1} &= (UDV^T)^{-1} \\ &= (V^T)^{-1}D^{-1}U^{-1} \\ &= VD^{-1}U^T, \end{aligned}$$

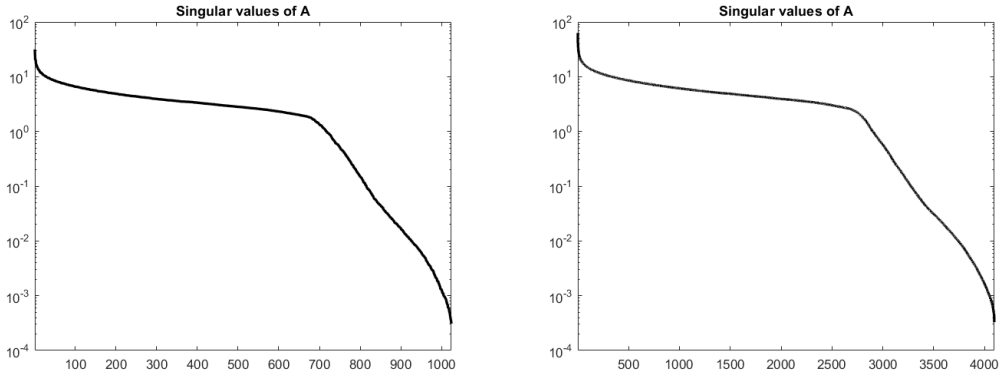
where

$$D^{-1} = \text{diag}\left(\frac{1}{d_1}, \dots, \frac{1}{d_K}\right),$$

with d_1 the biggest singular value and d_K the smallest singular value. Since zero is a singular value, there is at least one $d_i = 0, 1 \leq i \leq K$, with $\frac{1}{d_i}$ not defined. Thus we conclude that D^{-1}, and consequently also A^{-1}, does not exist.

The MATLAB code by Mueller and Siltanen [7] can be used to compute the SVD of a matrix A. It also plots the diagonal of D, containing the order singular values of A.

In figure 4.6 one can see a logarithmic plot of the singular values of the matrix A for different numbers of pixels. Note that each of the logarithmic plots has a singular value after which the magnitude of the following singular values decrease more rapidly. This principle can be used in truncated singular value decomposition regularization in section 5.4.



(a) Singular values for 32×32 pixels.

(b) Singular values for 64×64 pixels.

Figure 4.6: A logarithmic plot of the singular values of the matrix A, for different numbers of pixels.

4.6. Minimum norm solution and pseudoinverse

If A is not (known to be) invertible, one can look for the minimum norm least-squares solution as naive reconstruction. The definitions and theorem in this section are based on chapter 4 of the book by Mueller and Siltanen [2].

Definition 4.6.1 (Minimum norm solution)**.** *A vector $\mathcal{L}(\boldsymbol{m}) \in \mathbb{R}^B$ is called a least-squares solution of the equation $A\boldsymbol{\mu} = \boldsymbol{m}$, where $\boldsymbol{\mu} \in \mathbb{R}^B$ and $\boldsymbol{m} \in \mathbb{R}^K$, if*

$$\left\|A\mathcal{L}(\boldsymbol{m}) - \boldsymbol{m}\right\| = \min_{\boldsymbol{z}\in\mathbb{R}^B} \|A\boldsymbol{z} - \boldsymbol{m}\|. \tag{4.7}$$

Furthermore, $\mathcal{L}(\boldsymbol{m})$ is called the minimum norm solution if

$$\left\|\mathcal{L}(\boldsymbol{m})\right\| = \inf\{\|\boldsymbol{z}\| : \boldsymbol{z} \text{ is a least squares solution of } A\boldsymbol{\mu} = \boldsymbol{m}\}. \tag{4.8}$$

In equation (4.7) we are looking for a alternative solution, $\mathcal{L}(\boldsymbol{m})$ as an approximation of $\boldsymbol{\mu}$, such that the distance between $A\mathcal{L}(\boldsymbol{m})$, the approximation of $\boldsymbol{m}$, and the real known $\boldsymbol{m}$, is as small as possible. Since we are minimizing over the distance, there could be multiple solutions for $\mathcal{L}(\boldsymbol{m})$, that are different, but have the same distance to $\boldsymbol{m}$. This can occur when the column and/or rows of A do not span $\mathbb{R}^B$ and/or $\mathbb{R}^B$ respectively. Therefore we need equation (4.8). Of all options given by equation (4.7), it takes the one that has the shortest length. This ensures that the $\mathcal{L}(\boldsymbol{m})$ is unique, satisfying well-posedness condition H2.

Definition 4.6.2 (Pseudoinverse)**.** *Let A be a $K \times B$ matrix and denote by $A = UDV^T$ the singular value decomposition of A. Let r be the largest index for which the corresponding singular value is nonzero: $r = \max_{1\le j\le \min(K,B)}\{j | d_j > 0\}$. Then the matrix $A^+ = VD^+U^T$ is called the pseudoinverse of A, with*

$$D^+ = \mathit{diag}\left(\frac{1}{d_1}, \frac{1}{d_2}, \dots, \frac{1}{d_r}, 0, \dots, 0\right) = \begin{bmatrix} \frac{1}{d_1} & 0 & \cdots & & & \cdots & 0 \\ 0 & \frac{1}{d_2} & & & & & \vdots \\ \vdots & & \ddots & & & & \\ & & & \frac{1}{d_r} & & & \\ & & & & 0 & & \\ \vdots & & & & & \ddots & \vdots \\ 0 & \cdots & & & & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{B\times K}.$$

If zero is a singular value of the matrix A, then the matrix is not invertible, but the pseudoinverse does exist. By only considering the nonzero singular values, $\frac{1}{d_j}$ is always defined, ensuring that the pseudoinverse always exists.

We can now relate the minimum norm least-squares solution and the pseudoinverse via the following theorem.

Theorem 4.6.1. *Let A be a $K \times B$ matrix and denote by $A = UDV^T$ the singular value decomposition of A. The minimum norm solution of the equation $A\boldsymbol{\mu} = \boldsymbol{m}$ is given by $A^+\boldsymbol{m}$, where A^+ is the pseudoinverse of A.*

Proof. In Definition 4.5.2 the matrix V was defined to be fully orthogonal, so its column vectors form an orthonormal basis for $\mathbb{R}^B$. This means that any vector in $\mathbb{R}^B$ can be written as a linear combination of the column vectors of V, thus

$$\boldsymbol{f} = a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + \cdots + a_B\boldsymbol{v}_B = \sum_{j=1}^{B} a_j\boldsymbol{v_j} = V\boldsymbol{a}.$$

We want to find $\boldsymbol{a}$ such that $\boldsymbol{f}$ becomes the minimum norm solution. It will follow that $\boldsymbol{a} = D^+U^T$.
To find the least-squares solution $\boldsymbol{f}$ we need to calculate $\|A\boldsymbol{f} - \boldsymbol{m}\|$. We will compute this minimum using the squared norm, as it gives the same minimum, but makes the computation easier. First we will use that $A = UDV^T$ and $\boldsymbol{f} = V\boldsymbol{a}$. We will also make use of the fact that U is a unitary orthogonal matrix, meaning that $I_K = UU^T$, and $\|U\boldsymbol{x}\| = \|\boldsymbol{x}\|$ for any matrix $\boldsymbol{x} \in \mathbb{R}^K$.

$$\begin{aligned} \|A\boldsymbol{f} - \boldsymbol{m}\|^2 &= \left\|(UDV^T)(V\boldsymbol{a}) - (UU^T)\boldsymbol{m}\right\|^2 \\ &= \left\|UD\boldsymbol{a} - UU^T\boldsymbol{m}\right\|^2 \\ &= \left\|U(D\boldsymbol{a} - U^T\boldsymbol{m})\right\|^2 \\ &= \left\|D\boldsymbol{a} - U^T\boldsymbol{m}\right\|^2 \end{aligned}$$

Using the definition of the squared norm, we can now write out the norm, and get

$$\begin{aligned}\left\|D\boldsymbol{a}-U^T\boldsymbol{m}\right\|^2 &= \sum_{j=1}^{K}\left[(D\boldsymbol{a})_j-(U^T\boldsymbol{m})_j\right]^2,\\ &= \sum_{j=1}^{K}\left[d_j a_j-(U^T\boldsymbol{m})_j\right]^2,\\ &= \sum_{j=1}^{r}\left[d_j a_j-(U^T\boldsymbol{m})_j\right]^2+\sum_{j=r+1}^{K}\left[(U^T\boldsymbol{m})_j\right]^2,\end{aligned}$$

using that $d_j = 0$ for $j \geq r$. Since D, U^T and $\boldsymbol{m}$ are fixed, this expression is minimized by the choice of the a_j. The expression is minimized if for every $j \leq r$ we have $d_j a_j = (U^T\boldsymbol{m})_j$, thus

$$a_j = \frac{1}{d_j}(U^T\boldsymbol{m})_j, j \leq r.$$

From this we get the following least-squares solution $\boldsymbol{f}$

$$\boldsymbol{f} = V\boldsymbol{a} = V\begin{bmatrix}\frac{1}{d_1}(U^T\boldsymbol{m})_1\\ \frac{1}{d_2}(U^T\boldsymbol{m})_2\\ \vdots\\ \frac{1}{d_r}(U^T\boldsymbol{m})_r\\ a_{r+1}\\ \vdots\\ a_B\end{bmatrix}.$$

$\boldsymbol{f}$ is called the minimum norm solution when equation (4.8) holds. The smallest norm $\|\boldsymbol{f}\|$ is given by taking $a_j = 0$ for $j > r$. This means that the minimum norm solution is given by

$$\boldsymbol{f} = V\boldsymbol{a} = V\begin{bmatrix}\frac{1}{d_1}(U^T\boldsymbol{m})_1\\ \frac{1}{d_2}(U^T\boldsymbol{m})_2\\ \vdots\\ \frac{1}{d_r}(U^T\boldsymbol{m})_r\\ 0\\ \vdots\\ 0\end{bmatrix} = VD^+U^T\boldsymbol{m},$$

where we recognize the pseudoinverse A^+ of A.
Thus we conclude that the minimum norm least-squares solution is given by the pseudoinverse $A^+ = VD^+U^T$. □

Now that we have found the minimum norm least-squares solution for the problem $\boldsymbol{m} = A\boldsymbol{\mu}$, we can check whether it is well-posed.

Existence To check whether there always exists a solution using the pseudoinverse we take a look at Coker(A). Suppose that there exists $\boldsymbol{m}_A \in$ Coker(A). Then A^+ maps $\boldsymbol{m}_A$ to zero. Thus there always exists a solution.

Uniqueness As already mentioned before, by definition of the minimum norm solution, we always choose the least-squares solution of the shortest length, ensuring that the solution is always unique.

Stability The condition number $\text{Cond}(A^+) = \frac{d_1}{d_r} = \text{Cond}(A)$ does not change when using the pseudoinverse instead of the regular inverse of A. This means that, even when A is invertible and the solution exists and is unique, the reconstruction can still be quite unstable. This means that condition H3 is generally not met by using the pseudoinverse as the minimum norm least-squares solution.

Now that we know about the principle of naive inversion and that this reconstruction is usually computed using the minimum norm least-squares solution, we can produce such images and see the effect of the instability. In figure 4.7 one can see minimum norm least-squares reconstructions for the 32×32 Shepp-Logan phantom, compared to the original. Note that figure 4.7a is a perfect reconstruction, since no noise was added. In figure 4.7b noise was added to the simulated measurement before reconstruction. One can see that the Shepp-Logan phantom is not recognizable in this reconstruction. This is a result of the instability of the reconstruction method used.

4.7. Inverse Crime

Inverse crime refers to when the same model is used to generate, as well as to invert, synthetic data [9]. In practice, inverse crimes arise when [10]:

- The numerically produced simulated data is produced by the same model that is used to invert the data,
- The discretization in the numerical solution is the same as the one used in the inversion.

In simulated discrete X-ray tomography, inverse crime is often committed by using the same grid for simulating the Shepp-Logan phantom and reconstruction of the same Shepp-Logan phantom using naive reconstruction. A way to avoid this, is by using a different grid for the simulation than for the reconstruction.

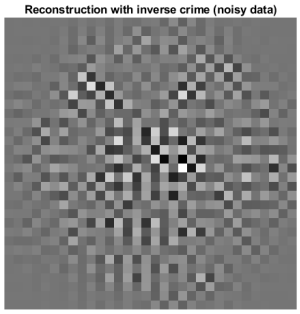
The MATLAB code by Mueller and Siltanen [7] can be used to used to reconstruct the Shepp-Logan phantom using least-squares for naive inversion. It produces a reconstruction committing inverse crime, with and without noisy data, and a reconstruction where inverse crime is avoided, also one with and one without noisy data. For the 32×32 Shepp-Logan phatom, these reconstructed images can be seen in figure 4.7.

Figure 4.7a is a very good reconstruction, as the relative error is 0%, but it is no realistic. It involves inverse crime and no noise in the data. This naive reconstruction method therefore does not work in reality.

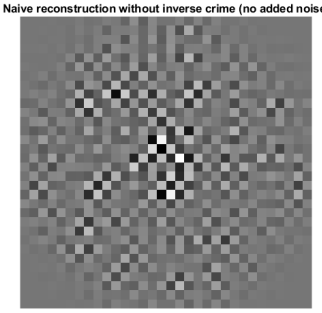
Figure 4.7b is the reconstruction where random noise was added to the data, but where inverse crime was still committed. The noisy data contains 0.1% noise. The noise was added by adding a vector of the same size of the noise-free measurement, containing random scalars from the normal distribution with mean 0 and standard deviation equal to the maximum absolute number of the noise-free measurement vector. The relative error is 860% and the reconstruction is therefore not accurate at all, as can also be concluded from looking at the image. This large relative error comes from the instability. As concluded before, adding a bit of noise to the measurement can produce a drastically different reconstruction, due to instability.

Figures 4.7c and 4.7d are constructed without inverse crime. They are therefore more realistic for real-life applications, but this comes with a larger relative error. Note that even without adding noise, the reconstruction is not accurate, emphasizing how important it is to avoid inverse crime when investigating reconstruction methods.

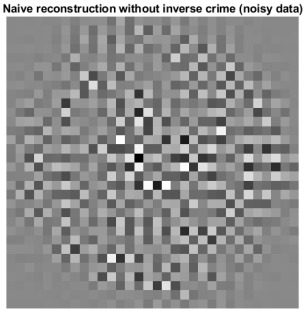
(a) Naive reconstruction with inverse crime and no noise added, relative error: 0%.



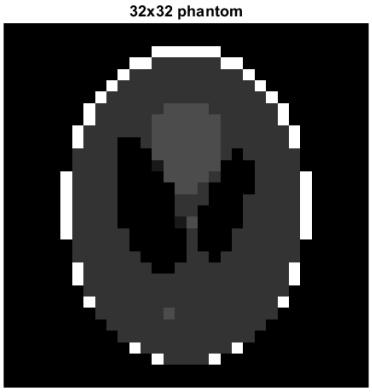
(b) Naive reconstruction with inverse crime and noisy data, relative error: 860%.



(c) Naive reconstruction without inverse crime and no noise added, relative error: 7030%.



(d) Naive reconstruction without inverse crime and noisy data, relative error: 9105%.



(e) Original 32×32 Shepp-Logan phantom.

Figure 4.7: Naive reconstruction of the Shepp-Logan phantom using least-squares, with and without inverse crime and with and without noisy data.

5

Regularization methods

5.1. Introduction

In this chapter I will consider some solution methods for X-ray tomography as a discrete linear inverse problem that can overcome the instability of the original problem, as concluded in chapter 4. First, section 5.2 will explain what such a regularization method consists of and what the general idea behind it is. In the following section 5.3 will describe how we can test the well-posedness, and thus stability, of such a regularization method, based on section 4.3. Sections 5.4 and 5.6 will briefly cover the regularization methods of truncated singular value decomposition and Tikhonov regularization, respectively. Sections 5.5 and 5.7 contain the reconstructions obtained after applying TSVD and Tikhonov regularization, respectively. Comments on the quality of the reconstruction are given. In the last section, section 5.8, I will compare the methods using MATLAB code.

5.2. General regularization methods

To overcome the instability of X-ray tomography with noisy data, regularization methods have to be used to obtain a stable approximate solution. This can be done by replacing the ill-posed X-ray tomography by a similar auxiliary well-posed problem. A regularization parameter $\alpha > 0$ controls the trade-off between the similarity to the original X-ray tomography problem, given by small values of α, and high stability of the auxiliary problem, given by large values of α [11].

This definition is based on the text by Chengg and Hofmann [11] and chapter 3 of the book by Mueller and Siltanen [2].

Definition 5.2.1 (Regularization method)**.** *Let $A : \mathbb{R}^B \to \mathbb{R}^K$ be an injective bounded linear operator. Consider the measurement $\boldsymbol{m} = A\boldsymbol{\mu} + \boldsymbol{\epsilon}$. A family of linear maps $\mathcal{R}_\alpha : \mathbb{R}^K \to \mathbb{R}^B$ parameterized by $0 < \alpha < \infty$ is called a regularization method if*

$$\lim_{\alpha \to 0} \mathcal{R}_\alpha A\boldsymbol{\mu} = \boldsymbol{\mu}, \tag{5.1}$$

for every $\boldsymbol{\mu} \in \mathbb{R}^B$.
Further, assume we are given a noise level $\delta > 0$ so that $\|\boldsymbol{m} - A\boldsymbol{\mu}\| \leq \delta$. A choice of regularization parameter $\alpha = \alpha(\delta)$ as a function of δ is admissible if

- $\alpha(\delta) \to 0$ *as* $\delta \to 0$,
- $\sup_{\boldsymbol{m}} \{\|\mathcal{R}_{\alpha(\delta)}\boldsymbol{m} - \boldsymbol{\mu}\| : \|A\boldsymbol{\mu} - \boldsymbol{m}\| \leq \delta\} \to 0$ *as* $\delta \to 0$ *for every* $\boldsymbol{\mu} \in \mathbb{R}^B$.

In search for a regularization method we are essentially looking for a map, called $\mathcal{R}_\alpha$ in this case, that is an approximate for A^{-1}, since this inverse either does not exist, or does not lead to stable solutions. As said before, the regularization parameter α determines the trade-off between the similarity

to the original problem and the stability of the auxiliary problem. This can also be seen in equation (5.1). As the regularization parameter α converges to zero, the auxiliary problem should converge to the original problem, giving that $\mathcal{R}_\alpha A\boldsymbol{\mu} = \boldsymbol{\mu}$ exactly.

The first admissibility condition given in Definition 5.2.1 says that if the error becomes so small such that it converges to zero, the regularization parameter as a function of δ should also converge to zero. If there is no error in our measurement $\boldsymbol{m}$ anymore, the auxiliary problem should equal the original problem, as it becomes stable without an error term. The similarity between the auxiliary and original problem is indicated by a small regularization parameter α. Thus if there is no error and the two problems are the same, the regularization parameter should be zero. Thus when the norm of the error converges to zero, so should the regularization parameter.

The second admissibility condition states that when the norm of the error converges to zero, the largest distance between a reconstructed point $\mathcal{R}_{\alpha(\delta)}\boldsymbol{m}$, for a certain $\mathcal{R}_{\alpha(\delta)}$, and the real point $\boldsymbol{\mu}$ should converge to zero, for every point $\boldsymbol{\mu}$ in $\mathbb{R}^B$. Again, when the size of the error is zero, the auxiliary problem should equal the original problem and should thus result in a perfect reconstruction where $\mathcal{R}_{\alpha(\delta)}\boldsymbol{m} = \boldsymbol{\mu}$.

Once a regularization method $\mathcal{R}_\alpha$ is determined, naive reconstruction using this regularization method, instead of the inverse A^{-1}, becomes

$$\begin{aligned} \boldsymbol{m} &= A\boldsymbol{\mu} + \boldsymbol{\epsilon}, \\ \mathcal{R}_\alpha\boldsymbol{m} &= \mathcal{R}_\alpha A\boldsymbol{\mu} + \mathcal{R}_\alpha\boldsymbol{\epsilon}, \\ \mathcal{R}_\alpha\boldsymbol{m} &= \boldsymbol{\mu}_{app} + \mathcal{R}_\alpha\boldsymbol{\epsilon}, \\ \boldsymbol{\mu}_{app} &= \mathcal{R}_\alpha\boldsymbol{m} - \mathcal{R}_\alpha\boldsymbol{\epsilon}, \end{aligned} \tag{5.2}$$

where $\boldsymbol{\mu}_{app} = \mathcal{R}_\alpha A\boldsymbol{\mu}$ is the approximation of the original $\boldsymbol{\mu}$.

5.3. Well-posedness of a regularization method

Before continuing to regularization methods, we take a step back and consider how we can determine when a regularization method is well-posed, as defined in section 4.3. So in the coming section where we study the regularization methods, we should check the three well-posedness conditions each time. Based on the conclusions in section 4.3 I will summarize methods to easily check whether a regularization method is well-posed or not.

Existence If the matrix A is not square and $K > B$, then there are certain points in $\mathbb{R}^K$ that cannot be reached by $A\boldsymbol{\mu}$. By definition, $\text{Rank}(A) \leq B < K$, so there exists at least one $\boldsymbol{m}_0 \in \text{Coker}(A)$. Just as described in section 4.3, it can then occur that although $A\boldsymbol{\mu} \in \text{Range}(A)$, $\boldsymbol{m} = A\boldsymbol{\mu} + \boldsymbol{\epsilon} \in \text{Coker}(A)$, leading to a non-existing solution.

Thus when checking whether a solution exists, one can look at $\text{Coker}(A)$. If $\text{Coker}(A)$ is nontrivial and there exists an $\boldsymbol{m}_0 \in \text{Coker}(A)$, one should check $\mathcal{R}_\alpha\boldsymbol{m}_0$ to see if it exists.

Uniqueness If the matrix A is not squared and $B < K$, then $\dim(\text{Ker(A)}) > 0$ so we can choose a $\boldsymbol{\mu}_0 \in \text{Ker}(A)$, such that for an $\boldsymbol{m} \in \text{Range}(A)$ it holds that $A(A^{-1}(\boldsymbol{m})) = \boldsymbol{m} = A(A^{-1}(\boldsymbol{m}) + \boldsymbol{\mu}_0)$. Two elements in $\mathbb{R}^B$ map to the same element in $\mathbb{R}^K$, giving an nonunique solution, as we also saw in section 4.3.

Thus when checking whether a solution is unique, one can look at $\text{Ker}(A)$. If $\text{Ker}(A)$ is nontrivial and there exists $\boldsymbol{\mu}_0 \in \text{Ker}(A)$, one should check $\mathcal{R}_\alpha(A\boldsymbol{\mu}_0)$ to see if it is unique.

Stability By Definition 4.3.2, we can estimate whether a matrix is stable or unstable by comparing the largest and smallest nonzero singular values of the matrix.

Thus when checking whether a solution method is stable, one can look at $\text{Cond}(\mathcal{R}_\alpha)$ and check if it is not too small.

5.4. Truncated Singular Value Decomposition Regularization

The first regularization method we will consider is truncated singular value decomposition regularization, abbreviated by TSVD regularization. As the name indicates, it uses the singular value decomposition described in section 4.5, but truncates at a certain point to only keep some of the matrix A, determined by the SVD, in return for a more stable solution. The regularization parameter α determines the trade-off between stability of the auxiliary problem and similarity to the original problem. First I will state the definition of the TSVD regularization solution, after which I will give the expression that satisfies the definition. The definitions and theorem in this section are based on chapter 4 of the book by Mueller and Siltanen [2].

Definition 5.4.1 (Truncated Singular Value Decomposition Regularized solution)**.** *The truncated singular value decomposition regularized solution of equation $\boldsymbol{m} = A\boldsymbol{\mu} + \boldsymbol{\epsilon}$ is the α-dependent vector $\mathcal{L}_\alpha(\boldsymbol{m}) \in \mathbb{R}^B$ that minimizes the expression*

$$\left\|A\mathcal{L}_\alpha(\boldsymbol{m}) - \boldsymbol{m}\right\|, \tag{5.3}$$

such that

$$\left\|\mathcal{L}_\alpha(\boldsymbol{m})\right\| = \inf\{\|\boldsymbol{z}\| : \boldsymbol{z} \text{ minimizes } \left\|A\mathcal{L}_\alpha(\boldsymbol{m}) - \boldsymbol{m}\right\|\}, \tag{5.4}$$

where $\alpha > 0$ is the regularization parameter.

In equation (5.3) we are looking for the least-squares solution, taking the regularization parameter α into consideration. Of course, like in Definition 4.6.1, taking $\alpha = 0$ produces the best least-squares solution, but since $\alpha = 0$ means that the problem is unstable, this is not desirable. When there are multiple solutions satisfying equation (5.3), equation (5.4) ensures the TSVD regularized solution is unique, by taking the solution with the shortest norm.

Theorem 5.4.1 will show that the TSVD regularized solution is given by the inverse of the truncated singular value decomposition of matrix A. Before this theorem is considered, the TSVD of A is defined.

Definition 5.4.2 (Truncated Singular Value Decomposition)**.** *Let A be a $K \times B$ matrix and denote by $A = UDV^T$ the singular value decomposition of A. Let r_α be the largest index for which the corresponding singular value is greater than α, $r_\alpha = \max_{1\le j\le \min(K,B)}\{j | d_j > \alpha\}$. For any $\alpha > 0$, define the truncated singular value decomposition by*

$$A_\alpha^+ = VD_\alpha^+U^T,$$

where

$$D_\alpha^+ = \begin{bmatrix} \frac{1}{d_1} & 0 & \cdots & & & \cdots & 0 \\ 0 & \frac{1}{d_2} & & & & & \vdots \\ \vdots & & \ddots & & & & \\ & & & \frac{1}{d_{r_\alpha}} & & & \\ & & & & 0 & & \\ \vdots & & & & & \ddots & \vdots \\ 0 & \cdots & & & & \cdots & 0 \end{bmatrix}.$$

Now that the TSVD A_α^+ of a matrix A is defined, the following theorem will show that the TSVD regularized solution $\mathcal{L}_\alpha(\boldsymbol{m})$ is given by $A_\alpha^+\boldsymbol{m}$.

Theorem 5.4.1. *Let A be a $K \times B$ matrix and denote by $A_\alpha^+ = VD_\alpha^+U^T$ the TSVD of A (see Definition 5.4.2). The TSVD regularized solution of equation $\boldsymbol{m} = A\boldsymbol{\mu} + \boldsymbol{\epsilon}$ is given by*

$$A_\alpha^+\boldsymbol{m} = VD_\alpha^+U^T\boldsymbol{m}.$$

The proof of this theorem follows the same structure as the proof of Theorem 4.6.1.

Now that we have an expression for the TSVD regularized solution of $\boldsymbol{m} = A\boldsymbol{\mu} + \boldsymbol{\epsilon}$, we can check whether it is well-posed.

Existence In paragraph 5.3 we concluded that in order to check existence, one should check whether Coker(A) is nontrivial and what happens when $\mathcal{L}_\alpha$ is applied to an element in Coker(A).
Suppose that there is a nontrivial $\boldsymbol{m}_0 \in$ Coker(A). If UDV^T is the SVD of A, then U and V are defined to span the Range(A), and are filled with zero vector to make them square. Due to these added zero vectors, any $\boldsymbol{m}_0 \in$ Coker(A) will be mapped to 0, so $\mathcal{L}_\alpha \boldsymbol{m}_0 = 0$. Thus using TSVD regularization, there always exists a solution.

Uniqueness In paragraph 5.3 we concluded that in order to check uniqueness, one should check whether Ker(A) is nontrivial and what happens when $\mathcal{L}_\alpha$ is applied to an element in $A($Ker$(A))$.
Suppose that there is a nontrivial $\boldsymbol{\mu}_0 \in$ Ker(A). Consider $A(\mathcal{L}_\alpha \boldsymbol{m})$ compared to $A(\mathcal{L}_\alpha \boldsymbol{m} + \boldsymbol{\mu}_0)$. Since $\boldsymbol{\mu}_0 \in$ Ker(A), $A(\mathcal{L}_\alpha \boldsymbol{m} + \boldsymbol{\mu}_0) = A\mathcal{L}_\alpha \boldsymbol{m} + A\boldsymbol{\mu}_0 = A\mathcal{L}_\alpha \boldsymbol{m}$, suggesting that the solution is not unique. However, in the definition of the TSVD regularization method, we also have equation (5.4). In case of multiple elements mapping to the same answer, it chooses the one with the smallest norm, ensuring that the solution is always unique. So either $\mathcal{L}_\alpha \boldsymbol{m} + \boldsymbol{\mu}_0$ or $\mathcal{L}_\alpha \boldsymbol{m}$ is the solution, never both.

Stability In paragraph 5.3 we concluded that in order to check for stability, one should check the magnitude of the condition number, which in this case is Cond$((A_\alpha^+)^{-1}) = \frac{d_1}{d_{r_\alpha}}$. This number depends on the regularization parameter α and thus α can be chosen such that problem becomes stable.

All three conditions hold, meaning that TSVD regularization, the map $\mathcal{L}_\alpha : \mathbb{R}^K \to \mathbb{R}^B$ is a well-posed regularization method to our ill-posed initial problem.

In the TSVD regularized solution it is clear how the regularization parameter α determines the trade-off between similarity and stability. In TSVD, the r_α determines until what value singular values to take into account. The more singular values in D_α^+, the more similar $VD_\alpha^+U^T$ is to A, but the bigger the condition number Cond$((A_\alpha^+)^{-1}) = \frac{d_1}{d_{r_\alpha}}$ becomes. So a small α gives similarity to the original problem, but instability, and a large α gives stability, but might be unsimilar to the original problem.

Initially, naive reconstruction using the regular inverse A^{-1}, as in equation (4.2), failed, because this inverse either did not exist, or the problem was too unstable. If we now repeat with TSVD regularization $\mathcal{L}_\alpha$ instead of A^{-1}, as in equation (5.2), we get

$$\begin{aligned}\boldsymbol{\mu}_{app} &= \mathcal{L}_\alpha A\boldsymbol{\mu}, \\ &= \mathcal{L}_\alpha \boldsymbol{m} - \mathcal{L}_\alpha \boldsymbol{\epsilon}, \\ &= VD_\alpha^+U^T\boldsymbol{m} - VD_\alpha^+U^T\boldsymbol{\epsilon}.\end{aligned}$$

In naive inversion using the regular inverse, the norm $\left\|A^{-1}\right\|$ could become very large due to instability. Now we get

$$\begin{aligned}\|\mathcal{L}_\alpha \boldsymbol{\epsilon}\| &= \left\|VD_\alpha^+U^T\boldsymbol{\epsilon}\right\|, \\ &\leq \|V\|\|D_\alpha^+\|\left\|U^T\right\|\|\boldsymbol{\epsilon}\|, \\ &= \|D_\alpha^+\|\|\boldsymbol{\epsilon}\|, \\ &= d_{r_\alpha}^{-1}\|\boldsymbol{\epsilon}\|, \\ &\leq d_{r_\alpha}^{-1}\delta.\end{aligned}$$

We used $\left\|U^T\right\| = 1$ and $\|V\| = 1$ due to orthogonality, $\|D_\alpha^+\| = d_{r_\alpha}^{-1}$ since the norm of a diagonal matrix is its largest entry, and that we approximate and upper bound for the error, $\|\boldsymbol{\epsilon}\| \leq \delta$. In an unstable problem the error term can be magnified, causing an inaccurate reconstruction. Now that we have a stable problem, we can determine an approximated upper bound for the total error term using the regularization parameter α.

5.5. TSVD regularized reconstructions

We will now consider several figures of TSVD regularized reconstructions for the 32×32 and 64×64 Shepp-Logan phantoms with an added noise level of 0.1% as random draws from the normal distribution

with mean 0 and standard deviation equal to the largest absolute entry of the noise-free simulated measurement vector. We will also consider reconstructions from real-life walnut data [4], where the noise level is unknown. The MATLAB code used to generate the reconstructions was based on the MATLAB code by Mueller and Siltanen [7] and can be found in appendix A.1.

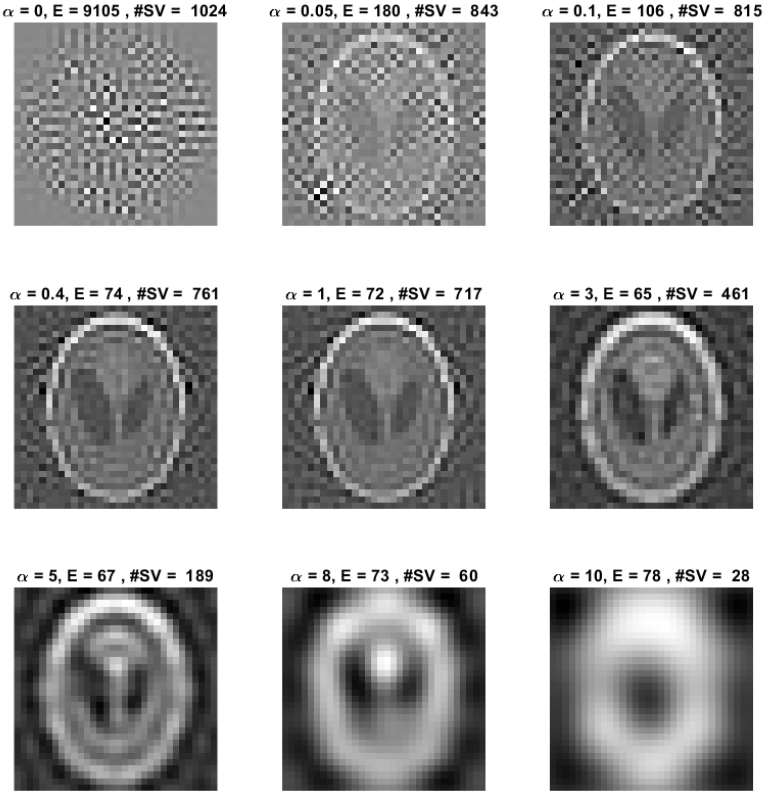


Figure 5.1: TSVD regularized reconstructions for different values of α for the 32×32 Shepp-Logan phantom. For each figure, the α, relative error percentage and number of singular values used for the reconstruction is given.

In figure 5.1 one can see the TSVD regularized reconstructions for the 32×32 Shepp-Logan phantom, for several increasing values of regularization parameter α. Above each reconstruction one can see the value of α, the relative error percentage and the number of singular values that is used in the reconstruction (based on α). In the top row, for low values of α one can see that the reconstruction is still too unstable, as noise blurs the reconstruction. In the bottom row, for higher values of α one can see that the problem is stable, as there is no visible noise effect, but that the reconstruction is too distinct to the original problem since there are only a few singular values used in this reconstruction. In figure 5.2 one can again see the TSVD regularized reconstruction, but this time for the 64×64 Shepp-Logan phantom. Finally, in figure 5.3 one can see the TSVD regularized reconstructions of the real-life walnut measurement data. Here the same pattern can be seen as for the simulated data. In the first row, the solution is still too unstable due to the noise. In the middle row one can clearly see the reconstruction of the inside of the walnut, but the different α values do give different resolutions of reconstructions. In the last row, the reconstruction takes into account too little singular values, so it is too distinct from the original.

The question now remains on what value of regularization parameter α should be used to get the best results. In order to answer this question, one first needs to decide on what a 'good' reconstruction looks like in terms of the image or in terms of error or condition number. Below I will describe several

ways on how I analyzed the problem to give an estimate of choice of the α value. Each time I will comment on the visual quality of the reconstruction.

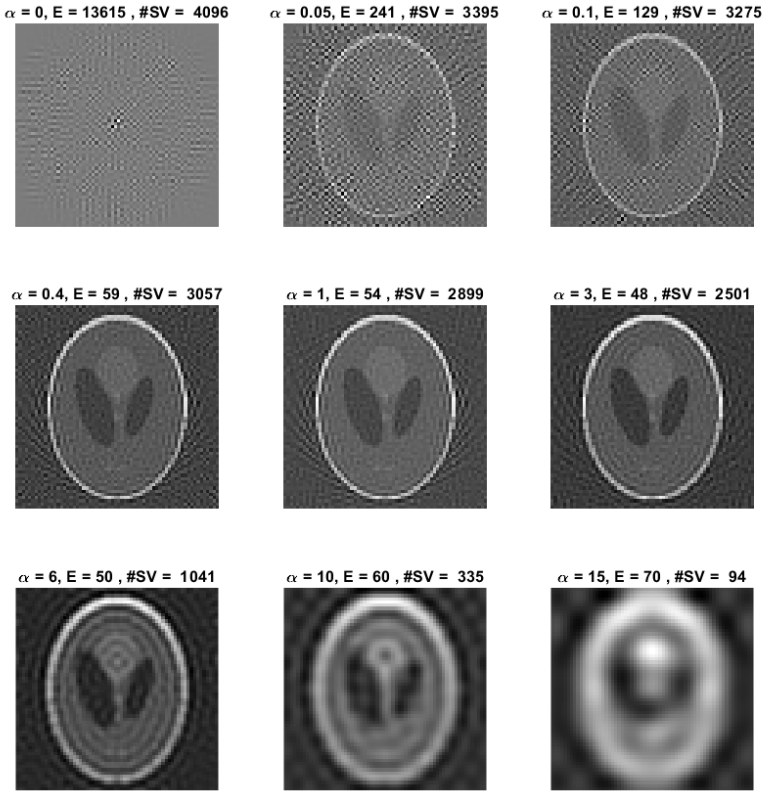


Figure 5.2: TSVD regularized reconstructions for different values of α for the 32×32 Shepp-Logan phantom. For each figure, the α, relative error percentage and number of singular values used for the reconstruction is given.

Lowest relative error Since the Shepp-Logan phantom is simulated data, the reconstruction can be compared to the original phantom. This way, one can calculate the relative error percentage, given by

$$\frac{\|\boldsymbol{\mu}_{app} - \boldsymbol{\mu}\|}{\|\boldsymbol{\mu}\|} \times 100\%.$$

In figure 5.4a one can see the plot of the relative error percentage against the values of α between 0 and 10. The two dotted red lines indicate the lowest and highest α value respectively that give the lowest relative error, which is 64% for the 32×32 phantom (see table 5.1). In this graph one can see that the relative errors are quite large for small α values, but also for large α values, with a minimum in between. This is because the α determines the trade-off between stability and similarity. For low α values the reconstruction is unstable, hence the high relative error percentage. For high α values the problem is too distinct from the original that it is compared with, hence the high relative error percentage. If one wants a reconstruction with the lowest relative error, I would conclude that for the 32×32 phantom an alpha value between $[2.35, 3.07]$ would result in the best reconstruction. In figure 5.4b the same plot was generated for the 64×64 Shepp-Logan phantom. Here, the lowest error is given for α between $[3.76, 3.81]$, with a relative error percentage of 46%. Both these ranges for α give visually good looking reconstructions. A problem with basing α on the relative error percentage is that this can only be done with simulated data where one has the original of what has to be reconstructed. In reality this is not the case, like for the real-life walnut data (data taken from [4]).

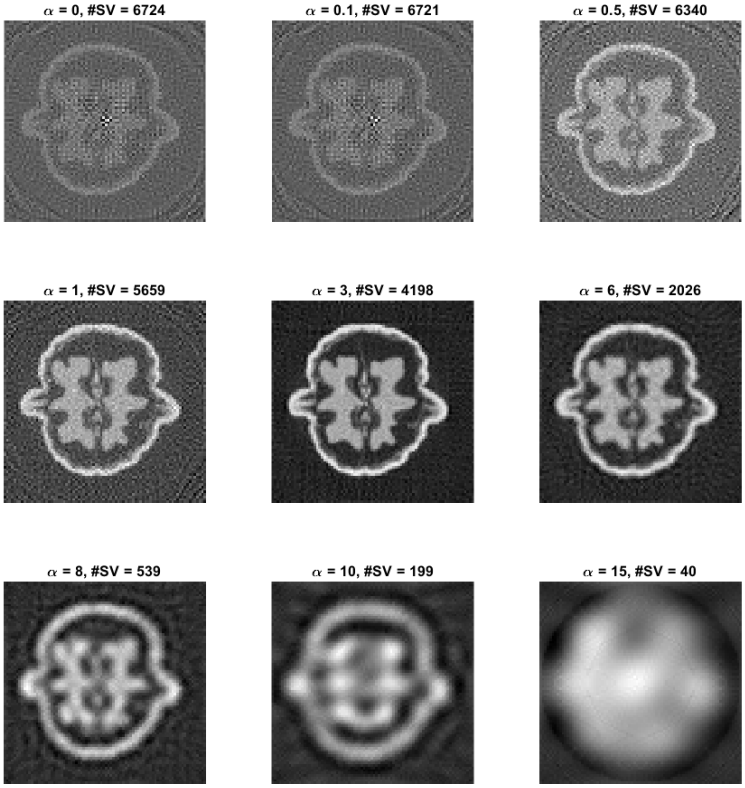
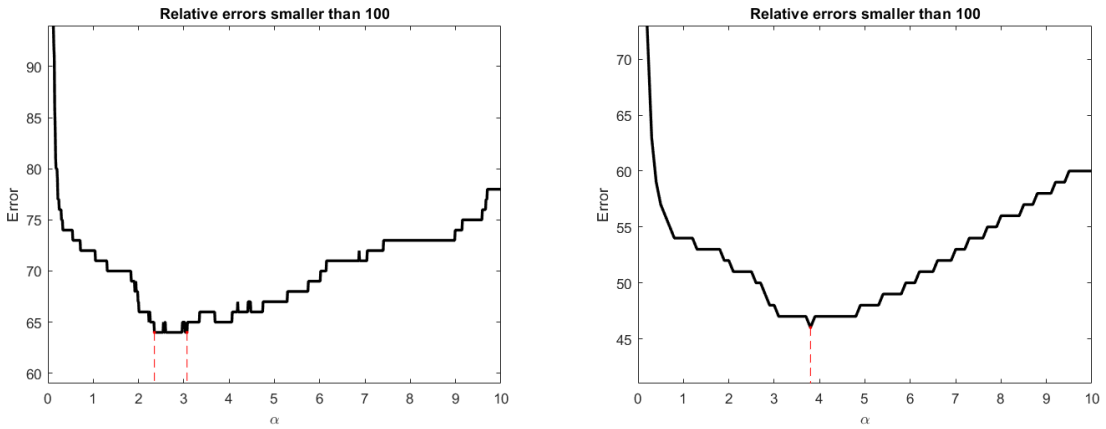


Figure 5.3: TSVD regularized reconstructions for different values of α for the real-life walnut data. For each figure, the α and number of singular values used for the reconstruction is given.



(a) The relative errors corresponding to the TSVD regularized reconstructions with α from 0 to 10 for the 32×32 Shepp-Logan phantom, only including relative errors smaller than 100%.

(b) The relative errors corresponding to the TSVD regularized reconstructions with α from 0 to 10 for the 64×64 Shepp-Logan phantom, only including relative errors smaller than 100%.

Figure 5.4: The relative errors for the 32×32 and 64×64 Shepp-Logan phantom for α between 0 and 10. The lowest relative errors are indicated with a red dot and dashed line.

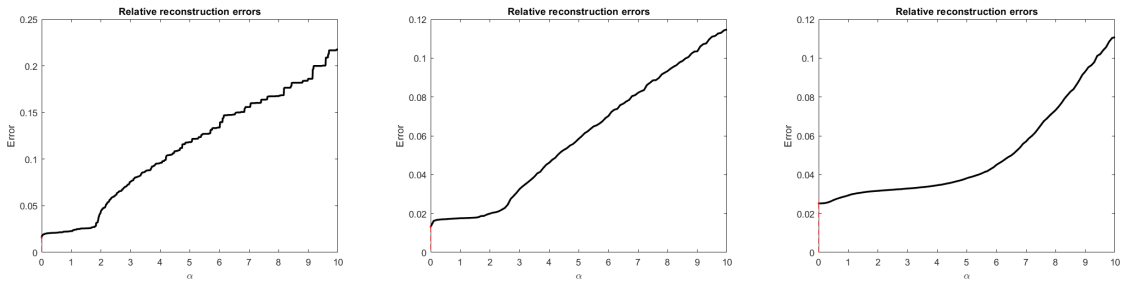
	32×32 phantom	64×64 phantom
Lowest relative error	64%	46%
Corresponding α-values	2.35 - 3.07	3.76 - 3.81

Table 5.1: Lowest relative error percentages for TSVD regularization with the corresponding α values for the 32×32 and 64×64 Shepp-Logan phantom.

Relative reconstruction error Instead of taking the relative error, one can also calculate the relative reconstruction error, given by

$$\frac{\|A\boldsymbol{\mu}_{app} - \boldsymbol{m}\|}{\|\boldsymbol{m}\|} \times 100\%.$$

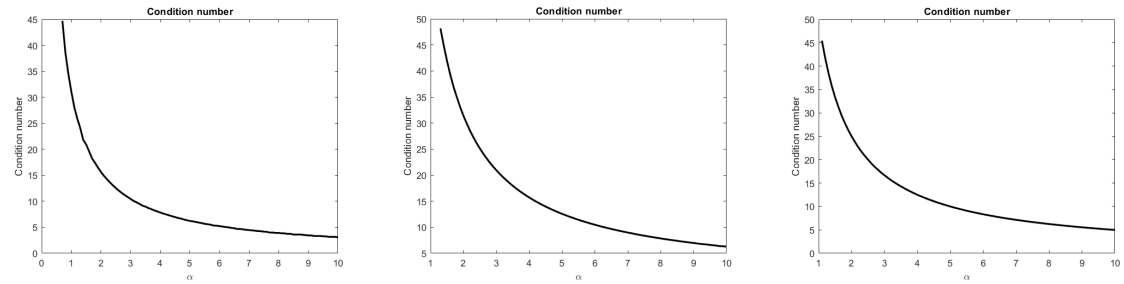
After calculating $\boldsymbol{\mu}_{app}$ you now multiply with A to get what the measurement would be, so the reconstructed measurement, and compare this with the original measurement. This error is different compared to the relative error as described before, as it only takes the similarity to the original measurement into account. In figure 5.5 one can see the relative reconstruction errors plotted against values of α for the 32×32 and 64×64 Shepp-Logan phantoms and the real-life walnut data. Since low α values mean that the reconstruction is more similar to the original, the lowest relative reconstruction errors can be found for the lowest α values. Both plots can be divided into two linear parts, one for low α values and one for high α values. This inflection point can be taken into consideration when deciding on an α value. Note that the inflection point occurs for a higher value of α for the real-life walnut data than for the two phantoms, and it is not as clearly visible. The order of the relative reconstruction errors for the 32×32 phantom is about twice as high as the order of relative reconstruction errors for the 64×64 phantom and the real-life walnut data.



(a) 32×32 Shepp-Logan phantom. (b) 64×64 Shepp-Logan phantom. (c) Real-life walnut data.

Figure 5.5: The relative reconstruction errors for the 32×32 and 64×64 Shepp-Logan phantom and the real-life walnut data for α between 0 and 10. The lowest relative reconstruction error is indicated with a red dot and dashed line.

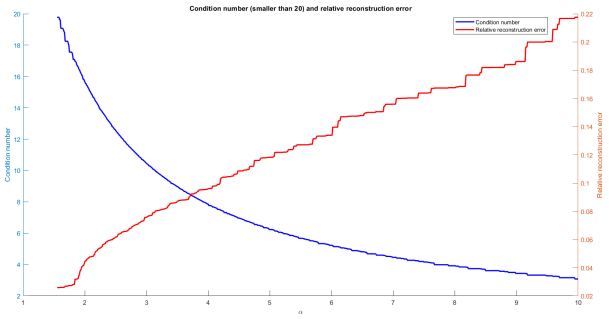
Condition number In figure 5.6 one can see the plots of the condition number against α for the 32×32 and 64×64 Shepp-Logan phantoms and the real-life walnut data. From the definition of the condition number 4.3.2 the shape of the plot makes sense, as it is defined as a fraction with constant numerator. The shape of the three plots is roughly the same and the maximum and minimum values seems to roughly match in all three plots.



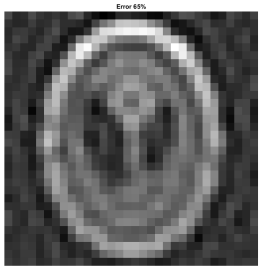
(a) 32×32 Shepp-Logan phantom. (b) 64×64 Shepp-Logan phantom. (c) Real-life walnut data.

Figure 5.6: The condition numbers smaller than 50 for the 32×32 and 64×64 Shepp-Logan phantom and the real-life walnut data for α between 0 and 10.

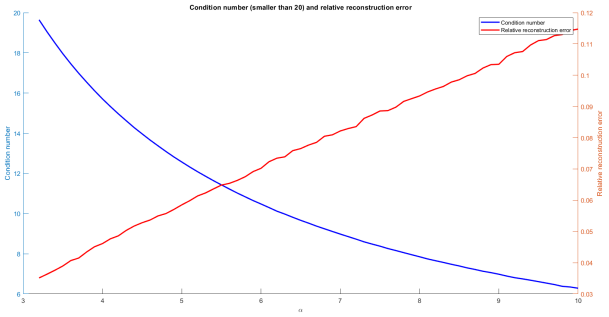
Comparing condition number and relative reconstruction error Since α should give a good trade-off between similarity and stability, I decided to compare a measure of similarity, the relative reconstruction error, and a measure of stability, the condition number. In figure 5.7 one can find the two-sided plots of the condition number and relative reconstruction error for the 32×32 and 64×64 Shepp-Logan phantoms and the real-life walnut data. In these plots one could be tempted to consider the alpha corresponding to the intersection of these two lines as this point should be a balance between the two, but one should keep in mind that the right y-axis do not have the same scale in each plot. However, these reconstructions are included in figure 5.7. Comparing these reconstructions with figures 5.1, 5.2 and 5.3 one can see that each they are not as good visually as they could be. Each time the α value is taken too high, causing the reconstruction to be too distinct from the original. Despite that, the α given by this intersection might be a good starting point to search for a (lower) α value that gives a visually better reconstruction.



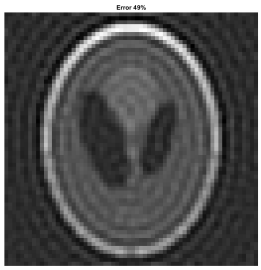
(a) 32×32 Shepp-Logan phantom.



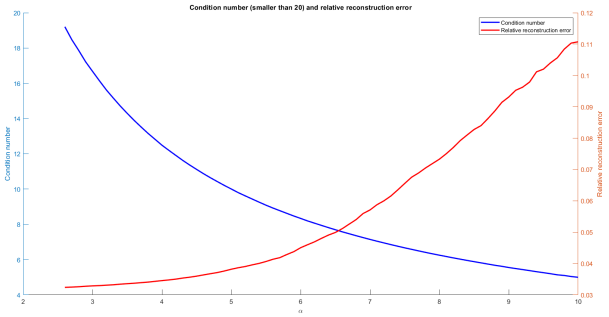
(b) Reconstruction with $\alpha = 3.69$ for the 32×32 Shepp-Logan phantom.



(c) 64×64 Shepp-Logan phantom.



(d) Reconstruction with $\alpha = 5.60$ for the 64×64 Shepp-Logan phantom.



(e) Real-life walnut data.

(f) Reconstruction with $\alpha = 6.4$ for the real-life walnut data.

Figure 5.7: Two-sided plot with the condition number on the left y-axis and the relative reconstruction error on the right y-axis plotted against α between 0 and 10 for the 32×32 and 64×64 Shepp-Logan phantoms and the real-life walnut data.

5.6. Tikhonov Regularization

The second regularization method to consider is Tikhonov regularization. First I will state the definition of the Tikhonov regularized solution, after which I will give the expression that satisfies the definition.

The definitions and theorem in this section are based on chapter 5 from the book by Mueller and Siltanen [2]

Definition 5.6.1 (Tikhonov Regularized Solution)**.** *The Tikhonov regularized solution of equation* $\boldsymbol{m} = A\boldsymbol{\mu} + \boldsymbol{\epsilon}$ *is the vector* $T_\alpha(\boldsymbol{m}) \in \mathbb{R}^B$ *that minimizes the expression*

$$\left\|AT_\alpha(\boldsymbol{m}) - \boldsymbol{m}\right\|^2 + \alpha\left\|T_\alpha(\boldsymbol{m})\right\|^2, \tag{5.5}$$

where $\alpha > 0$ *is the regularization parameter.*

Where in the definition for TSVD regularization, 5.4.1, two expressions were needed to ensure a unique solution, with Tikhonov regularization that is not necessary, due to the second term in equation (5.5). When there are multiple vectors minimizing the first part, the second part is only minimized by the vector with the smallest L^2-norm, thus the total expression (5.5) will be uniquely minimized by one vector.

I will describe two expressions that minimize equation (5.5) and are equivalent, but determined in a different way. Method one is based on singular value decomposition while the second method equates the derivative to zero. One can use the SVD-based method to easily compare the expression of TSVD and Tikhonov regularization. However, since SVD can be computationally demanding, for higher dimensional problems this method is not desirable. Therefore one would use differential method, as it is computationally less demanding while giving the same results.

Tikhonov regularization computed using SVD

Theorem 5.6.1. *Let* A *be a* $K \times B$ *matrix. The Tikhonov regularized solution for equation* $\boldsymbol{m} = A\boldsymbol{\mu} + \boldsymbol{\epsilon}$ *is given by*

$$T_\alpha(\boldsymbol{m}) = V\mathcal{D}_\alpha^+ U^T\boldsymbol{m},$$

where $A = UDV^T$ *is the SVD of* A*, and*

$$\mathcal{D}_\alpha^+ = diag\left(\frac{d_1}{d_1^2 + \alpha}, \ldots, \frac{d_{\min(K,B)}}{d_{\min(K,B)}^2 + \alpha}\right) \in \mathbb{R}^{B\times K}.$$

The proof of this theorem follows the same structure as the proof of Theorem 4.6.1, and can be found as the proof of Theorem 5.1 in Chapter 5 of the book by Mueller and Siltanen [2].

Tikhonov regularization computed without SVD Let A be a $K \times B$ matrix. The Tikhonov regularized solution for equation $\boldsymbol{m} = A\boldsymbol{\mu} + \boldsymbol{\epsilon}$ is given by

$$T_\alpha(\boldsymbol{m}) = (A^TA + \alpha I)^{-1}A^T\boldsymbol{m}.$$

The proof of this theorem is based on finding the minimum using the derivative and equating it to 0. The proof can be found in Section 5.2 in Chapter 5 of the book by Mueller and Siltanen [2].

Note that the expressions given above for $T_\alpha(\boldsymbol{m})$ are equivalent.

Now that we have an expression for the Tikhonov regularized solution of $\boldsymbol{m} = A\boldsymbol{\mu} + \boldsymbol{\epsilon}$, we can check whether it is well-posed. I will show this using the SVD-based expression for $T_\alpha(\boldsymbol{m})$.

Existence In paragraph 5.3 we concluded that in order to check existence, one should check whether Coker(A) is nontrivial and what happens when T_α is applied to an element in Coker(A). The argumentation for Tikhonov is similar to that of TSVD.

Suppose that there is a nontrivial $\boldsymbol{m}_0 \in$ Coker(A). If UDV^T is the SVD of A, then U and V are defined to span the Range(A), and are fill with zero vectors to make them square, if necessary. Due to these added zero vectors, any $\boldsymbol{m}_0 \in$ Coker(A) will now be mapped to zero, $T_\alpha(\boldsymbol{m})_0 = 0$ instead of the solution not being defined. Thus using Tikhonov regularization, there always exists a solution.

Uniqueness By looking at the definition of the Tikhonov regularized solution, Definition 5.6.1, one can already conclude that the term $a\|T_\alpha(\boldsymbol{m})\|$ ensures that when there are multiple solutions possible, the one with the shortest norm is chosen, ensuring a unique solution. In paragraph 5.3 we concluded that in order to check uniqueness, one should check whether Ker(A) is nontrivial and what happens when $T_\alpha(\boldsymbol{m})$ is applied to an element in AKer(A).

Suppose that there is a nontrivial $\boldsymbol{\mu}_0 \in$ Ker(A). Consider $A(T_\alpha(\boldsymbol{m}))$ compared to $A(T_\alpha(\boldsymbol{m}) + \boldsymbol{\mu}_0)$. Since $\boldsymbol{\mu}_0 \in$ Ker(A), $A(T_\alpha(\boldsymbol{m}) + \boldsymbol{\mu}_0) = AT_\alpha(\boldsymbol{m}) + A\boldsymbol{\mu}_0 = AT_\alpha(\boldsymbol{m})$, suggesting that the solution is not unique. However, if we fill the two possibilities in into equation (5.5), we get

$$T_\alpha(\boldsymbol{m}) : \left\|AT_\alpha(\boldsymbol{m})\right\|^2 + \alpha\left\|T_\alpha(\boldsymbol{m})\right\|^2,$$
$$T_\alpha(\boldsymbol{m}) + \boldsymbol{\mu}_0 : \left\|A(T_\alpha(\boldsymbol{m}) + \boldsymbol{\mu}_0)\right\|^2 + \alpha\left\|T_\alpha(\boldsymbol{m}) + \boldsymbol{\mu}_0\right\|^2 \leq \left\|AT_\alpha(\boldsymbol{m})\right\|^2 + \alpha\left\|T_\alpha(\boldsymbol{m})\right\|^2 + \alpha\|\boldsymbol{\mu}_0\|^2$$

If $\boldsymbol{\mu}_0 = 0$, then the two solutions are equivalent. However, when $\boldsymbol{\mu}_0 \neq 0$, then $\|\boldsymbol{\mu}_0\| > 0$ and $T_\alpha(\boldsymbol{m})$ is the expression that minimizes. So the solution is unique.

Stability In paragraph 5.3 we concluded that in order to check for stability, one should check the magnitude of the condition number. In this case that is

$$\text{Cond}((V\mathcal{D}_\alpha^+U^T)^{-1}) = \text{Cond}(U(\mathcal{D}_\alpha^+)^{-1}V^T) = \text{Cond}((\mathcal{D}_\alpha^+)^{-1}) = \frac{(d_1^2 + \alpha)d_{\min(K,B)}}{d_1(d_{\min(K,B)}^2 + \alpha)}.$$

All three conditions hold, meaning that Tikhonov regularization, the map $T_\alpha(\boldsymbol{m}) : \mathbb{R}^K \to \mathbb{R}^B$ is a well-posed regularization method to our ill-posed initial problem.

Where with TSVD regularization the parameter α determined the number of singular values taken into account, there the α is used in a different way to regularize. Here, α puts a weight on singular values. By increasing α, less weight is placed on the small singular values, which again are the one causing most of the instability of the original problem. Note that when α becomes very small, $T_\alpha(\boldsymbol{m})$ effectively becomes the same as the pseudoinverse seen in 4.6.2.

Initially, naive reconstruction using the regular inverse A^{-1} failed, because this inverse either did not exist, or the problem was too unstable. If we now repeat with Tikhonov regularization $T_\alpha(\boldsymbol{m})$ instead of A^{-1}, we get

$$\begin{aligned}\boldsymbol{\mu}_{app} &= T_\alpha(\boldsymbol{m})A\boldsymbol{\mu}, \\ &= T_\alpha(\boldsymbol{m}) - T_\alpha(\boldsymbol{\epsilon}), \\ &= V\mathcal{D}_\alpha^+U^T\boldsymbol{m} + V\mathcal{D}_\alpha^+U^T\boldsymbol{\epsilon}.\end{aligned}$$

In naive inversion using the regular inverse, the norm $\left\|A^{-1}\right\|$ could become very large due to instability. Now we get

$$\begin{aligned}\left\|T_\alpha(\boldsymbol{\epsilon})\right\| &= \left\|V\mathcal{D}_\alpha^+U^T\boldsymbol{\epsilon}\right\|, \\ &\leq \|V\|\|\mathcal{D}_\alpha^+\|\left\|U^T\right\|\|\boldsymbol{\epsilon}\|, \\ &= \|\mathcal{D}_\alpha^+\|\|\boldsymbol{\epsilon}\|, \\ &= \frac{d_{\min(K,B)}^2 + \alpha}{d_{\min(K,B)}}\|\boldsymbol{\epsilon}\|, \\ &\leq \frac{d_{\min(K,B)}^2 + \alpha}{d_{\min(K,B)}}\delta.\end{aligned}$$

We used $\left\|U^T\right\| = 1$ and $\|V\| = 1$ due to orthogonality, $\|\mathcal{D}_\alpha^+\| = \frac{d_{\min(K,B)}^2 + \alpha}{d_{\min(K,B)}}$ since the norm of a diagonal matrix is its largest entry, and that we approximate and upper bound for the error, $\|\boldsymbol{\epsilon}\| \leq \delta$.

5.7. Tikhonov regularized reconstructions

We will now consider several figures of Tikhonov regularized reconstructions for the 32×32 and 64×64 Shepp-Logan phantoms with an added noise level of 0.1% as random draws from the normal distribution with mean 0 and standard deviation equal to the largest absolute entry of the noise-free simulated measurement vector. We will also consider reconstructions from real-life walnut data, where the noise level is unknown [4]. The MATLAB code used to generate the reconstructions was based on the MATLAB code by Mueller and Siltanen [7] and [4] and can be found in appendices A.2 and A.3.

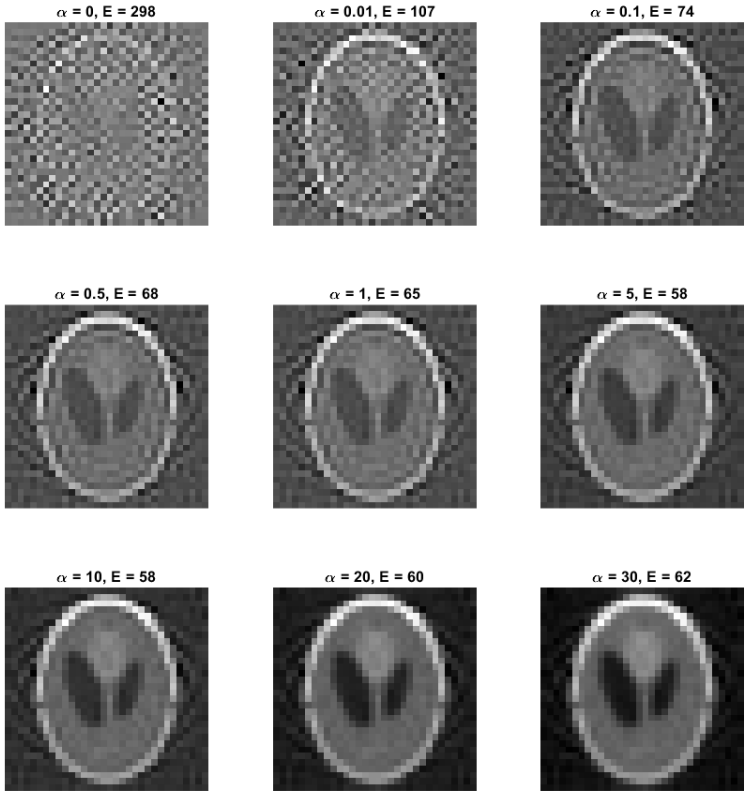


Figure 5.8: Tikhonov regularized reconstructions for different values of α for the 32×32 Shepp-Logan phantom. For each figure, the α and the relative error percentage are given.

In figures 5.8. 5.9 and 5.10 one can see Tikhonov regularized solutions for the 32×32 Shepp-Logan phantom, 64×64 Shepp-Logan phantom and real-life walnut data, respectively. Just as for TSVD regularization, low values of α give noisy, unrecognizable reconstructions. However, higher values of α do not seem to make the reconstruction too distinct from the original, as the reconstructions for high α appear to be quite good. This difference can be explained by the definition of Tikhonov regularization, where the α parameter determines a weight put on the low singular values that cause instability. Since all singular values are taken into account with a weight instead of just some, the Tikhonov regularized reconstruction will always be more similar to the original problem than the TSVD regularized reconstruction for large values of α. For the reconstruction of the walnut data one can see that the parameter choice for α does not appear to make much of a difference. If one looks closely they can see that the plot for the $\alpha = 30$ appears to have more contrast than the plot for $\alpha = 0$, but in both the walnut is clearly visible.

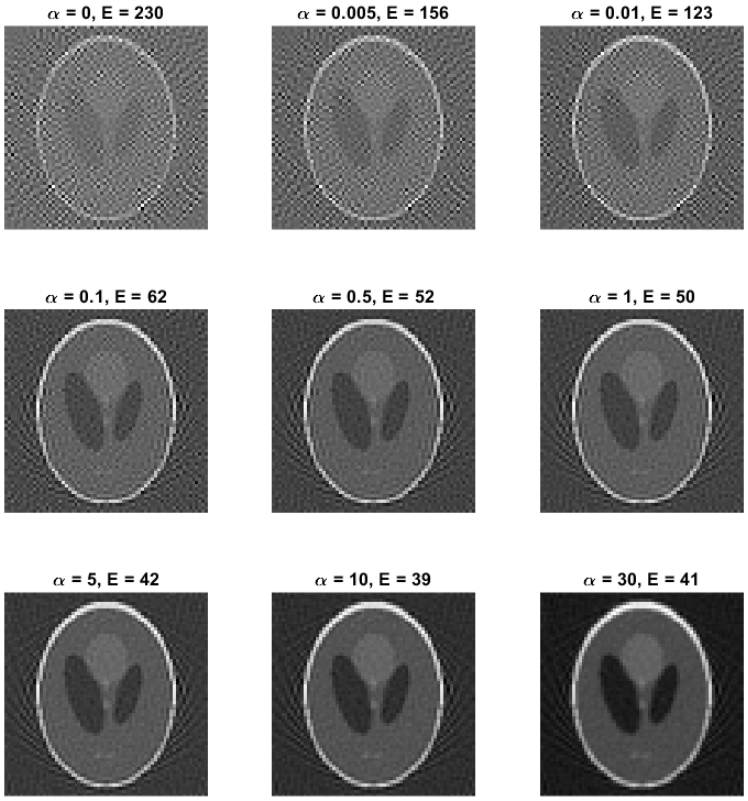


Figure 5.9: Tikhonov regularized reconstructions for different values of α for the 64×64 Shepp-Logan phantom. For each figure, the α and the relative error percentage are given.

Lowest relative error Just as for TSVD regularized solutions, we can compare the reconstruction with the original for the 2 Shepp-Logan phantoms by calculating the relative error. The plots for these errors can be found in figure 5.11. The red dots and dashed lines indicate the α values corresponding to the lowest relative errors. These results can also be found in table 5.2. Again the problem arises that this relative error cannot be computed for the real-life walnut data, so we have to consider another analysis. Note that the range of α values is quite broad and that the reconstruction seems to not so dependent on the α value.

	32×32 phantom	64×64 phantom
Lowest relative error	58%	39%
Corresponding α-values	4.9 - 13.5	8.8 - 20.1

Table 5.2: Lowest relative error percentages for Tikhonov regularization with the corresponding α values for the 32×32 and 64×64 Shepp-Logan phantom.

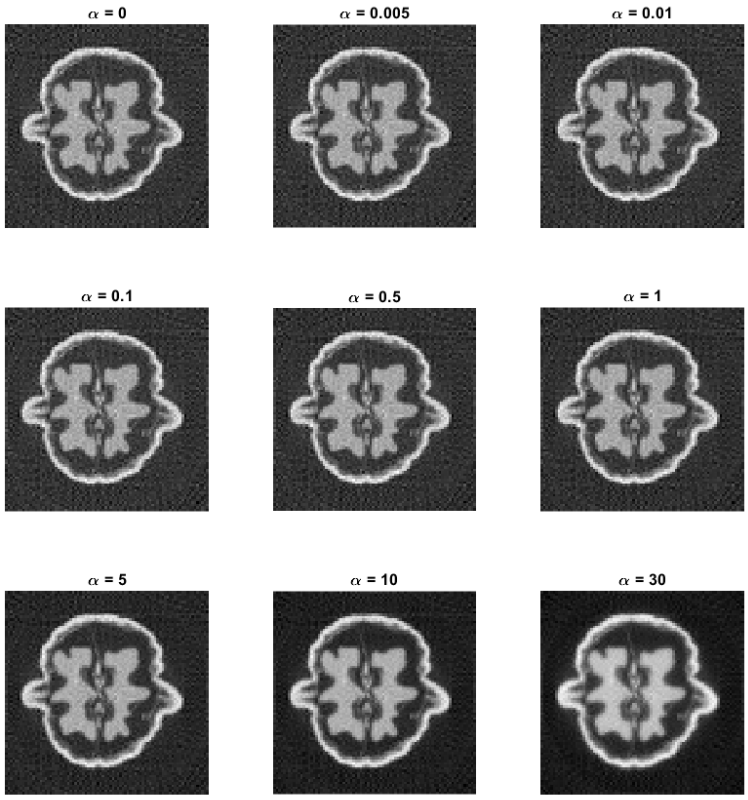
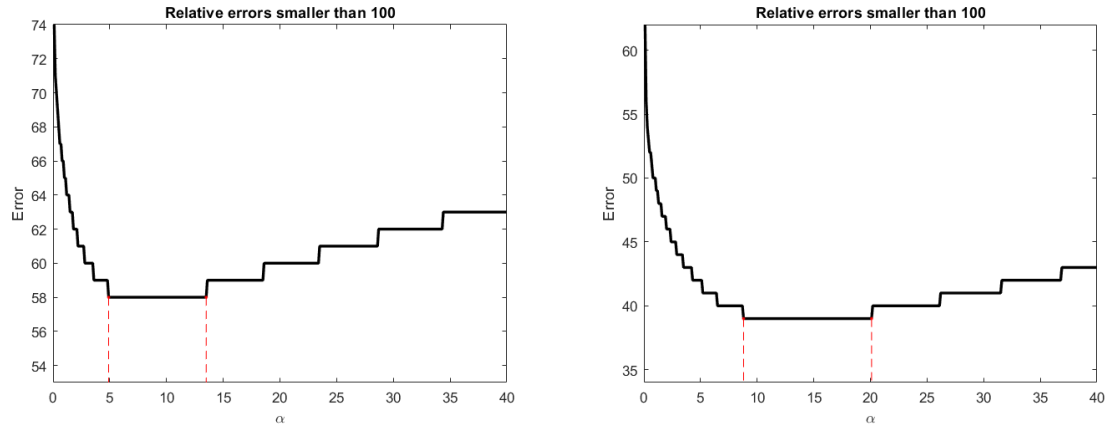


Figure 5.10: Tikhonov regularized reconstructions for different values of α for the real-life walnut data. For each figure, the α is given.



(a) 32×32 Shepp-Logan phantom.

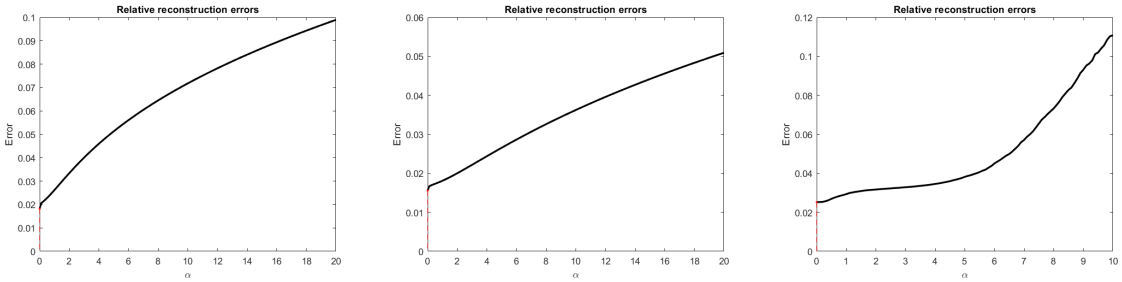
(b) 64×64 Shepp-Logan phantom.

Figure 5.11: The relative errors for the Tikhonov regularized solutions for the 32×32 and 64×64 Shepp-Logan phantom for α between 0 and 10. The lowest relative errors are indicated with a red dot and dashed line.

Relative reconstruction error Instead of taking the relative error, one can also calculate the relative reconstruction error, given by

$$\frac{\|A\boldsymbol{\mu}_{app} - \boldsymbol{m}\|}{\|\boldsymbol{m}\|} \times 100\%.$$

In figure 5.12 the relative reconstruction errors are plotted for both the Shepp-Logan phantoms and the real-life walnut data. Since the reconstruction is compared with the noisy measurement and not with the original, $\alpha = 0$ gives the lowest relative reconstruction error. This makes it difficult to draw conclusions based on these plots. There is a difference between the relative reconstruction errors for the Shepp-Logan phantoms and the real-life walnut data. The plots for the simulated data appear roughly linear, while the real-life walnut data appears convex. An explanation for this difference could be the difference in simulated data versus real-life data, or a noise difference in the measurement. Further research is necessary to determine the exact cause of the shape of the plots for the relative reconstruction errors.



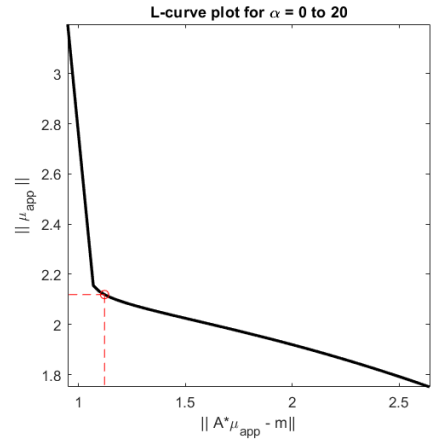
(a) 32×32 Shepp-Logan phantom. (b) 64×64 Shepp-Logan phantom. (c) Real-life walnut data.

Figure 5.12: The relative reconstruction errors for the 32×32 and 64×64 Shepp-Logan phantom and the real-life walnut data for α between 0 and 20. The lowest relative reconstruction error is indicated with a red dot and dashed line.

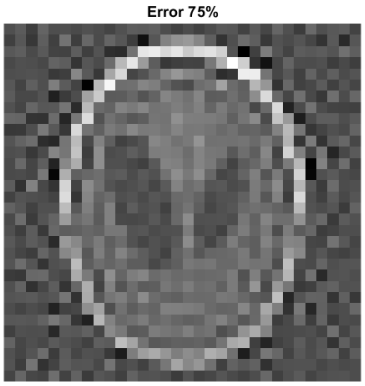
L-curve method A method to obtain an α for Tikhonov regularization is the L-curve method, see chapter 5 of the book by Mueller and Siltanen [2]. In this method one plots $\|A\boldsymbol{\mu}_{app} - \boldsymbol{m}\|$ on the x-axis against $\|\boldsymbol{\mu}_{app}\|$ on the y-axis. Since the aim of Tikhonov regularization is to minimize the sum of these expressions, the far left corner of this plot should give a good α value. In figure 5.13 one can see the L-curves corresponding to the 32×32 and 64×64 Shepp-Logan phantom and the real-life walnut data. The red circle indicates the point that lies closest to the origin. The α corresponding to this point is the chosen α value resulting from the L-curve method. The reconstruction with the determined α is also shown in figure 5.13. Note that the L-curve for the real-life walnut data looks different than the L-curves of the simulated Shepp-Logan phantoms. Note that it seems that the minimum point in the L-curve is reached earlier for the real-life walnut data, but the corresponding α value is 1, which is higher than for both the phantoms. The relative error percentages for the Shepp-Logan phantoms are both higher than the lowest relative error percentage found in table 5.2. If the best α is the one generating the lowest relative error, the L-curve method does not result in the best α. However, when visually comparing the reconstruction that has the lowest relative error with the L-curve based reconstruction, there are no large visual differences.

	32×32	64×64	Walnut data
α	0.3	0.4	1.0
Relative error	75%	59%	-

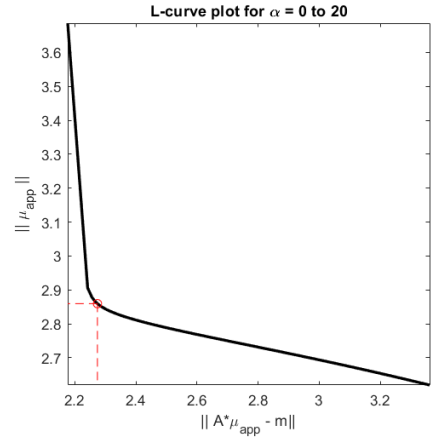
Table 5.3: Determined α value from the L-curve method and relative error percentage.



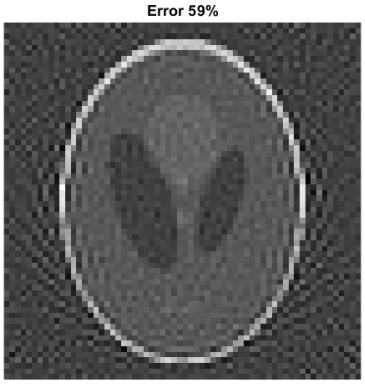
(a) 32×32 Shepp-Logan phantom.



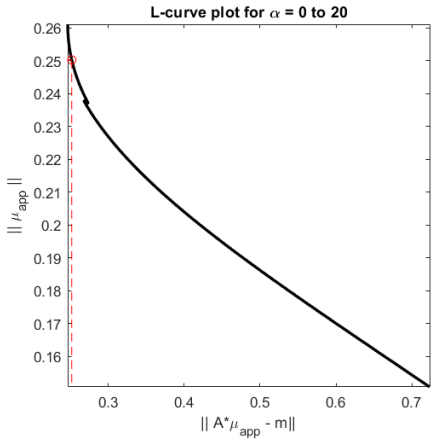
(b) Reconstruction with $\alpha = 0.3$ for the 32×32 Shepp-Logan phantom.



(c) 64×64 Shepp-Logan phantom.



(d) Reconstruction with $\alpha = 0.4$ for the 64×64 Shepp-Logan phantom.



(e) Real-life walnut data.

(f) Reconstruction with $\alpha = 1$ for the real-life walnut data.

Figure 5.13: The L-curves and corresponding Tikhonov regularized reconstructions for the 32×32 and 64×64 Shepp-Logan phantom and the real-life walnut data for α between 0 and 10. The chosen α is indicated with a red circle.

5.8. Regularization method comparison

In this section I will compare the two discussed regularization methods, TSVD and Tikhonov regularization, for the Shepp-Logan phantom of different resolution as well as for the real-life walnut data.

Simulated Shepp-Logan Phantom data Since the data for the Shepp-Logan phantom is simulated data, it is easy to compare the reconstruction with the original. For both TSVD and Tikhonov we were therefore able to calculate the relative error. An overview of the lowest relative errors for both methods and the corresponding values of regularization parameter α can be found in table 5.4.

Method		32×32	64×64
TSVD	Lowest relative error	64%	46%
	α	2.35 - 3.07	3.76 - 3.81
Tikhonov	Lowest relative error	58%	39%
	α	4.9 - 13.5	8.8 - 20.1

Table 5.4: The lowest relative errors and corresponding range of α values for the 32×32 and 64×64 Shepp-Logan phantoms, for TSVD regularization and Tikhonov regularization.

From here one can see that the 64×64 Shepp-Logan phantom gives lower errors than the 32×32 phantom. This is to be expected, since a higher resolution phantom can be more precise in the reconstruction. The lowest relative errors found using Tikhonov regularization are lower than the lowest relative errors resulting from TSVD regularization. This can be explained by the different approached used for TSVD and Tikhonov regularization. In TSVD each singular value has the same weight, while in Tikhonov regularization the singular values causing the instability have a lower weight than the rest of the singular values. Therefore one can be more precise in the regularization process, with lower relative errors as a result. However, it is not clear whether this minimum of relative error is always lower for Tikhonov regularization. This has to be tested for more (different) simulated data tests in order to conclude.

Also note that the range of values of α is higher for the Tikhonov regularized solutions than for the TSVD regularized solutions. This indicates that the TSVD regularized solutions are more sensitive to the choice of parameter α than the Tikhonov regularized solutions. This sensitivity can also be noticed visually when comparing the TSVD reconstruction figures 5.1 and 5.2 with the Tikhonov reconstruction figures 5.8 and figures 5.9. The difference among the reconstructions for TSVD regularized solutions is more noticeable than for the Tikhonov regularized solutions, especially for higher α values. This difference is again a result of the difference in approach of the two regularization methods. Increasing the α in Tikhonov regularization decreases the weight of the lower singular values but has little effect on the higher singular values. Increasing the α in TSVD regularization decreases the number of singular values taken into account for the reconstruction. At some point there are too little singular values considered, resulting in a too general reconstruction. In a situation with real-life data where it is difficult to calculate relative errors, it might be easier to use Tikhonov regularization instead of TSVD regularization, since the choice of α has less influence on the final reconstruction.

Real-life walnut data Since the data for the walnut is real-life data it is not possible to calculate relative errors to determine the quality of the reconstruction, which makes a comparison between regularization methods more complicated. However, a visible judgement on the quality of the reconstruction is possible. Comparing the reconstructions for the two methods, figures 5.3 and 5.10 there are noteworthy differences.

From the 9 chosen reconstructions for TSVD regularization, only the middle 3 can be deemed acceptable reconstructions, with a preference for the middle reconstruction, for $\alpha = 3$. For the 9 chosen Tikhonov regularized reconstructions the opposite holds. When one studies the reconstructions, some differences can be spotted, the main one being the difference in contrast for the lowest and highest α values. Based on these reconstructions alone, it is not really possible to choose a preferred α value. This may sound undesirable, but it does not have to be, since it can be to our advantage that we know that for any α the reconstruction will be good, unlike for the TSVD regularized reconstructions.

Therefore it might be preferred to use Tikhonov regularization over TSVD regularization, as Tikhonov regularization is more likely to give visually accurate looking reconstructions for a broad scope of α values than TSVD regularized reconstrucions.

Difference between the simulated and real-life data A method to determine an appropriate α for the real-life data is to find a method for the simulated Shepp-Logan phantoms and to apply this to the real-life walnut data. This was already done in sections 5.4 and 5.6, but an overview of the findings will be presented here.

Based on the fact that Tikhonov regularization is less sensitive the the choice of α, it is also more difficult to determine when an α value results in a good reconstruction. The determination of α using the L-curve method for Tikhonov regularization found in figure 5.13 gives visually correct looking reconstructions of the phantoms and the walnut. However, as concluded in the previous paragraph, noticing differences in reconstructions for the real-life walnut data of Tikhonov regularized solution is difficult. These differences are more noticeable for the Shepp-Logan phantoms. One can also see a difference in the shapes of the L-curve and relative reconstruction error for the Shepp-Logan phantoms and the real-life walnut data (figures 5.13 and 5.12). It could be that these differences occur as a result of differences between simulated and real-life data, or differences in the resolution (grid) of the reconstruction. Further research using a wider scope of simulated and real-life data is needed to determine the cause of these differences.

The TSVD regularized solutions from the real-life walnut data seem to be more similar to the Shepp-Logan phantom reconstructions than is the case for Tikhonov regularization. The reconstructions looks visually more similar, and the plots for the condition number have the same shape. The biggest difference is spotted in the plot for the relative reconstruction error, just as for Tikhonov regularization. The results for real-life and simulated data are more the same for TSVD regularization than for Tikhonov regularization, likely because TSVD regularization is a more restrictive regularization method, leaving less room for visual differences between the two types of data sets.

6

Conclusion and discussion

The purpose of this thesis was to provide an introduction to the mathematical fundamentals of X-ray tomography as an inverse problem, by showing different reconstruction methods and their mathematical derivations. The main considered methods were unfiltered and filtered backprojection for perfect noise-free data, and minimum norm least-squares, truncated singular value decomposition regularization and Tikhonov regularization for noisy data. These methods were described and visualized by reconstruction of simulated data resulting from Shepp-Logan phantoms and real-life data of the CT measurements of a walnut. Each reconstruction method was analyzed on visual quality and error.

Filtered and unfiltered backprojection Filtered and unfiltered backprojection reconstruction was applied to noise-free data only. The unfiltered backprojection image contains a blur as a result of backprojecting evenly throughout the reconstruction. Filtered backprojection gives a completely accurate reconstruction of the noise-free data. High frequencies are amplified and low frequencies are oppressed, removing the blur found in unfiltered backprojection.

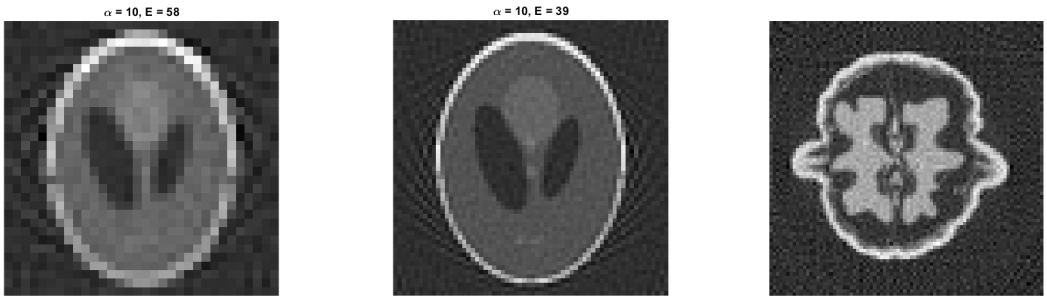
Ill-posedness and the minimum norm least-squares solution It was concluded that practical X-ray tomography is an ill-posed linear inverse problem. This means that either the solution does not always exist, the solution is not always unique, or the solution is unstable. It was also determined that instability immensely deteriorates the reconstruction of the simulated Shepp-Logan phantoms after random noise of 0.1% was added. The naive reconstruction by means of the minimum norm least-squares solution of the 32×32 Shepp-Logan phantom resulted in a relative error of 9105%.

TSVD regularization Using truncated singular value decomposition regularization the quality of the reconstruction visually improved from the minimum norm least-squares reconstruction. For the 32×32 Shepp-Logan phantom the lowest relative error percentage was 64% for regularization parameter α between $2.35 - 3.07$, and for the 64×64 Shepp-Logan phantom this was 46% for α between $3.76 - 3.81$. It was concluded that these α ranges give visually good-looking reconstructions. In order to also determine an appropriate α value for real-life data another method was considered, where the condition number was compared with the relative reconstruction error. It was concluded that these α values give a visually adequate reconstruction, but slightly lower values of α give a visually better looking reconstruction. The α following from this method could function as a starting point when determining a regularization parameter α that results in the visually best looking reconstruction.

Tikhonov regularization Using Tikhonov regularization the quality of the reconstructions visually improved compared to the minimum norm least-squared reconstructions. For the 32×32 Shepp-Logan phantom the lowest relative error percentage was 58% for regularization paramter α between $4.9 - 13.5$, and for the 64×64 Shepp-Logan phantom this was 39% for α between $8.8 - 20.1$. It was concluded that these α ranges give visually good-looking reconstructions. In order to also determine an appropriate α value for real-life data the L-curve method was applied. The α values generated with this method were lower for both the Shepp-Logan phantoms, namely $\alpha = 0.3$ for the 32×32 phantom and $\alpha = 0.4$ for

the 64×64 phantom. However, there were no large visual differences for the reconstructions with the α based on the lowest relative error compared to the α based on the L-curve method.

Regularization method comparison The two regularization methods were compared by means of comparing the reconstructions of the 32×32 and 64×64 Shepp-Logan phantoms and real-life data of a walnut. It was concluded that TSVD regularization is more sensitive to the choice of the α parameter value than Tikhonov regularization. Therefore, when handling real-life data that cannot be compared to an original, a Tikhonov approach is preferred, since Tikhonov regularized solutions for a broad scope of α values are more likely to be of good visual quality than TSVD regularized solutions. The Tikhonov regularized solutions from the simulated data result in lower relative reconstruction errors than the TSVD regularized solutions, implying that Tikhonov regularization is preferred when one has the goal to minimize the relative error. During the analysis of the reconstructions of the simulated data, as well as the real-life data, it was concluded that there are some differences between the two types of data sets, but the reason behind these differences was not determined.



(a) 32×32 Shepp-Logan phantom. (b) 64×64 Shepp-Logan phantom. (c) Real-life walnut data.

Figure 6.1: Visually good looking reconstructions using Tikhonov regularization for the 32×32 and 64×64 Shepp-Logan phantoms and the real-life walnut data.

Recommendations for further research This report considered simulated data from the Shepp-Logan phantom and real-life data of CT scan measurements of a walnut. The differences between two regularization methods and the two data sets were compared. However, since only one real-life data set was considered it is not possible to draw strong conclusions about generalized the findings from the simulated data to the real-life data. Therefore it is recommended to consider more and different real-life data sets for TSVD and Tikhonov regularization. The conclusions from this report can be tested for other real-life data sets to determine whether these methods and conclusions also hold for more real-life data sets.

A considered method to determine a value for the regularization parameter α for TSVD regularization was comparing the condition number and relative reconstruction error. A different version of this approach can be further investigated, where one does not look for an intersection point, but plots the two values against each other and looks for the value with the shortest distance to the origin. This idea is similar to the L-curve method.

The resolutions of the reconstructions considered in this report were 32×32 and 64×64 for the simulated Shepp-Logan phantoms and 82×82 for the real-life walnut data. These resolutions were chosen partially based on computational limits. In further research higher order resolution reconstructions could be computed by reviewing the used code or by investigating different algorithms to use. These higher order resolution reconstructions can be compared to the lower resolution reconstructions to determine whether there are any differences or similarities in for example relative errors and regularization parameter choice.

A

Matlab code

A.1. Truncated Singular Value Decomposition Regularization

This MATLAB code was adjusted from [7] and used in section 5.5 for generating the reconstructions and images for the Shepp-Logan phantoms. After a minor changes in parameter notation, this code also works for the real-life walnut data.

```
% Compute the TSVD reconstruction for the phantom with chosen alpha
% Needed: XRME_SVD_comp.m and XRMC_NoCrimeData_comp.m

% Choose resolution
N = 32;

% % Choose alpha
alpha_u = 10;
alpha_l = 0;
step = 0.1;
alpha_r = [alpha_l:step:alpha_u];
L = length(alpha_r);

% Load noise measurement without inverse crime
eval(['load XRMC_NoCrime', num2str(N), ' N mnc mncn']);
mn = mncn;

% Load singular value decomposition of the measurement matrix
eval(['load XRME_SVD', num2str(N), ' U D V A measang target N P Nang']);

% Make empty matrices
[row,col] = size(D.');
svals = diag(D);
relerr_all = [];
reconTSVD_all = sparse(N^2,length(alpha_r));
ral_all = [];
err_all = [];
d_min = [];
d_1 = [];

figure(1);

for aaa = 1:L
    alpha = alpha_r(aaa);
    Dplus = sparse(row,col);
```

```
    for iii = 1:length(diag(D))
        if svals(iii) > alpha
            Dplus(iii,iii) = 1/svals(iii);
            ral = iii;
            d_min(aaa) = svals(iii);
        end
    end
    ral_all(aaa) = ral;
    reconTSVD = V*Dplus*U.'*mn(:);
    reconTSVD_all(:,aaa) = reconTSVD;
    err = norm(A*reconTSVD(:)-mn(:))/norm(mn(:));
    err_all(aaa) = err;
    d_1 = Dplus(1,1);
    relerr = round(norm(reconTSVD(:)-target(:))/norm(target(:))*100);
    relerr_all(aaa) = relerr;
    % Plot the reconstruction in one large plot for all alpha
    subplot(ceil(L/ceil(sqrt(L))),ceil(sqrt(L)),aaa);
    imagesc(reshape(reconTSVD,N,N));
    title(['\alpha = ', num2str(alpha), ', E = ', num2str(relerr), ' , #SV
        =  ',num2str(ral)]);
    colormap gray
    axis square
    axis off
    disp(aaa);
end

% calculate all condition numbers
cond = (1./d_1)./d_min;

% Plot the singular values (log y)
XRME_SVD_plot(N)

% Plot the relative errors, for errors smaller than 100
minrerr = min(relerr_all);
minrerr_r = find(relerr_all == minrerr);
ind_e = min(find(relerr_all < 100));

figure(2);
clf
plot(alpha_r(ind_e:L), relerr_all(ind_e:L), 'k','linewidth',2);
title('Relative errors smaller than 100');
xlabel('\alpha');
ylabel('Error');
ylim([minrerr-5 max(relerr_all(ind_e:L))])
hold on
plot(alpha_r(min(minrerr_r)),minrerr, 'r.');
plot(alpha_r(max(minrerr_r)),minrerr, 'r.');
plot([alpha_r(min(minrerr_r)) alpha_r(min(minrerr_r))], [minrerr minrerr
    -5], 'r--');
plot([alpha_r(max(minrerr_r)) alpha_r(max(minrerr_r))], [minrerr minrerr
    -5], 'r--');


% Plot the relative reconstruction errors
minerr = min(err_all);
minerr_r = find(err_all == minerr);
```

```
figure(3);
clf
plot(alpha_r, err_all, 'k','linewidth',2);
title('Relative reconstruction errors');
xlabel('\alpha');
ylabel('Error');
hold on
plot(alpha_r(min(minerr_r)),minerr, 'r.');
plot(alpha_r(max(minerr_r)),minerr, 'r.');
plot([alpha_r(min(minerr_r)) alpha_r(min(minerr_r))], [minerr 0], 'r--');
plot([alpha_r(max(minerr_r)) alpha_r(max(minerr_r))], [minerr 0], 'r--');

% Plot the condition numbers
ind_con50 = min(find(cond<50));

figure(4);
clf
plot(alpha_r(ind_con50:L),cond(ind_con50:L), 'k','linewidth',2);
title('Condition number');
xlabel('\alpha');
ylabel('Condition number');
hold on
plot(alpha_r(ind_con50:L),ral_all(ind_con50:L)/10);


% Plot rescaled sum of condition number and relative reconstruction errors
conlim = 20;
ind_con20 = min(find(cond < conlim));
cond_re = rescale(cond(ind_con20:L),0,1);
err_all_re = rescale(err_all(ind_con20:L),0,1);
intersec = min(find(cond_re <= err_all_re));

figure(5);
clf
plot(alpha_r(ind_con20:L), (cond_re+err_all_re), 'k','linewidth',2);
title(['Condition number (smaller than ',num2str(conlim), ') plus relative
     reconstruction error, scaled between 0 and 1']);
xlabel('\alpha');
ylabel('Condition number plus relative reconstruction error');
hold on
plot(alpha_r(ind_con20:L),cond_re,'b', 'linewidth', 1);
plot(alpha_r(ind_con20:L),err_all_re, 'r','linewidth', 1);
plot(alpha_r(intersec+ind_con20-1), cond_re(intersec), 'ko','linewidth',
    2);
plot([alpha_r(intersec+ind_con20-1) alpha_r(intersec+ind_con20-1)], [
    cond_re(intersec) 0], 'k--');
plot([alpha_r(intersec+ind_con20-1) 1], [cond_re(intersec) cond_re(
    intersec)], 'k--');
plot(alpha_r(intersec+ind_con20-2), cond_re(intersec-1), 'ko','linewidth',
     2);
plot([alpha_r(intersec+ind_con20-2) alpha_r(intersec+ind_con20-2)], [
    cond_re(intersec-1) 0], 'k--');
plot([alpha_r(intersec+ind_con20-2) 1], [cond_re(intersec-1) cond_re(
    intersec-1)], 'k--');
```

```
legend('Sum condition number and relative reconstruction error', '
    Condition number', 'Relative reconstruction error');

figure(6);
clf
yyaxis left
title(['Condition number (smaller than 20) and relative reconstruction
    error']);
% Plot the condition number with left y-axis
xlabel('\alpha');
hold on
ylabel('Condition number');
plot(alpha_r(ind_con20:L),cond(ind_con20:L),'b','linewidth', 2);
% Plot the relative reconstruction error with right y-axis
hold on
yyaxis right
ylabel('Relative reconstruction error');
plot(alpha_r(ind_con20:L),err_all(ind_con20:L),'r','linewidth', 2);
legend('Condition number','Relative reconstruction error');
```

A.2. Tikhonov Regularization for Shepp-Logan Phantoms

This MATLAB code was adjusted from [7] and used in section 5.7 for generating the reconstructions and images for the Shepp-Logan phantoms.

```
    % Choose resolution
N = 32;

% Choose alpha range
alpha_l = 0;
alpha_u = 20;
step = 0.1;
alpha_r = [alpha_l:step:alpha_u];
L = length(alpha_r);

tic

% Load measurement matrix A
eval(['load RadonMatrix', num2str(N), ' A measang target N P Nang']);

% Load noise measurement without inverse crime
eval(['load XRMC_NoCrime', num2str(N), ' N mnc mncn']);
mn = mncn;

% A transpose times measurement
b = A.'*mn(:);


% Make empty matrices
relerr_all = [];
err_all = [];
reconTi_all = [];
lognormdiff = [];
lognormrecon = [];

figure(1);
clf
```

```
for aaa = 1:L
    K = 400;                % Number of iterations
    x = b;
    rho = zeros(K,1);
    alpha = alpha_r(aaa);
    Hx = (A.')*(A*x) + alpha*x;
    r = b - Hx;
    rho(1) = r.'*r;

    % Start iteration
    for kkk = 1:K
        if kkk==1
            p = r;
        else
            beta = rho(kkk)/rho(kkk-1);
            p = r + beta*p;
        end
        w = (A.')*(A*p) + alpha*p;
        a = rho(kkk)/(p.'*w);
        x = x + a*p;
        r = r - a*w;
        rho(kkk+1) = r.'*r;
        disp([kkk K]);
    end
    % Calculate reconstruction
    recn = reshape(x,N,N);
    reconTi_all(:,aaa) = x;
    % Calculate relative error
    relerr = round(norm(recn(:)-target(:))/norm(target(:))*100);
    relerr_all(aaa) = relerr;
    % Calculate norms for L-curve
    lognormdiff(aaa) = log(norm(A*x-mn(:)));
    lognormrecon(aaa) = log(norm(x));
    % Calculate relative reconstruction errors
    err = norm(A*recn(:)-mn(:))/norm(mn(:));
    err_all(aaa) = err;
    % Plot reconstruction
    subplot(ceil(L/ceil(sqrt(L))),ceil(sqrt(L)),aaa);=
    imagesc(recn);
    title(['\alpha = ', num2str(alpha), ', E = ', num2str(relerr)]);
    colormap gray
    axis square
    axis off
    disp(aaa);
end

% Plot the relative errors, for errors smaller than 100
minrerr = min(relerr_all);
minrerr_r = find(relerr_all == minrerr);
ind_e = min(find(relerr_all < 100));

figure(2);
clf
plot(alpha_r(ind_e:L), relerr_all(ind_e:L), 'k','linewidth',2);
title('Relative errors smaller than 100');
```

```
xlabel('\alpha');
ylabel('Error');
ylim([minrerr-5 max(relerr_all(ind_e:L))])
hold on
plot(alpha_r(min(minrerr_r)),minrerr, 'r.');
plot(alpha_r(max(minrerr_r)),minrerr, 'r.');
plot([alpha_r(min(minrerr_r)) alpha_r(min(minrerr_r))], [minrerr minrerr
    -5], 'r--');
plot([alpha_r(max(minrerr_r)) alpha_r(max(minrerr_r))], [minrerr minrerr
    -5], 'r--');

% Plot the relative reconstruction errors
minerr = min(err_all);
minerr_r = find(err_all == minerr);

figure(3);
clf
plot(alpha_r, err_all, 'k','linewidth',2);
title('Relative reconstruction errors');
xlabel('\alpha');
ylabel('Error');
hold on
plot(alpha_r(min(minerr_r)),minerr, 'r.');
plot(alpha_r(max(minerr_r)),minerr, 'r.');
plot([alpha_r(min(minerr_r)) alpha_r(min(minerr_r))], [minerr 0], 'r--');
plot([alpha_r(max(minerr_r)) alpha_r(max(minerr_r))], [minerr 0], 'r--');


% Plot the L-currve
figure(5);
clf

minx = min(lognormdiff);
miny = min(lognormrecon);
maxx = max(lognormdiff);
maxy = max(lognormrecon);

dis2 = [];
dis3 = [];
for bbb=1:L
    dis2(bbb) = norm([lognormdiff(bbb),lognormrecon(bbb)] - [minx,miny]);
    dis3(bbb) = norm([lognormdiff(bbb),lognormrecon(bbb)]);
end

min(dis3);
a3 = find(dis3 == min(dis3));
alpha_r(a3)
disp(['L-cuve alpha value = ', num2str(alpha_r(a3))]);

plot(lognormdiff,lognormrecon,'k', 'linewidth',2);
title(['L-curve plot for \alpha = ', num2str(alpha_l),' to ', num2str(
    alpha_u)])
axis([minx maxx miny maxy]);
axis square
xlabel('|| A*\mu_{app} - m||');
ylabel('|| \mu_{app} ||');
```

```
hold on
plot(lognormdiff(a3),lognormrecon(a3),'ro');
plot([lognormdiff(a3) lognormdiff(a3)],[lognormrecon(a3) miny],'r--');
plot([lognormdiff(a3) minx],[lognormrecon(a3) lognormrecon(a3)],'r--');

toc
```

A.3. Tikhonov Regularization for real-life walnut data

This MATLAB code was adjusted from [4] and [7] and used in section 5.7 for generating the reconstructions and images for the real-life walnut data.

```
    % Choose resolution
N = 82;

% Choose range of regularization parameter
alpha_u = 20;
alpha_l = 0;
step = 0.1;
alpha_r = [alpha_l:step:alpha_u];
L = length(alpha_r);

% Load measurement and A
eval(['load Data', num2str(N)]);

% Empty matrices
reconTi_all = [];
err_all = [];
lognormdiff = [];
lognormrecon = [];

figure(1);
clf

for aaa = 1:L
    alpha = alpha_r(aaa);
    fun = @(x) A.'*(A*x)+alpha*x;
    b = A.'*m(:);
    x = pcg(fun,b);
    reconTi_all(:,aaa) = x;
    recon = reshape(x,N,N);
    lognormdiff(aaa) = log(norm(A*x - m(:)));
    lognormrecon(aaa) = log(norm(x));
    err = norm(A*recon(:)-m(:))/norm(m(:));
    err_all(aaa) = err;
    subplot(ceil(L/ceil(sqrt(L))),ceil(sqrt(L)),aaa);
    imagesc(recon);
    title(['\alpha = ', num2str(alpha)]);
    colormap gray
    axis square
    axis off
    disp(aaa);
end

% Plot the relative reconstruction errors
figure(2);
clf
```

```
minerr = min(err_all);
minerr_r = find(err_all == minerr);

plot(alpha_r, err_all, 'k','linewidth',2);
title('Relative reconstruction errors');
xlabel('\alpha');
ylabel('Error');
hold on
plot(alpha_r(min(minerr_r)),minerr, 'r.');
plot(alpha_r(max(minerr_r)),minerr, 'r.');
plot([alpha_r(min(minerr_r)) alpha_r(min(minerr_r))], [minerr 0.02], 'r--'
    );
plot([alpha_r(max(minerr_r)) alpha_r(max(minerr_r))], [minerr 0.02], 'r--'
    );


% Plot the L-curve
figure(3);
clf

minx = min(lognormdiff);
miny = min(lognormrecon);
maxx = max(lognormdiff);
maxy = max(lognormrecon);

dis2 = [];
dis3 = [];
for bbb=1:L
    dis2(bbb) = norm([lognormdiff(bbb),lognormrecon(bbb)] - [minx,miny]);
    dis3(bbb) = norm([lognormdiff(bbb),lognormrecon(bbb)]);
end

min(dis3);
a3 = find(dis3 == min(dis3));
alpha_r(a3)
disp(['L-cuve alpha value = ', num2str(alpha_r(a3))]);

plot(lognormdiff,lognormrecon,'k', 'linewidth',2);
title(['L-curve plot for \alpha = ', num2str(alpha_l),' to ', num2str(
    alpha_u)])
axis([minx maxx miny maxy]);
axis square
xlabel('|| A*\mu_{app} - m||');
ylabel('|| \mu_{app} ||');
hold on
plot(lognormdiff(a3),lognormrecon(a3),'ro');
plot([lognormdiff(a3) lognormdiff(a3)],[lognormrecon(a3) miny],'r--');
plot([lognormdiff(a3) minx],[lognormrecon(a3) lognormrecon(a3)],'r--');
```

Bibliography

[1] W.G. Bradley. “History of Medical Imaging”. In: *Proceedings of the American Philosophical Society* 152.3 (2008), pp. 349–361. ISSN: 0003049X. URL: http://www.jstor.org/stable/40541591 (visited on 07/03/2023).

[2] J.Mueller and S.Siltanen. *Linear and nonlinear inverse problems with practical applications*. Computational Science and Engineering. Society for Industrial and Applied Mathematics, 2012. ISBN: 978-1-611972-33-7.

[3] L.A. Shepp and B.F. Logan. “The Fourier reconstruction of a head section”. In: *IEEE Transactions on Nuclear Science* 21.3 (1974), pp. 21–43. DOI: 10.1109/TNS.1974.6499235.

[4] K. Hämäläinen et al. *Tomographic X-Ray Data Of A Walnut*. en. Feb. 2015. DOI: 10.5281/ZENODO.1254206. URL: https://zenodo.org/record/1254206.

[5] J.L. Prince and J.M. Links. *Medical Imaging Signals and Systems*. 2nd ed. Pearson, 2015.

[6] T.K. Wong and S C.P. Yam. “A probabilistic proof for Fourier inversion formula”. In: *Statistics & Probability Letters* 141 (2018), pp. 135–142. ISSN: 0167-7152. DOI: https://doi.org/10.1016/j.spl.2018.05.028. URL: https://www.sciencedirect.com/science/article/pii/S0167715218302153.

[7] J. Mueller and S. Siltanen. *MATLAB*. Oct. 2012. URL: https://archive.siam.org/books/cs10/Xray_WithMatrix/index.php.

[8] H. Yanai, K. Takeuchi, and Y. Takane. “Singular Value Decomposition (SVD)”. In: *Projection Matrices, Generalized Inverse Matrices, and Singular Value Decomposition*. New York, NY: Springer New York, 2011, pp. 125–149. ISBN: 978-1-4419-9887-3. DOI: 10.1007/978-1-4419-9887-3_5. URL: https://doi.org/10.1007/978-1-4419-9887-3_5.

[9] A. Wirgin. “The inverse Crime”. In: *Math Phys* (Feb. 2004). DOI: https://doi.org/10.48550/arXiv.math-ph/0401050.

[10] J. Kaipio and E. Somersalo. *Statistical and computational inverse problems*. Vol. 160. Springer Science & Business Media, 2006.

[11] J. Cheng and B. Hofmann. “Regularization Methods for Ill-Posed Problems”. In: *Handbook of Mathematical Methods in Imaging*. Ed. by Otmar Scherzer. New York, NY: Springer New York, 2011, pp. 87–109. ISBN: 978-0-387-92920-0. DOI: 10.1007/978-0-387-92920-0_3. URL: https://doi.org/10.1007/978-0-387-92920-0_3.