Technische Universiteit Delft
Faculteit Elektrotechniek, Wiskunde en Informatica
Delft Institute of Applied Mathematics

Computer Tomography: Image reconstruction in the presence of noise

Verslag ten behoeve van het
Delft Institute of Applied Mathematics
als onderdeel ter verkrijging

van de graad van

BACHELOR OF SCIENCE
in
TECHNISCHE WISKUNDE

door

I.N. van Dorp

Delft, Nederland
Juli 2015

BSc verslag TECHNISCHE WISKUNDE

"Computer Tomography: Image reconstruction in the presence of noise"

I.N. van Dorp

Technische Universiteit Delft

Begeleiders

Dr.ir. M.B. van Gijzen
Prof.dr.ir. G. Jongbloed

Overige commissieleden

Dr. D.C. Gijswijt
Dr.ir. M. Keijzer

Juli, 2015
Delft

PREFACE

Making choices never has been one of my strengths. Four years ago, I decided to start studying at the faculty of Industrial Design Engineering (IDE) at Delft University of Technology. But after a couple of months, I had to conclude that my love for mathematics was not fulfilled at IDE. A time of consideration started and after visiting Applied Mathematics, I finally made the choice of studying mathematics: a choice I have enjoyed the last three years and still enjoy every day!

My difficulty of making choices was also notable when it came to choosing a project for my Bachelor Thesis. During the three years of my bachelor in Applied Mathematics, I did like studying almost every area of mathematics, making it even harder to decide in which field I would like to do my final research. There was a small preference in me for Numerical Analysis and Statistics, which led to my first step of talking to my mentors.

Luckily, they understood my doubt and were even open for a collaboration: both fields did work with Computer Tomography, but from another perspective. I would like to thank my mentors, Martin van Gijzen and Geurt Jongbloed, for all the effort they have put into providing me with a project in both my fields of interest and for supporting me throughout the formation of this report.

This report is the final output of my bachelor Applied Mathematics at Delft University of Technology and the conclusion to my first scientific degree. I hope you will enjoy reading, criticizing and valuing this Bachelor Thesis.

I.N. van Dorp
Delft, July 2015

ABSTRACT

In this research, a stochastic model for attenuation in Computer Tomography is developed. This model gives rise to the idea of using path dependent variance of the measurements (instead of constant variance) to improve image reconstruction. The distribution of the measurements in this model is determined and a difference with the current literature is found, which leads to a refinement of the noise or measurement errors in the model.

To use the information about the variance of the measurements in the image reconstruction, a numerical model is considered in which a discretization is made of the tomographic image that has to be reconstructed, i.e., the unknown attenuation coefficients. Incorporating weights to reflect the relation between the area that is traversed by an X-ray beam and the entire area of a pixel in the grid results in a linear system of equations. Because the measurements are not exact, noise is added to this linear system of equations, which leads to a perturbed problem.
A transformation of the measurements is needed to obtain the desired linear system of equations. The Delta Method is used for this purpose. Another method used for the transformation of stochastic models, variance stabilization, is briefly considered.

The log-likelihood of the unknown attenuation coefficients is determined under different assumptions for the mean and variance of the measurements. A connection is made between the log-likelihood and the (weighted) Least-Squares Estimation, leading to different ideas for the adjustment of the current reconstruction algorithm.

Several new reconstruction algorithms are developed to improve the image reconstruction by using the path dependent variance of the noise. Most of these new algorithms result in a better reconstruction than the current algorithm, but a problem is found when the convergence of the iterative algorithms is considered.
In addition to the relative error, the log-likelihood function and the weighted sum of squared errors are used to investigate the convergence of the iterative reconstruction algorithms. Relaxation is incorporated into the iterative reconstruction algorithm to improve the convergence. A slightly better convergence is obtained, but progress could be made if a convergent iterative algorithm is found.

CONTENTS

1

INTRODUCTION

The goal of this bachelor project is to use the path dependent variance of the measurements to improve the image reconstruction. In the current algorithm used for the reconstruction, the variance of the errors is assumed to be constant. This algorithm will be adapted to a fixed point algorithm, which uses the path dependent variance and hopefully results in a more accurate image reconstruction.

1.1. RESEARCH OBJECTIVES

In order to reach the goal of this bachelor project, more knowledge is obtained about Computer Tomography and the underlying mathematical model. It is made clear what kind of 'noise' we are dealing with during the image reconstruction. A stochastic model which incorporates this noise is constructed. From this model, more information about the 'noise' or measurement errors is obtained. This information is used to improve the image reconstruction.

To use this information about the measurement errors, a connection should be made between the stochastic model and the numerical model. The Delta Method could be used to this end.
For a more accurate image reconstruction, it has to be determined how the current algorithm can be changed to take information about the measurement errors into account. The Maximum-Likelihood Estimation (especially the log-likelihood function) and the Least-Squares Estimation could be deployed to obtain more insight in the possible improvements.

1.2. STRUCTURE OF THE REPORT

Firstly, an introduction to Computer Tomography is given in Chapter 2. We will give some background information about Computer Tomography and the numerical model that is currently used. Also some remarks are made on measurement errors, linking the numerical model to the stochastic model.

Subsequently, a stochastic model for attenuation will be derived in Chapter 3. This model will be built up from scratch and will give rise to the use of the path dependent variance of the measurements. We will also compare this result with the current literature.
To link the stochastic model to the numerical model as seen in Chapter 2, a transformation of the data is needed. What the consequences of this transformation are for the distribution of the measurements will become clear by using the Delta Method. At the end of this chapter, we will make some remarks about variance stabilization.

In Chapter 4 the log-likelihood of the attenuation coefficients will be derived. The relation between the Maximum-Likelihood and Least-Squares Estimation will be illustrated under different assumptions for the variance of the measurements.

Next, a brief description is given in Chapter 5 of the numerical methods we will use. We will explain what kind of problem we are dealing with and why we use certain functions to solve this problem. Finally we will give a description of the algorithms we have used for the image reconstruction.

Thereafter, we will look in Chapter 6 at the results of the image reconstruction and we will compare the results with these of the current reconstruction algorithm.

Finally, conclusions are drawn and stated in Chapter 7. A summary of all results is given to get a overview of this bachelor project. In Chapter 8 we will argue which improvements could have been made, but were too much for the scope of this project.

1.3. NOTATION

For clarification, a list of mathematical notations used in this report can be found in Table 1.1.

Mathematical Object	Notation	Example
Vector	Bold or bold Greek	$\mathbf{x}$, $\boldsymbol{\theta}$
Matrix	Bold capital	$\mathbf{A}$
Transpose	Superscript T	$\mathbf{A}^T$
Random variable	Capital	Y_b, S_b
Observed value (of a random variable)	Lowercase	y_b, s_b
Absolute value	Single vertical line	$\lvert\cdot\rvert$
Euclidean norm (2-norm)	Double vertical line	$\lVert\cdot\rVert$

Table 1.1: Notations used in report.

An exception to this notation is made for the measurement errors ϵ_i in the numerical model. These ϵ_i's are random variables, but will not be written as uppercase.

2

What is Computer Tomography?

2.1. Introduction to Computer Tomography

Computer Tomography (CT) is an imaging method that uses X-rays to create tomographic images reflecting the amount of X-ray attenuation in some part of the human body. To this end the body part of interest is divided into a large number of transverse sections and for each section a two-dimensional image of the X-ray attenuation is constructed. Combining all these two-dimensional images belonging to the transverse sections gives a three-dimensional representation of the X-ray attenuation of the body part and hence of the soft tissue structure.

A series of X-ray attenuation measurements is obtained at different angles around the body part of interest, by means of radiation detectors. From these data, the two-dimensional CT-images can be reconstructed by a computer algorithm. These measurements can contain noise and this should be taken into account in the image reconstruction. [Ter-Pogossian, 1977]

2.2. Transmission Computed Tomography

In transmission computed tomography, a beam of X-ray radiation passes through a patient and is detected on the other side. The measurements are obtained at different angles and can be used to construct a CT-image. This sectional image therefore represents the distribution of the attenuation of X-rays in the examined tissue.

An X-ray tube of conventional design is energized and thus produces multiple beams of radiation. This fan-shaped collection of X-ray beams passes through the patient and is detected by a scintillation detector. The latter is connected to an electronic circuit. The X-ray tube and the detector are always placed opposite of each other and thus the body part of interest is scanned in a linear translational motion, see Figure 2.1. After one scan, the X-ray tube and the detector are rotated by 1° about an axis perpendicular to the section to be imaged, and another scan is performed. Typically, this procedure is repeated 180 times such that data is collected for a total of 180°. The measured data consists of a series of profiles of the attenuation of X-rays in the tissue traversed at the 180 different angles. From these profiles a CT-image can be reconstructed by a computer algorithm.

Transmission CT offers great advantages in diagnostic radiology: not only offers CT the possibility to construct a three-dimensional image for radiological examinations, but it also provides a quantitatively accurate distribution of X-ray attenuation in the section imaged. [Ter-Pogossian, 1977]

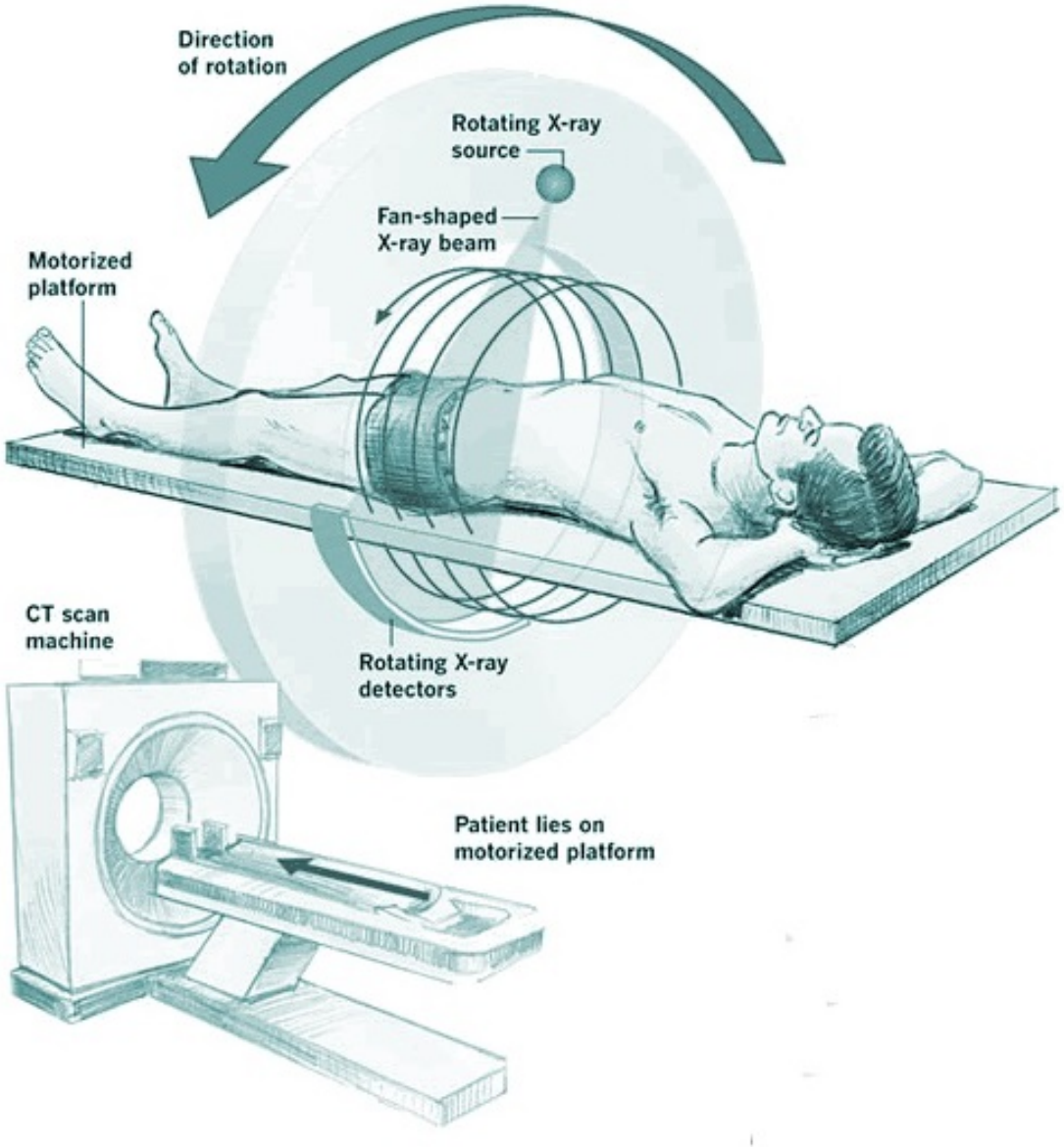
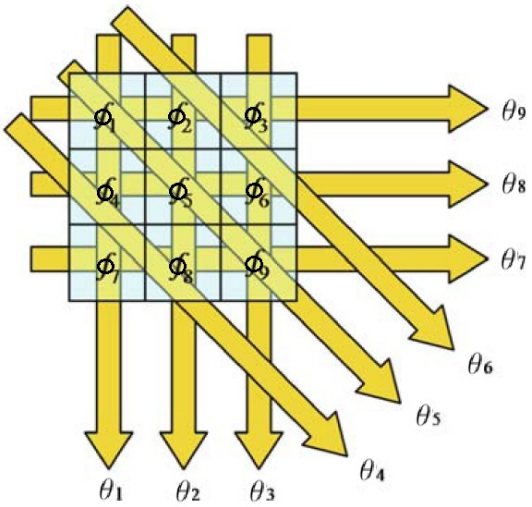


Figure 2.1: Schematic representation of a transmission CT-scan. [CyberPhysics]

2.3. NUMERICAL MODEL

To work with the obtained measurements of the X-ray attenuations, the following numerical model is used:

First of all, a discretization is made of the sectional image where the fan-shaped collection of X-ray beams is passing through. The measurements, θ_i, with $i = \{1,2,\ldots,M\}$, are discretized due to the design of the detector array based on a set of discrete detector elements. The tomographic image that has to be reconstructed consists of a discrete array of unknown variables, ϕ_j, with $j = \{1,2,\ldots,N\}$, i.e., the unknown attenuation coefficients. This leads to the following situation for $N = 9$ and motivates the set-up of a corresponding system of linear equations:



$$\begin{aligned}
\phi_1 + \phi_4 + \phi_7 &= \theta_1 \\
\phi_2 + \phi_5 + \phi_8 &= \theta_2 \\
\phi_3 + \phi_6 + \phi_9 &= \theta_3 \\
\phi_4 + \phi_7 + \phi_8 &= \theta_4 \\
\phi_1 + \phi_5 + \phi_9 &= \theta_5 \\
\phi_2 + \phi_3 + \phi_6 &= \theta_6 \\
\phi_7 + \phi_8 + \phi_9 &= \theta_7 \\
\phi_4 + \phi_5 + \phi_6 &= \theta_8 \\
\phi_1 + \phi_2 + \phi_3 &= \theta_9
\end{aligned}$$

Figure 2.2: A grid for algebraic reconstruction, with $N = 9$ pixels. If the grid that is to be reconstructed is finer, more projections have to be measured. [Buzug, 2008]

Looking at the diagonal travelling rays, a difference is apparent compared with the horizontal and vertical travelling rays: the path length through each element of the object is obviously different. This has to be taken into account in the linear system of equations, so a system of linear equations as stated in Figure 2.2 is not accurate.

Assume that the transmitted X-ray beam has a certain width $\Delta\xi$. Only a part of each pixel that has to be reconstructed is passed through by the beam. Therefore, weights a_{ij} have to be introduced that reflect the relation between the area that is illuminated by the beam and the entire area of the pixel:

$$a_{ij} = \frac{\text{illuminated area of pixel } j \text{ by ray } i}{\text{total area of pixel } j}$$

Clearly, $0 \leq a_{ij} \leq 1$. The next figure is added to clarify the introduced variables:

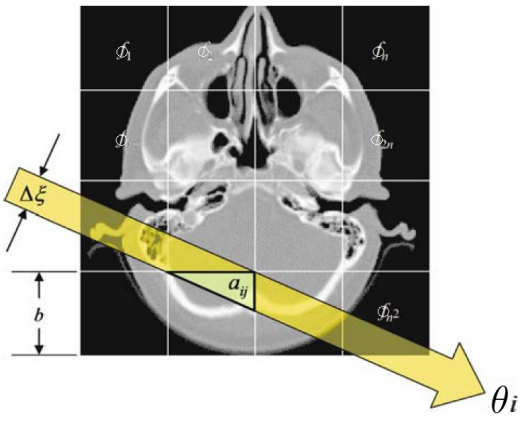


Figure 2.3: The X-ray beam of width $\Delta\xi$ does not traverse all pixels of size b^2 equally. Therefore, the area of the pixel that has actually been passed through must be included into the linear system of equations as a weight. [Buzug, 2008]

This leads to the following linear system of equations:

$$\sum_{j=1}^{N} a_{ij}\phi_j = \theta_i \qquad \text{for all X-ray beams } i \in \{1,\dots,M\}$$

Writing all measurements as column vector

$$\boldsymbol{\theta} = (\theta_1,\dots,\theta_M)^T,$$

writing the (unknown) attenuation values that are to be reconstructed as a column vector as well

$$\boldsymbol{\phi} = (\phi_1,\dots,\phi_N)^T,$$

and collecting the weights in an $M \times N$ matrix

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{pmatrix},$$

the system of equations becomes

$$\mathbf{A}\boldsymbol{\phi} = \boldsymbol{\theta}$$

where $\mathbf{A}$ in CT often is referred to as the system matrix. $\mathbf{A}$ is normally large, sparse and not square. [Buzug, 2008]

2.3.1. MEASUREMENT ERRORS

The measurements, θ_i, with $i = \{1,2,\dots,M\}$, consist of the loss of intensity of the X-rays recorded by the detector. Let I_0 denote the source intensity, let I_i denote the intensity of the ith X-ray after having passed through the body part of interest, and let $\phi(\mathbf{x})$ denote the linear attenuation coefficient at $\mathbf{x} \in \mathbb{R}^2$.
Then [Hansen and Saxild-Hansen, 2012] claims that the line integral of $\phi(\mathbf{x})$ along the jth ray satisfies

$$\theta_i = \int_{\text{ray } i} \phi(\mathbf{x})\, \mathrm{d}\ell = \log\left\{\frac{I_0}{I_i}\right\} \qquad \text{for } i = \{1,2,\dots,M\}.$$

However, these measurements are not exact and can obtain some noise which is not taken into account. Hence, a more realistic model of the measurements could be

$$\theta_i + \epsilon_i = \int_{\text{ray } i} \phi(\mathbf{x})\, \mathrm{d}\ell + \epsilon_i = \log\left\{\frac{I_0}{I_i}\right\} \qquad \text{for } i = \{1,2,\dots,M\},$$

where ϵ_i represents the measurement error in the ith measurement.

Using the discretization described above, we obtain for the ith X-ray

$$\theta_i + \epsilon_i = \int_{\text{ray } i} \phi(\mathbf{x})\, \mathrm{d}\ell + \epsilon_i \approx \sum_{j=1}^{N} a_{ij}\phi_j + \epsilon_i = \log\left\{\frac{I_0}{I_i}\right\} \qquad \text{for } i = \{1,2,\dots,M\}.$$

Defining $\psi_i = \log\left\{\frac{I_0}{I_i}\right\}$ for $i = \{1,2,\dots,M\}$ and combining all M measurements, the linear system of equations becomes

$$\mathbf{A}\boldsymbol{\phi} + \boldsymbol{\epsilon} = \boldsymbol{\psi}.$$

The question, now, arises: if more information about $\boldsymbol{\epsilon}$ is available, could this improve the image reconstruction in the presence of noise? Therefore, a stochastic model for attenuation will be constructed in the next chapter.

3

STOCHASTIC MODEL FOR ATTENUATION

3.1. PROBLEM

During a CT-scan the transmitted X-rays will be attenuated, due to the density of the tissue they are travelling through. This attenuation is desirable, because it gives certain information about the path the X-rays have traveled. But the attenuation can also contribute to noise in the model. When an X-ray beam is transmitted, the X-ray photons decay with a certain probability in the traversed tissue and hence the measurements can contain some noise.

In numerical mathematics this noise is assumed to be standard normally distributed, but the reason why this assumption is made is questionable. In this chapter we will construct a stochastic model for the attenuation and hence for the noise.

3.2. FREQUENTLY USED VARIABLES AND TERMS

In this chapter, lots of variables are used in combination with terminology that might not be intuitively clear to the reader. Therefore, the most frequently used variables and terms are listed in this section.

Term	Corresponding variable	Explanation
Attenuation	-	Loss of intensity of an X-ray beam.
Attenuation coefficient	ϕ_j	Discretization of the attenuation function $\phi(\mathbf{x})$, with $j = \{1, 2, \ldots, N\}$.
Attenuation function	$\phi(\mathbf{x})$	Function for $\mathbf{x} \in \mathbb{R}^2$, defining the attenuation in each position $\mathbf{x}$.
Bernoulli distribution	Bernoulli (p)	Discrete probability distribution of a random variable which takes value 1 with probability p and value 0 with probability $1 - p$.
Binomial distribution	Binomial (n, p)	Discrete probability distribution with n independent trials, each of which yielding success with probability p.
Box	-	Grid used to discretize the density function $\phi_b(x)$
Cell	-	Grid used to discretize the attenuation function $\phi(\mathbf{x})$.
Density function	$\phi_b(x)$	Function for $x \in \mathbb{R}$ on the path that the X-ray photons in beam b travel, defining the attenuation in x for beam $b = \{1, 2, \ldots, M\}$.

Normal distribution	Normal (μ, σ^2)	Continuous probability distribution of a random variable with expected value μ and variance σ^2.
X-ray beam	b	Transmitted beam of X-rays over one angle.
X-ray photon	j	Elementary particle in an X-ray.

Table 3.1: Frequently used terms with their corresponding variable and explanation.

Variable	Explanation
M	Total number of X-ray beams transmitted during a CT-scan.
m	Number of X-ray photons in each beam $b = \{1,2,\ldots,M\}$. Because I_0 is constant for each beam b, we assume that m is also the same in each beam.
N	Number of cells used to discretize the attenuation function $\phi(\mathbf{x})$.
n_b	Number of boxes used to discretize the path that the X-ray photons in beam b travel.
l_b	Length of the path that the X-ray photons in beam b travel. Because we are dealing with fan-beam tomography, the path length is different for each beam b.
p_i	Probability that an X-ray photon survives in box i.
D_j	Random variable; detection variable for X-ray photon j.
I_b	Measured intensity of the transmitted X-ray beam b after having passed through the body part of interest.
I_0	Source intensity of the transmitted X-ray beams $b = \{1,2,\ldots,M\}$.
S_b	Random variable; number of detections from beam b.
s_b	Observed value of the random variable S_b.
Y_b	Random variable; proportion of detected photons in beam b.
y_b	Observed value of the random variable Y_b.
θ_b	The line integral $\int_0^{l_b} \phi_b(x)\,\mathrm{d}x$ for beam b (exact measurement).
ψ_b	Measurement of $\log\left\{\frac{I_0}{I_b}\right\}$ for beam b (measurement with noise).

Table 3.2: Frequently used variables with their explanation.

3.3. DERIVATION OF THE MODEL

Consider an X-ray beam b, with $b = \{1,2,\ldots,M\}$. First of all, the path that the X-ray photons in beam b travel is discretized by a grid of n_b boxes. Note that this is *not* the grid used to discretize the attenuation coefficients as in Figure 2.3. As a start only one X-ray photon will be considered. Figure 3.1 illustrates the idea. To obtain a realistic model, the number of boxes n_b should be large. Also, because an X-ray photon is incredibly small, the illustrated square boxes will be small and could almost be seen as a line.
Let p_i be the probability that an X-ray photon survives in box i, with $i = \{1,2,\ldots,n_b\}$. Then the probability of decay of the X-ray photon in box i is $1 - p_i$. This decay will be called an attenuation event. Let l_b be the length of the path that the X-ray photon in beam b travels (and thus the length of all boxes together). Assume that all the attenuation events are independent.

What is the probability that an X-ray photon does *not* decay anywhere on its path and thus will be detected?

Define the detection variable D_j as:

$$D_j = \begin{cases} 1 & \text{if detection takes place of the } j\text{th X-ray photon} \\ 0 & \text{otherwise} \end{cases}$$

Thus the probability that the jth X-ray photon in beam b will be detected is:

$$P(D_j = 1) = p_1 \cdot p_2 \cdots p_{n_b} = \prod_{i=1}^{n_b} p_i$$

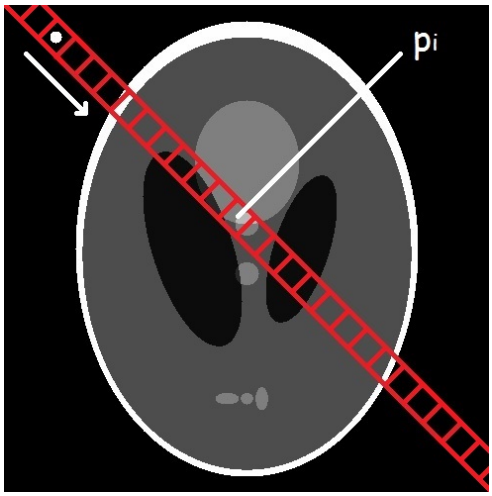


Figure 3.1: Discretisation of a path travelled by a transmitted X-ray photon.

How can the probability p_i be defined? p_i is dependent on the length of each box it travels and of the density of the tissue in these boxes.

Let $\phi : S \to \mathbb{R}$ be the attenuation function of the tissue, where $S \subset \mathbb{R}^2$ is the whole two-dimensional section of the body part of interest. Let $\phi_b : [0, l_b] \to \mathbb{R}$ be the density function of the tissue traversed by an X-ray photon from beam b. To make a connection between these two functions, we have to parametrize beam b. Let $\mathbf{x_{start}} \in \mathbb{R}^2$ be the first point on the path that an X-ray photon in beam b travels and let $\mathbf{x_{end}} \in \mathbb{R}^2$ be the last point on the path. Let $\mathbf{x} \in \mathbb{R}^2$ be an arbitrary point on the path that an X-ray photon in beam b travels. Then the corresponding parametrisation is:

$$\mathbf{x} = \mathbf{x_{start}} + \frac{\lambda}{l_b}(\mathbf{x_{end}} - \mathbf{x_{start}}) \qquad \text{for some } \lambda \in [0, l_b].$$

Hence, a connection can be made between the functions ϕ and ϕ_b:

$$\phi_b(\lambda) = \phi\left(\mathbf{x_{start}} + \frac{\lambda}{l_b}(\mathbf{x_{end}} - \mathbf{x_{start}})\right)$$

A large value of $\phi_b(x)$ should indicate a large density at x. Assume that $\phi_b(x)$ is constant in each box for a grid small enough, so for each x in box i the value of $\phi_b(x)$ is the same.

To illustrate these definitions, take a look at Figure 3.1. $\phi(\mathbf{x})$ is defined over the whole picture, while $\phi_b(x)$ is only defined over the red grid. For each x in a box of the red grid, the value of $\phi_b(x)$ is constant. For the rest of this section, we will look at $\phi_b(x)$ only as a function in $\mathbb{R}$ over a line, but keep in mind what its connection is with $\phi(\mathbf{x})$.
We assume that the probability of attenuation is proportional to the density. The larger $\phi_b(x)$ is (and hence the attenuation coefficient) in a certain box, the smaller the probability that an X-ray photon will survive in this box.

This leads to the following definition of p_i:

$$p_i = 1 - \frac{l_b}{n_b}\,\phi_b\left(\frac{i\,l_b}{n_b}\right) \qquad \text{for an X-ray photon in beam } b.$$

Here, $\frac{l_b}{n_b}$ is the length of the path that the X-ray photon travels in each box i, with $i = \{1, 2, \ldots, n_b\}$. The argument $\frac{i\,l_b}{n_b}$ is used such that in each box i the function $\phi_b(x)$ is considered.

The probability that the jth X-ray photon in beam b will be detected is:

$$P(D_j = 1) = \prod_{i=1}^{n_b} p_i = \prod_{i=1}^{n_b}\left(1 - \frac{l_b}{n_b}\,\phi_b\left(\frac{i\,l_b}{n_b}\right)\right)$$

Taking the logarithm of this expression results in:

$$\begin{aligned}\log\{\,P(D_j=1)\,\} &= \log\left\{\prod_{i=1}^{n_b}\left(1-\frac{l_b}{n_b}\,\phi_b\left(\frac{i\,l_b}{n_b}\right)\right)\right\}\\ &= \sum_{i=1}^{n_b}\log\left\{1-\frac{l_b}{n_b}\,\phi_b\left(\frac{i\,l_b}{n_b}\right)\right\}\\ &\approx \sum_{i=1}^{n_b}-\frac{l_b}{n_b}\,\phi_b\left(\frac{i\,l_b}{n_b}\right)\\ &= -\int_0^{l_b}\phi_b(x)\,\mathrm{d}x\end{aligned}$$

We will give a brief explanation of the steps we use in this derivation.

Recall the Taylor expansion: $\log\{1+x\} = x-\frac{x^2}{2}+\mathcal{O}(x^3)$, such that $\log\{1+x\}-x=\mathcal{O}(x^2)$. Hence, there is a $C\in\mathbb{R}$ such that

$$\left|\,\log\{1+x\}-x\,\right| \le C\left|x^2\right| = Cx^2 \qquad \text{for } x>-1.$$

Define $x_i = -l_b\,\phi\left(\frac{i\,l_b}{n_b}\right)$, then

$$\left|\sum_{i=1}^{n_b}\left(\log\left\{1+\frac{x_i}{n_b}\right\}-\frac{x_i}{n_b}\right)\right| \le \sum_{i=1}^{n_b}\left|\,\log\left\{1+\frac{x_i}{n_b}\right\}-\frac{x_i}{n_b}\,\right| \le \sum_{i=1}^{n_b}C\left(\frac{x_i}{n_b}\right)^2 = \frac{C}{n_b^2}\sum_{i=1}^{n_b}x_i^2.$$

This means that, because n_b is large, the approximation in the second to last step of the derivation is permitted. Note that in the last step the resulting summation defines a Riemann sum.

Because $\log\{\,P(D_j=1)\,\}\approx-\int_0^{l_b}\phi_b(x)\,\mathrm{d}x$ we choose as a model

$$P(D_j=1) = \exp\left[-\int_0^{l_b}\phi_b(x)\,\mathrm{d}x\right]$$

and this gives us the desired probability of survival for the jth X-ray photon in beam b.

Of course we are dealing with more than one X-ray photon during a CT-scan. What can we say about the distribution of a large number of X-ray photons?

During a CT-scan, multiple X-ray beams are transmitted and detected on the opposite site of the body part of interest, where the measurements take place. Assume that m X-ray photons are transmitted in beam b, with m large. Then the number of detections S_b resulting from the measurement of beam b is equal to $S_b=\sum_{j=1}^{m}D_j$.

Recall that the attenuation events are independent for each photon and

$$D_j \sim \text{Bernoulli}\left(\exp\left[-\int_0^{l_b}\phi_b(x)\,\mathrm{d}x\right]\right).$$

Hence,

$$S_b \sim \text{Binomial}\left(m,\,\exp\left[-\int_0^{l_b}\phi_b(x)\,\mathrm{d}x\right]\right).$$

Because we are dealing with a large number m of X-ray photons in beam b, we have approximately, by the Central Limit Theorem:

$$S_b \sim \text{Normal}\left(mp_b,\,mp_b(1-p_b)\right) \qquad \text{with } p_b=\exp\left[-\int_0^{l_b}\phi_b(x)\,\mathrm{d}x\right]. \tag{3.1}$$

We can define the random variable $Y_b=\frac{S_b}{m}$ as the proportion of detected photons in beam b. Because of (3.1), we have

$$Y_b=\frac{S_b}{m}\sim\text{Normal}\left(p_b,\,\frac{1}{m}\,p_b(1-p_b)\right) \qquad \text{with } p_b=\exp\left[-\int_0^{l_b}\phi_b(x)\,\mathrm{d}x\right] \text{ in beam } b.$$

Note that, for large m, the variance of Y_b becomes small which is beneficial for the reconstruction. The measured proportion of photons that survived in beam b will be close to the underlying probability p_b.

To obtain enough information to reconstruct a CT-image, multiple X-ray beams are transmitted with different angles towards the body part of interest. Assume that the number of X-ray beams transmitted is equal to M, so there are $b = \{1,2,\ldots,M\}$ beams. The random variables S_b are independent for each beam b, but not identically distributed. The parameter p_b for each beam is path dependent and hence different for all the M beams transmitted.
Each measurement y_b, with $b = \{1,2,\ldots,M\}$ gives, with a certain variance, a value for $p_b = \exp\left[-\int_0^{l_b} \phi_b(x)\,\mathrm{d}x\right]$.
We are interested in the true values of $\int_0^{l_b} \phi_b(x)\,\mathrm{d}x$ for each beam $b = \{1,2,\ldots,M\}$.

Discretizing $\phi(\mathbf{x})$ over N cells as described in Section 2.3 results in a discrete array of unknown variables, ϕ_j, with $j = \{1,2,\ldots,N\}$. Properly scaled, the ϕ_j's can be seen as attenuation coefficients. Figure 3.2 illustrates this idea for $N = 16$. Here, an X-ray beam is drawn as a red line (compare with Figure 3.1).
From this discretization of $\phi(\mathbf{x})$ it is possible to write $\theta_b = \int_0^{l_b} \phi_b(x)\,\mathrm{d}x \approx \sum_{j=1}^{N} a_{bj}\phi_j$, where a_{bj} denotes the length of the path that an X-ray photon from beam b travels through the cell of the discretized attenuation coefficient ϕ_j.
Combining these M line integrals results in an estimation for the unknown attenuation coefficients ϕ_j for $j = \{1,2,\ldots,N\}$.

Figure 3.2: Discretisation of $\phi(\mathbf{x})$ of the body part of interest, for $N = 16$

Our problem now reduces to estimating the unknown $\theta_b = \int_0^{l_b} \phi_b(x)\,\mathrm{d}x$ for each beam b.

One way of finding a suitable estimation for θ_b is the use of the method of Maximum-Likelihood Estimation (MLE). These results can be found in Appendix A.

3.4. DIFFERENCE WITH CURRENT LITERATURE

Let I_0 denote the source intensity of the transmitted X-ray beam b and let I_b denote the intensity of the beam after having passed through the body part of interest. Then [Hansen and Saxild-Hansen, 2012] claims that:

$$\int_0^{l_b} \phi_b(x)\,\mathrm{d}x = \log\left\{\frac{I_0}{I_b}\right\} \iff \frac{I_b}{I_0} = \exp\left[-\int_0^{l_b} \phi_b(x)\,\mathrm{d}x\right] = p_b \tag{3.2}$$

Because $S_b \sim \text{Binomial}(m, p_b)$, the Maximum-Likelihood Estimator for p_b based on one observation of the proportion of detected photons in beam b is $\hat{p}_b = \frac{S_b}{m} = Y_b$. It turns out that in CT-scans not the number of detected photons is measured, but only the percentual loss of intensity of the X-rays.

From [Waseda et al., 2011] it is known that each X-ray photon has an energy E, which is proportional to its frequency,

$$E = h\nu$$

where h is the Planck constant ($6.6260 \times 10^{-34} J \cdot s$) and ν is the frequency of the photon. Furthermore, intensity is equal to energy per unit time per unit area and we assume that all transmitted X-ray photons have the same frequency. Hence, the intensity is directly proportional to the number of X-ray photons. Therefore, $\frac{I_b}{I_0}$ is equal to the proportion of detected photons in beam b and we can take $\hat{p}_b = y_b = \frac{I_b}{I_0}$ as Maximum-Likelihood Estimation for p_b.

If it is possible to find out what the number of photons m is in the transmitted X-ray beam with source intensity I_0, then $\hat{p}_b$ satisfies approximately

$$\hat{p}_b = Y_b \sim \text{Normal}\left(p_b,\ \frac{1}{m}\, p_b(1-p_b)\right) \qquad \text{with } p_b = \exp\left[-\int_0^{l_b} \phi_b(x)\,\mathrm{d}x\right] \text{ in beam } b.$$

Hence, approximately

$$\hat{p}_b - p_b = Y_b - p_b \sim \text{Normal}\left(0,\ \frac{1}{m}\,\hat{p}_b(1-\hat{p}_b)\right) = \text{Normal}\left(0,\ \frac{I_b}{mI_0^2}\,(I_0 - I_b)\right)$$

where we use the estimation $\hat{p}_b$ to compute the variance.

Determining m from I_0 can be hard, because the constant of proportionality of the intensity and number of photons is unknown. In that case we can still use our model by stating that the 'noise' is approximately

$$\hat{p}_b - p_b \sim \text{Normal}\left(0,\ c\,\frac{I_b}{I_0^2}\,(I_0 - I_b)\right) \qquad \text{with } c \text{ unknown}$$

Note that for each scan in a different beam $\hat{p}_b$ and p_b are different, but the noise can still be modeled as stated above.

This consideration leads to a refinement of (3.2) which only approximately holds, where the noise is modeled. For larger m, the approximation gets better, but the equality still does not hold.

3.5. DELTA METHOD

Define the random variable E_b, the error between Y_b and the unknown p_b, as

$$E_b = Y_b - p_b = Y_b - \exp\left[-\int_0^{l_b} \phi_b(x)\,\mathrm{d}x\right]. \tag{3.3}$$

In the previous sections, it was found that approximately

$$E_b \sim \text{Normal}\left(0,\ \frac{1}{m}\,\hat{p}_b(1-\hat{p}_b)\right). \tag{3.4}$$

For the numerical analysis, a system of equations of the form

$$\mathbf{A}\boldsymbol{\phi} + \boldsymbol{\epsilon} = \boldsymbol{\psi} \tag{3.5}$$

is desired, as seen in Section 2.3.1.

How can we use (3.4) to determine the distribution of the ϵ_i, for $i = \{1,2,\ldots,M\}$, in equation (3.5)?

Recall from Section 2.3.1 that

$$\theta_b + \epsilon_b = \int_0^{l_b} \phi_b(x)\,\mathrm{d}x + \epsilon_b = \log\left\{\frac{I_0}{I_b}\right\} = -\log\{Y_b\} = \psi_b \qquad \text{for } b = \{1,2,\ldots,M\},$$

and hence

$$\epsilon_b = \psi_b - \theta_b = -\log\{Y_b\} - \int_0^{l_b} \phi_b(x)\,\mathrm{d}x \qquad \text{for } b = \{1,2,\ldots,M\}.$$

We are interested in the distribution of ϵ_b, but there is only information available about E_b. The following transformation is needed:

$$\begin{aligned}\epsilon_b &= -\log\{Y_b\} - \int_0^{l_b} \phi_b(x)\,\mathrm{d}x \\ &= -\log\{Y_b\} - \left(-\log\left\{\exp\left[-\int_0^{l_b}\phi_b(x)\,\mathrm{d}x\right]\right\}\right) \\ &= -\log\{Y_b\} - \left(-\log\{p_b\}\right)\end{aligned}$$

In [Papanicolaou, 2009] the following theorem is found, which is known as the 'Univariate Delta Method':

Theorem 1. *Let Y_n be a sequence of random variables that satisfies $\sqrt{n}(Y_n-\theta) \to Normal\left(0,\ \sigma^2\right)$ in distribution. For a given function and a specific value of θ, suppose that $g'(\theta)$ exists and is not 0. Then,*

$$\sqrt{n}\left(g(Y_n) - g(\theta)\right) \to Normal\left(0,\ \sigma^2 g'(\theta)^2\right) \text{ in distribution}$$

Define $g(x) = -\log(x)$, then $g'(x) = -\frac{1}{x}$. For ϵ_b it is possible to write

$$\epsilon_b = -\log\{Y_b\} - \left(-\log\{p_b\}\right) = g(Y_b) - g(p_b) \qquad \text{for } b = \{1,2,\ldots,M\}$$

and from Theorem 3.5 with $n = 1$ and equation (3.4) it is found that

$$\epsilon_b \sim \text{Normal}\left(0,\ \frac{1}{m}\frac{1-\hat{p}_b}{\hat{p}_b}\right)$$

which gives the desired distribution of the ϵ_b for $b = \{1,2,\ldots,M\}$.

3.6. VARIANCE STABILIZATION

In the previous section, the Delta Method was used to determine the distribution of the errors ϵ_b for $b = \{1,2,\ldots,M\}$ in the linear system of equations $\mathbf{A}\boldsymbol{\phi} + \boldsymbol{\epsilon} = \boldsymbol{\psi}$. But this method can also be used for another important purpose.
In some situations, it might be desirable to have errors with a constant variance. In this case the Delta Method can be used to transform the data such that errors with the same constant variance remain.

Let E_b, the error between Y_b and the unknown p_b, be defined again as in (3.3) such that

$$E_b \sim \text{Normal}\left(0,\ \frac{1}{m}\,\hat{p}_b(1-\hat{p}_b)\right).$$

Suppose that we want to obtain errors ϵ_b, with $b = \{1,2,\ldots,M\}$, that all have the same constant variance $\frac{1}{m}$. To use the Delta Method for this purpose, a function $h(x)$ has to be found such that

$$h'(x) = \frac{1}{\sqrt{x(1-x)}}$$

because this results in

$$\sigma^2 h'(\hat{p}_b)^2 = \frac{1}{m}\,\hat{p}_b(1-\hat{p}_b)\left(\frac{1}{\sqrt{\hat{p}_b(1-\hat{p}_b)}}\right)^2 = \frac{1}{m}$$

After some calculus, the desired function $h(x)$ was found:

$$h(x) = \arcsin(2x-1)$$

and note that this function is defined on the interval $[0,1]$ as desired. This function $h(x)$ that can be used to transform the data such that errors with constant variance remain is called the *variance-stabilizing transformation.*

Applying this variance-stabilizing transformation to the data results in

$$\epsilon_b = h(Y_b) - h(p_b) = \arcsin(2Y_b - 1) - \arcsin(2p_b - 1) \qquad \text{for } b = \{1, 2, \ldots, M\}.$$

From Theorem 3.5 with $n = 1$ and equation (3.3) it follows that

$$\epsilon_b \sim \text{Normal}\left(0,\ \frac{1}{m}\right)$$

and, because m is the same for each beam b, errors ϵ_b, with $b = \{1, 2, \ldots, M\}$, are obtained that all have the same constant variance.

4

LOG-LIKELIHOOD AND LEAST-SQUARES ESTIMATION

Recall the normal distribution: this continuous distribution has probability density function

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

on the real line. Here, μ represents the mean or expected value of a random variable and σ^2 denotes the variance.
Let $y_b = \frac{I_b}{I_0}$, with $b = \{1, 2, \ldots, M\}$, be the measurements of a CT-scan.

In Section 3.3 the model

$$Y_b \sim \text{Normal}\left(p_b,\ \frac{1}{m}\, p_b(1-p_b)\right) \qquad \text{with } p_b = \exp\left[-\int_0^{l_b} \phi_b(x)\, \mathrm{d}x\right] \text{ in beam } b$$

was derived. Using the discretisation described in Section 2.3, it is possible to write

$$\int_0^{l_b} \phi_b(x)\, \mathrm{d}x \approx \sum_{j=1}^{N} a_{bj}\phi_j \qquad \text{in beam } b,$$

where ϕ_j, with $j = \{1, 2, \ldots, N\}$, is the discretisation of the attenuation function $\phi(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^2$ (see Table 3.1). Hence, $Y_b \sim \text{Normal}\left(\mu_b, \sigma_b^2\right)$ with

$$\mu_b = \mu_b(\boldsymbol{\phi}) = \exp\left[-\sum_{j=1}^{N} a_{bj}\phi_j\right] \tag{4.1}$$

$$\sigma_b^2 = \sigma_b(\boldsymbol{\phi})^2 = \frac{1}{m} \exp\left[-\sum_{j=1}^{N} a_{bj}\phi_j\right]\left(1 - \exp\left[-\sum_{j=1}^{N} a_{bj}\phi_j\right]\right) \tag{4.2}$$

In this chapter, the relation between the log-likelihood and the Least-Squares Estimation (LSE) will be illustrated.

4.1. LOG-LIKELIHOOD OF $\boldsymbol{\phi}$

For a Maximum-Likelihood Estimation, the log-likelihood of $\boldsymbol{\phi}$ is maximized over the parameter space. Because Maximum-Likelihood Estimation and Least-Squares Estimation are closely related, it is useful to determine the log-likelihood of $\boldsymbol{\phi}$ and to check if the value of the log-likelihood increases with better Least-Squares Estimations of $\boldsymbol{\phi}$. Ideally, the log-likelihood of $\boldsymbol{\phi}$ would increase until it reaches its maximum value: then the Maximum-Likelihood Estimation is obtained.

Let $\boldsymbol{\phi} = (\phi_1, \phi_2, \ldots, \phi_N)^T$ be a vector that contains the unknown attenuation coefficients. Then the corresponding likelihood of $\boldsymbol{\phi}$ is equal to

$$\begin{aligned}\mathscr{L}(\boldsymbol{\phi};\ y_1, y_2, \ldots, y_M) &= f(y_1, y_2, \ldots, y_M \mid \boldsymbol{\phi}) = \prod_{i=1}^{M} f(y_i \mid \boldsymbol{\phi}) \\ &= \prod_{i=1}^{M} \frac{1}{\sqrt{2\pi\sigma_i(\boldsymbol{\phi})^2}} \exp\left[-\frac{(y_i - \mu_i(\boldsymbol{\phi}))^2}{2\sigma_i(\boldsymbol{\phi})^2}\right]\end{aligned}$$

Taking the natural logarithm of the likelihood results in the log-likelihood

$$\begin{aligned}\ell(\boldsymbol{\phi};\ y_1, y_2, \ldots, y_M) &= \ln\{\mathscr{L}(\boldsymbol{\phi};\ y_1, y_2, \ldots, y_M)\} \\ &= \sum_{i=1}^{M}\left(\ln\left\{\frac{1}{\sqrt{2\pi\sigma_i(\boldsymbol{\phi})^2}}\right\} - \frac{(y_i - \mu_i(\boldsymbol{\phi}))^2}{2\sigma_i(\boldsymbol{\phi}^2)}\right) \\ &= \sum_{i=1}^{M}\left(-\frac{1}{2}\ln\{2\pi\sigma_i(\boldsymbol{\phi})^2\} - \frac{(y_i - \mu_i(\boldsymbol{\phi}))^2}{2\sigma_i(\boldsymbol{\phi})^2}\right)\end{aligned}$$

Substituting $\mu_i(\boldsymbol{\phi})$ and $\sigma_i(\boldsymbol{\phi})^2$ in the log-likelihood gives:

$$\begin{aligned}\ell(\boldsymbol{\phi};\ y_1, y_2, \ldots, y_M) = \sum_{i=1}^{M}\Bigg(&-\frac{1}{2}\ln\{2\pi\} - \frac{1}{2}\ln\left\{\frac{1}{m}\right\} + \frac{1}{2}\sum_{j=1}^{N} a_{ij}\phi_j - \frac{1}{2}\ln\left\{1 - \exp\left[-\sum_{j=1}^{N} a_{ij}\phi_j\right]\right\} \\ &- \frac{m\left(y_i - \exp\left[-\sum_{j=1}^{N} a_{ij}\phi_j\right]\right)^2}{2\exp\left[-\sum_{j=1}^{N} a_{ij}\phi_j\right] - 2\exp\left[-\sum_{j=1}^{N} a_{ij}\phi_j\right]}\Bigg)\end{aligned} \tag{4.3}$$

In the numerical analysis, the log-likelihood will be evaluated for the different values of $\boldsymbol{\phi}$. Comments on the Maximum-Likelihood Estimation can be found in Appendix A.

In the next sections, the log-likelihood of $\boldsymbol{\phi}$ will be derived under different assumptions for the mean and variance of the measurements. These different scenarios will be taken into account during the numerical analysis of the reconstruction.

4.2. CONSTANT VARIANCE σ^2

Suppose that $\sigma_b^2 = \sigma^2$ is constant, known and not path dependent for all beams $b = \{1, 2, \ldots, M\}$. Let $\mu_b = \mu_b(\boldsymbol{\phi})$ be path dependent for each beam b and defined as in (4.1). This situation would correspond to a constant $\boldsymbol{\phi}$ on a circular object that is concentric with the circular scanner. The corresponding likelihood of $\boldsymbol{\phi}$ is equal to:

$$\begin{aligned}\mathscr{L}(\boldsymbol{\phi}; y_1, y_2, \ldots, y_M) &= \prod_{i=1}^{M} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y_i - \mu_i(\boldsymbol{\phi}))^2}{2\sigma^2}\right] \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^M \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{M}(y_i - \mu_i(\boldsymbol{\phi}))^2\right]\end{aligned}$$

Taking the natural logarithm of the likelihood results in the log-likelihood of $\boldsymbol{\phi}$:

$$\begin{aligned}\ell(\boldsymbol{\phi}; y_1, y_2, \ldots, y_M) &= \ln\left\{\mathscr{L}(\boldsymbol{\phi}; y_1, y_2, \ldots, y_M)\right\} \\ &= -\frac{M}{2}\ln\{2\pi\} - \frac{M}{2}\ln\{\sigma^2\} - \frac{1}{2\sigma^2}\sum_{i=1}^{M}(y_i - \mu_i(\boldsymbol{\phi}))^2 \\ &= -\frac{M}{2}\ln\{2\pi\} - \frac{M}{2}\ln\{\sigma^2\} - \frac{1}{2\sigma^2}\sum_{i=1}^{M}\left(y_i - \exp\left[-\sum_{j=1}^{N} a_{ij}\phi_j\right]\right)^2\end{aligned}$$

For the Maximum-Likelihood Estimation the log-likelihood has to be maximized over $\boldsymbol{\phi}$. Because the first two terms in the log-likelihood of $\boldsymbol{\phi}$ are constant, this is equal to minimizing

$$\sum_{i=1}^{M}\left(y_i - \exp\left[-\sum_{j=1}^{N} a_{ij}\phi_j\right]\right)^2$$

which reduces the problem to a (non-linear) Least-Squares Estimation with constant weights.

But, from Section 3.3, it is known that the variance $\sigma_b(\boldsymbol{\phi})$ is different for each X-ray beam $b = \{1,2,\ldots,M\}$. In the next section, the non-constant variance is evaluated to see if this leads to a different (and hopefully better) Least-Squares Estimation.

4.3. NON-CONSTANT KNOWN VARIANCE σ_b^2

Suppose that σ_b^2 is non-constant but known in advance for all X-ray beams $b = \{1,2,\ldots,M\}$. The σ_b^2 could for example be chosen as the length of the path that the X-ray photons have travelled in beam b, giving a larger weight to the measurements obtained from the beams that form a shorter path through the body part of interest. Let $\mu_b = \mu_b(\boldsymbol{\phi})$ be path dependent for each X-ray beam b and defined as in (4.1). Then the corresponding likelihood of $\boldsymbol{\phi}$ is equal to:

$$\begin{aligned}\mathscr{L}(\boldsymbol{\phi}; y_1, y_2, \ldots, y_M) &= \prod_{i=1}^{M} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left[-\frac{(y_i - \mu_i(\boldsymbol{\phi}))^2}{2\sigma_i^2}\right] \\ &= \left(\frac{1}{\sqrt{2\pi}}\right)^M \left(\prod_{i=1}^{M} \frac{1}{\sqrt{\sigma_i^2}}\right) \exp\left[-\frac{1}{2}\sum_{i=1}^{M} \frac{(y_i - \mu_i(\boldsymbol{\phi}))^2}{\sigma_i^2}\right]\end{aligned}$$

Taking the natural logarithm of the likelihood results in the log-likelihood of $\boldsymbol{\phi}$:

$$\begin{aligned}\ell(\boldsymbol{\phi}; y_1, y_2, \ldots, y_M) &= \ln\left\{\mathscr{L}(\boldsymbol{\phi}; y_1, y_2, \ldots, y_M)\right\} \\ &= -\frac{M}{2}\ln\{2\pi\} - \frac{1}{2}\sum_{i=1}^{M}\left(\ln\{\sigma_i^2\} + \left(\frac{y_i - \mu_i(\boldsymbol{\phi})}{\sigma_i}\right)^2\right) \\ &= -\frac{M}{2}\ln\{2\pi\} - \frac{1}{2}\sum_{i=1}^{M}\left(\ln\{\sigma_i^2\} + \left(\frac{y_i - \exp\left[-\sum_{j=1}^{N} a_{ij}\phi_j\right]}{\sigma_i}\right)^2\right)\end{aligned}$$

Again, for the Maximum-Likelihood Estimation the log-likelihood has to be maximized over $\boldsymbol{\phi}$. Because the first term in the log-likelihood of $\boldsymbol{\phi}$ and also the first term of the summation is constant, this is equal to minimizing

$$\sum_{i=1}^{M}\left(\frac{y_i - \exp\left[-\sum_{j=1}^{N} a_{ij}\phi_j\right]}{\sigma_i}\right)^2$$

which reduces the problem to a weighted (non-linear) Least-Squares Estimation.

4.3.1. PATH DEPENDENT VARIANCE $\sigma_b(\boldsymbol{\phi_0})^2$

Suppose that $\sigma_b^2 = \sigma_b(\boldsymbol{\phi_0})^2$ is non-constant but known in advance, path dependent for each X-ray beam $b = \{1,2,\ldots,M\}$ and defined as in (4.2). We could, for example, use an initial estimation $\boldsymbol{\phi_0}$ for $\boldsymbol{\phi}$ and calculate $\sigma_b(\boldsymbol{\phi_0})$ by (4.2). Let $\mu_b = \mu_b(\boldsymbol{\phi})$ also be path dependent but unknown in advance for each X-ray beam b and defined as in (4.1).
Then the same likelihood and log-likelihood of $\boldsymbol{\phi}$ are found as in Section 4.3 and again, the Maximum-Likelihood Estimation is equal to the following weighted (non-linear) Least-Squares Estimation:

$$\sum_{i=1}^{M}\left(\frac{y_i - \exp\left[-\sum_{j=1}^{N} a_{ij}\phi_j\right]}{\sigma_i(\boldsymbol{\phi_0})}\right)^2$$

4.4. NON-CONSTANT UNKNOWN PATH DEPENDENT VARIANCE $\sigma_b(\boldsymbol{\phi})^2$

Suppose that $\sigma_b^2 = \sigma_b(\boldsymbol{\phi})^2$ is non-constant and unknown in advance, path dependent for each X-ray beam $b = \{1,2,\ldots,M\}$ and defined as in (4.2). Let $\mu_b = \mu_b(\boldsymbol{\phi})$ also be path dependent and unknown in advance for each X-ray beam b and defined as in (4.1).
Then the same likelihood and log-likelihood of $\boldsymbol{\phi}$ are found as in Section 4.3, but now we have to minimize:

$$\sum_{i=1}^{M}\left(\ln\{\sigma_i(\boldsymbol{\phi}\}+\left(\frac{y_i-\exp\left[-\sum_{j=1}^{N}a_{ij}\phi_j\right]}{\sigma_i(\boldsymbol{\phi})}\right)^2\right) \tag{4.4}$$

The most important difference here is that not a standard least-squares problem is found. In this case, $\boldsymbol{\phi}$ has to be estimated to minimize (4.4), where it is used in both the numerator and denominator and in the preceding term. This leads to a more complex optimization problem that we will not consider any further.

5

NUMERICAL METHODS

5.1. PROBLEM

In Computer Tomography, we want to find a solution $\mathbf{x}$ to the following system of linear equations:

$$\mathbf{Ax} = \mathbf{b}$$

Usually this problem does not have an unique solution. Hence, a numerical method is needed to compute a minimum norm solution $\mathbf{x}$ to the following linear least-squares problem:

$$\text{minimize } ||\mathbf{Ax} - \mathbf{b}||.$$

Here, $\mathbf{A}$ is a real matrix with M rows and N columns and $\mathbf{b}$ is a real vector of length M. These variables correspond with the problem from Section 2.3. $\mathbf{A}$ will normally be large and sparse.

5.2. THE MATLAB FUNCTION `LSQR`

In Matlab the function `lsqr` is used to find a minimum norm least-squares solution of the given problem. This function is based on a numerical method, to be called Algorithm LSQR in [Paige and Saunders, 1982], which uses a bidiagonalization procedure that will be stated in the next subsection. Because the derivation of this procedure is beyond the scope of this project, the interested reader is referred to [Golub and Kahan, 1965].

5.2.1. BIDIAGONALIZATION PROCEDURE

Let the matrix $\mathbf{A}$ and the starting vector $\mathbf{b}$ be given. The aim of the bidiagonalization procedure is to reduce the problem to lower bidiagonal form, i.e., to find matrices $\mathbf{U_k}$, $\mathbf{V_k}$ and $\mathbf{B_k}$ such that

$$\mathbf{AV_k} = \mathbf{U_{k+1}B_k} \qquad \text{with } \mathbf{B_k} = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix} \tag{5.1}$$

for some scalars α_i and β_i. In [Paige and Saunders, 1982], the bidiagonalization procedure is stated as

$$\left.\begin{aligned} \beta_1\mathbf{u_1} &= \mathbf{b} \\ \alpha_1\mathbf{v_1} &= \mathbf{A}^T\mathbf{u_1} \\ \beta_{i+1}\mathbf{u_{i+1}} &= \mathbf{Av_i} - \alpha_i\mathbf{u_i} \\ \alpha_{i+1}\mathbf{v_{i+1}} &= \mathbf{A}^T\mathbf{u_{i+1}} - \beta_{i+1}\mathbf{v_i} \end{aligned}\right\}, \quad i = 1,2,\ldots \tag{5.2}$$

Here, the scalars $\alpha_i \geq 0$ and $\beta_i \geq 0$ are chosen such that $||\mathbf{u_i}|| = ||\mathbf{v_i}|| = 1$. Rewriting the recurrence relation (5.2) gives

$$\left.\begin{aligned} \mathbf{Av_i} &= \alpha_i\mathbf{u_i} + \beta_{i+1}\mathbf{u_{i+1}} \\ \mathbf{A}^T\mathbf{u_{i+1}} &= \alpha_{i+1}\mathbf{v_{i+1}} + \beta_{i+1}\mathbf{v_i} \end{aligned}\right\}, \quad i = 1,2,\ldots \tag{5.3}$$

Defining $\mathbf{U_k} = (\mathbf{u_1}\ \mathbf{u_2}\ \cdots\ \mathbf{u_k})$, $\mathbf{V_k} = (\mathbf{v_1}\ \mathbf{v_2}\ \cdots\ \mathbf{v_k})$, and $\mathbf{B_k}$ as in (5.1), we see from (5.3) that we can write the recurrence relation as

$$\mathbf{U_{k+1}}(\beta_1 \mathbf{e_1}) = \mathbf{b} \tag{5.4}$$

$$\mathbf{AV_k} = \mathbf{U_{k+1}B_k} \tag{5.5}$$

$$\mathbf{A}^T\mathbf{U_{k+1}} = \mathbf{V_k}\mathbf{B_k}^T + \alpha_{k+1}\mathbf{v_{k+1}}\mathbf{e}_{\mathbf{k+1}}^T \tag{5.6}$$

where $\mathbf{e_1}$ and $\mathbf{e_{k+1}}$ are the standard basis vectors. This gives the desired lower bidiagonal form.

5.2.2. ALGORITHM LSQR

Define the residual $\mathbf{r_k}$ as $\mathbf{r_k} = \mathbf{b} - \mathbf{Ax_k}$. The Algorithm LSQR method generates a sequence of approximations $\{\mathbf{x_k}\}$ such that the residual norm $||\mathbf{r_k}||$ decreases monotonically.
In [Paige and Saunders, 1982] is explained how the bidiagonalization procedure can be used for the Algorithm LSQR. Let the quantities

$$\mathbf{x_k} = \mathbf{V_k y_k}$$
$$\mathbf{r_k} = \mathbf{b} - \mathbf{Ax_k}$$
$$\mathbf{t_{k+1}} = \beta_1 \mathbf{e_1} - \mathbf{B_k y_k}$$

be defined in terms of some vector $\mathbf{y_k}$. It readily follows from (5.4) and (5.5) that

$$\mathbf{r_k} = \mathbf{U_{k+1}t_{k+1}}$$

We want $||\mathbf{r_k}||$ to be small, and because $\mathbf{U_{k+1}}$ is bounded and theoretically has orthonormal columns, this suggests choosing $\mathbf{y_k}$ to minimize $||\mathbf{t_{k+1}}||$. This leads to the least-squares problem

$$\min ||\beta_1 \mathbf{e_1} - \mathbf{B_k y_k}||$$

which forms the basis for Algorithm LSQR.

5.3. RECONSTRUCTION ALGORITHMS

In the algorithm for the image reconstruction, the Matlab function `lsqr` will be frequently used. In this section, the reconstruction algorithm is given in a general form. The corresponding Matlab code can be found in Appendix C.
We will consider three different scenario's:

1. Constant variance.

   In the current reconstruction method, the assumption is made that the noise is standard normally distributed and thus has constant variance. This leads to a least-squares problem with constant weights. The minimum norm least-squares solution is computed with the Matlab function `lsqr`. See also Section 4.2.

2. Non-constant variance: path length.

   An alternative reconstruction method uses the length of the path that the X-ray photons have travelled in a beam, giving a larger weight to the measurements obtained from the beams that form a shorter path through the body part of interest. This leads to a weighted least-squares problem with weights corresponding to the path length. The Matlab function `lsqr` is used to compute the minimum norm least-squares solution to this problem. See also Section 4.3.

3. Non-constant path dependent variance.

   Under the assumption of non-constant path dependent variance, four reconstruction algorithms are developed.

   In the first reconstruction method, the exact solution is used to compute the variance in each beam. Next, the square of the variance is used as weights in the corresponding weighted least-squares problem. The minimum norm least-squares solution is computed with the Matlab function `lsqr`.

Starting with the minimum norm least-squares solution from the former reconstruction method, an iterative reconstruction algorithm is developed. In each iteration the Matlab function `lsqr` gives a new minimum norm least-squares solution to the weighted least-squares problem with weights based on the previous solution. The algorithm stops when the solution is converged or when the maximum number of iterations is reached.

Because this iterative reconstruction algorithm does not appear to be convergent, we use relaxation for the next algorithm. The same approach is used as in the iterative algorithm, but now the new solution is in each iteration computed as the weighted average of the previous solution and the solution obtained by the Matlab function `lsqr` for the weighted least-squares problem.

Finally, the same approach is used as in the former reconstruction algorithm, but starting with the minimum norm least-squares solution from the algorithm with constant variance to calculate the variance in the first iteration instead of the exact solution.

See also Section 4.3.1.

This leads to the following algorithms:

Reconstruction algorithms

1. (Constant variance.)

   Compute the minimum norm least-squares solution $\mathbf{x_{lsqr}}$ to $\mathbf{Ax} = \mathbf{b}$.

2. (Path length.)

   Define $w_i :=$ path length of beam $i := \sum_{j=1}^{N} a_{ij}$. Let $\mathbf{B}$ be an empty M by N matrix and $\mathbf{c}$ an empty vector of length M.

   for $i = 1 : M$

   if $w_i = 0$

   Row i of matrix $\mathbf{B}$ = row with all elements 0

   Element i of vector $\mathbf{c} = 0$

   else

   Row i of matrix $\mathbf{B} = \frac{\text{row } i \text{ of matrix } \mathbf{A}}{w_i}$

   Element i of vector $\mathbf{c} = \frac{\text{element } i \text{ of vector } \mathbf{b}}{w_i}$

   end if

   end for

   Compute the minimum norm least-squares solution $\mathbf{x_{path}}$ to $\mathbf{Bx} = \mathbf{c}$.

3. (Path dependent variance.)

   (a) (Exact variance.)

   Define

   $$\sigma_i(\mathbf{x}) := \text{standard deviation of the noise in beam } i := \sqrt{\frac{1}{m}\frac{1-p_i}{p_i}} \qquad \text{with } p_i = \exp\left[-\sum_{j=1}^{N} a_{ij}x_j\right]$$

   where $\mathbf{x}$ is a minimum norm least-squares solution of the problem. Let $\mathbf{D}$ be an empty M by N matrix and $\mathbf{e}$ an empty vector of length M. Suppose that we know the exact solution $\mathbf{x_{exact}}$ of the problem. Apply the following algorithm:

   ALGORITHM

   Input: $\sigma_i(\mathbf{x_{exact}})$, $\mathbf{D}$, $\mathbf{e}$

   Output: $\mathbf{x_1}$

for $i = 1 : M$
  if $\sigma_i(\mathbf{x_{exact}}) = 0$
    Row i of matrix $\mathbf{D}$ = row with all elements 0
    Element i of vector $\mathbf{e} = 0$
  else
    Row i of matrix $\mathbf{D} = \frac{\text{row } i \text{ of matrix } \mathbf{A}}{\sigma_i(\mathbf{x_{exact}})}$
    Element i of vector $\mathbf{e} = \frac{\text{element } i \text{ of vector } \mathbf{b}}{\sigma_i(\mathbf{x_{exact}})}$
  end if
end for
Compute the minimum norm least-squares solution $\mathbf{x_1}$ to $\mathbf{Dx} = \mathbf{e}$.

(b) (Iterative algorithm.)
Define $\sigma_i(\mathbf{x})$ as in (a) and let $\mathbf{x_1}$ be the minimum norm least-squares solution of (a).

$i = 0$
max = 50
$\mathbf{x_0}$ = vector of length M with all elements 0

while $||\mathbf{x_i} - \mathbf{x_{i+1}}|| > 10^{-6}$ **and** $i <$ max
  Compute $\sigma_i(\mathbf{x_{i+1}})$ for each beam $i = \{1, 2, \ldots, M\}$
  Construct an empty M by N matrix $\mathbf{E}$ and an empty vector $\mathbf{f}$ of length M.
  Apply ALGORITHM
    Input: $\sigma_i(\mathbf{x_{i+1}})$, $\mathbf{E}$, $\mathbf{f}$
    Output: $\mathbf{x_{i+2}}$
  $i = i + 1$
end while

(c) (Relaxation.)
Define $\sigma_i(\mathbf{x})$ as in (a) and let $\mathbf{x_1}$ be the minimum norm least-squares solution of (a).

$i = 0$
max = 50
$\alpha = 0.7$
$\mathbf{x_0}$ = vector of length M with all elements 0
$\mathbf{x_{relax_0}} = \mathbf{x_0}$
$\mathbf{x_{relax_1}} = \mathbf{x_1}$

while $||\mathbf{x_{relax_i}} - \mathbf{x_{relax_{i+1}}}|| > 10^{-6}$ **and** $i <$ max
  Compute $\sigma_i(\mathbf{x_{relax_{i+1}}})$ for each beam $i = \{1, 2, \ldots, M\}$
  Construct an empty M by N matrix $\mathbf{E}$ and an empty vector $\mathbf{f}$ of length M.
  Apply ALGORITHM
    Input: $\sigma_i(\mathbf{x_{relax_{i+1}}})$, $\mathbf{E}$, $\mathbf{f}$
    Output: $\mathbf{x_{i+2}}$
  $\mathbf{x_{relax_{i+2}}} = \alpha\, \mathbf{x_{i+1}} + (1 - \alpha)\, \mathbf{x_{i+2}}$
  $i = i + 1$
end while

(d) (New reconstruction algorithm.)
Define $\sigma_i(\mathbf{x})$ as in (a) and let $\mathbf{x_{lsqr}}$ be the minimum norm least-squares solution of 1. (Constant variance.).

$i = 0$
max = 50
$\alpha = 0.7$
$\mathbf{x_0}$ = vector of length M with all elements 0
$\mathbf{x_{relax_0}} = \mathbf{x_0}$

Let $\mathbf{D}$ be an empty M by N matrix and $\mathbf{e}$ an empty vector of length M.
Apply ALGORITHM
Input: $\sigma_i(\mathbf{x_{lsqr}})$, $\mathbf{D}$, $\mathbf{e}$
Output: $\mathbf{x_1}$
$\mathbf{x}_{\text{relax}_1} = \mathbf{x_1}$

while $||\mathbf{x_{relax_i}} - \mathbf{x_{relax_{i+1}}}|| > 10^{-6}$ **and** $i <$ max
Compute $\sigma_i(\mathbf{x_{relax_{i+1}}})$ for each beam $i = \{1, 2, \ldots, M\}$
Construct an empty M by N matrix $\mathbf{E}$ and an empty vector $\mathbf{f}$ of length M.
Apply ALGORITHM
Input: $\sigma_i(\mathbf{x_{relax_{i+1}}})$, $\mathbf{E}$, $\mathbf{f}$
Output: $\mathbf{x_{i+2}}$
$\mathbf{x_{relax_{i+2}}} = \alpha\, \mathbf{x_{i+1}} + (1-\alpha)\, \mathbf{x_{i+2}}$
$i = i + 1$
end while

Note that, when the standard deviation is equal to zero, there is chosen to set the corresponding weights (in the weighted least-squares sense) equal to zero. This might feel counter-intuitive, because a standard deviation of zero means an exact measurement. But in this case the standard deviation is equal to zero if and only if $\sum_{j=1}^{N} a_{ij} x_j = 0$. The vector with unknown attenuation coefficients is not equal to zero, so $x_j \neq 0$ for some j and all matrix coefficients $a_{ij} \geq 0$ because they denote the length of the travelled path (see Section 2.3). This leads to the conclusion that the standard deviation is equal to zero if in row i of matrix $\mathbf{A}$ all coefficients are zero and thus no X-ray beam has travelled that path. It is therefore beneficial to set the ith element in the vector with measurements equal to zero, because only noise could have been measured there.

6

RESULTS

In this chapter, the knowledge gained in the previous chapters is used to find a better reconstruction algorithm for the CT-scan. Several experiments are performed to see if improvements can be made on the current reconstruction method.

Because real data is not available, the Shepp-Logan phantom is used for the experiments. This standard test image created by Larry Shepp and Benjamin F. Logan serves as the model of a human head. It is widely used in the development and testing of image reconstruction algorithms. This phantom is implemented in Matlab. For the experiments, a slightly smaller phantom is used as test problem to speed up the calculations.
In this test problem, $M = 12,780$ measurements are available to reconstruct the attenuation coefficients ϕ_j, for $j = \{1,2,\ldots,N\}$, discretized over $N = 2,500$ cells. The number of rays per beam is equal to 71 and a beam angle spacing of 2° is used. In each experiment, a noise level of 10% is used and the maximum number of iterations is set to 100. Furthermore, the accuracy is set to 10^{-6} and a seed number is used for the random number generators such that it is possible to compare the results of the experiments. The corresponding Matlab code can be found in Appendix C.
For the results the relative error is calculated by

$$\frac{||\boldsymbol{\phi}_{\mathbf{exact}} - \boldsymbol{\phi}||}{||\boldsymbol{\phi}_{\mathbf{exact}}||} \tag{6.1}$$

where $\boldsymbol{\phi}$ is the minimum norm least-squares solution obtained from the different reconstruction methods.

6.1. CURRENT RECONSTRUCTION METHOD

As a start, the current reconstruction method will be evaluated. This method uses the assumption that the noise is standard normally distributed. The LSQR algorithm then gives the minimum norm least-squares solution to the corresponding problem.

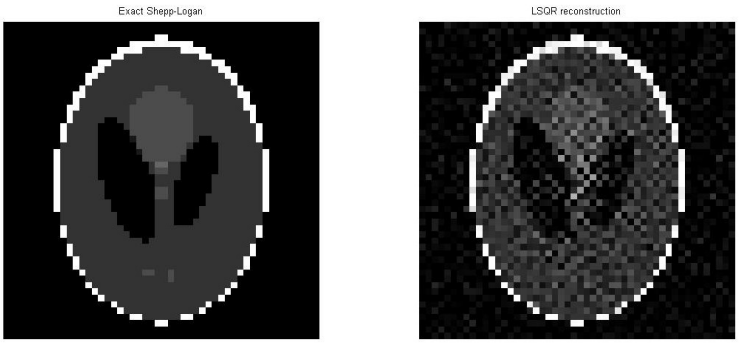


Figure 6.1: Current reconstruction method with standard normally distributed noise.

From Figure 6.1 we can see that the image reconstruction gives a good result. The relative error of this reconstructed solution is equal to 0.34636.
But we know from Section 3.3 that the assumption of standard normally distributed noise is not true. Therefore, we will now use the theoretical noise distribution and see if the current reconstruction algorithm still performs well.

6.1.1. THEORETICAL NOISE DISTRIBUTION

Recall from Section 3.5 that for the linear system of equations

$$\mathbf{A}\boldsymbol{\phi} + \boldsymbol{\epsilon} = \boldsymbol{\psi}$$

the noise is distributed as

$$\epsilon_b \sim \text{Normal}\left(0,\ \frac{1}{m}\frac{1-p_b}{p_b}\right) \qquad \text{with } p_b = \exp\left[-\int_0^{l_b} \phi_b(x)\,\mathrm{d}x\right] \text{ in beam } b.$$

Using the exact solution to compute p_b and applying the theoretical noise distribution to the problem gives a more realistic situation. A value of $m = 10{,}000$ is chosen for the number of photons in each X-ray beam. The image reconstruction obtained from the current reconstruction method can be found in Figure 6.2.

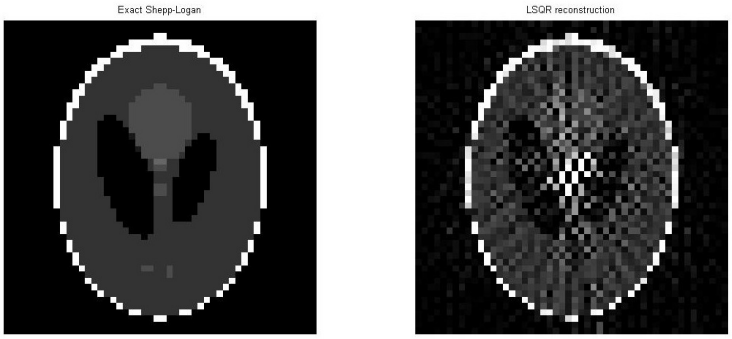


Figure 6.2: Current reconstruction method with the theoretically distributed noise.

This time the current reconstruction method gives a much worse result. The relative error of the reconstructed solution is now equal to 0.66723.
Because the current reconstruction method is no longer adequate, we will take a look at alternative reconstruction methods in the next sections. From now on, only the theoretical distribution is used for noise in the model.

6.2. PATH LENGTH

One alternative reconstruction method uses the length of the path that the X-ray photons have travelled in beam b, giving a larger weight to the measurements obtained from the beams that form a shorter path through the body part of interest. This gives rise to a weighted least-squares problem

$$\sum_{i=1}^{M}\left(\frac{\psi_i - (\mathbf{A}\boldsymbol{\phi})_i}{w_i}\right)^2 = \sum_{i=1}^{M}\left(\frac{\epsilon_i}{w_i}\right)^2$$

where w_i, for $i = \{1, 2, \ldots, M\}$, is the path length corresponding to the ith beam. The LSQR algorithm gives the minimum norm least-squares solution to this problem.

Looking at Figure 6.3 it can be seen that the current LSQR reconstruction and the reconstruction that uses the path length as weights perform almost equally bad. The relative error of the weighted LSQR is 0.53277 and hence this reconstruction is actually slightly better than the standard LSQR reconstruction, which has a relative error of 0.66723.

Because this alternative reconstruction method is not as good as we hoped, another approach will be considered in the next section.

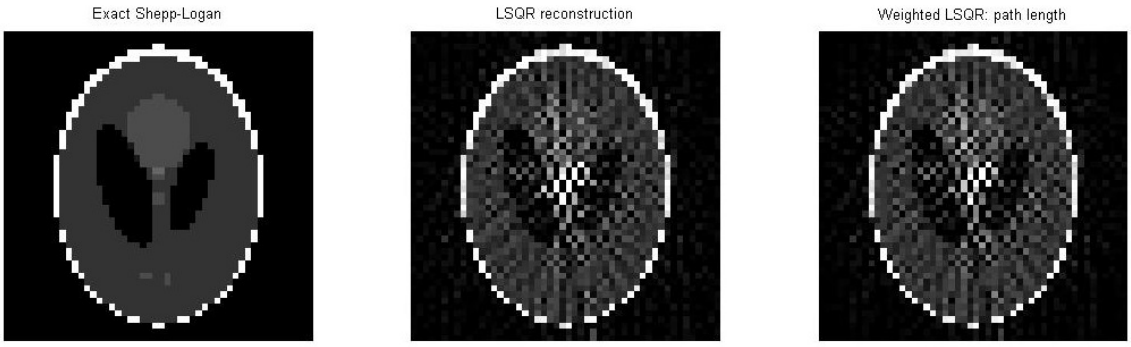


Figure 6.3: Weighted LSQR reconstruction based on the length of the path that the X-ray photons travel.

6.3. NON-CONSTANT EXACT VARIANCE

In Section 4.3.1 it was found that for non-constant path dependent variance the square root of the variance could be used as weight in the weighted least-squares problem instead of the path length. This idea forms the basis for the next reconstruction algorithm.

Let $\sigma_i(\boldsymbol{\phi})^2$ be the path dependent variance of the noise for each X-ray beam $i = \{1,2,\ldots,M\}$. We assume that $\sigma_i(\boldsymbol{\phi_0})^2$ is known in advance, for example from an initial estimation $\boldsymbol{\phi_0}$. Then the corresponding weighted least-squares problem is:

$$\sum_{i=1}^{M}\left(\frac{\psi_i-(\mathbf{A}\boldsymbol{\phi})_i}{\sigma_i(\boldsymbol{\phi_0})}\right)^2=\sum_{i=1}^{M}\left(\frac{\epsilon_i}{\sigma_i(\boldsymbol{\phi_0})}\right)^2 \tag{6.2}$$

In the first reconstruction method, $\sigma_i(\boldsymbol{\phi_{\text{exact}}})^2$ is calculated by using the exact solution $\boldsymbol{\phi_{\text{exact}}}$. If this method does not improve the image reconstruction whilst using the exact solution in computing $\sigma_i(\boldsymbol{\phi_{\text{exact}}})^2$, it is useless to try to obtain also a better image reconstruction using a non-exact solution.

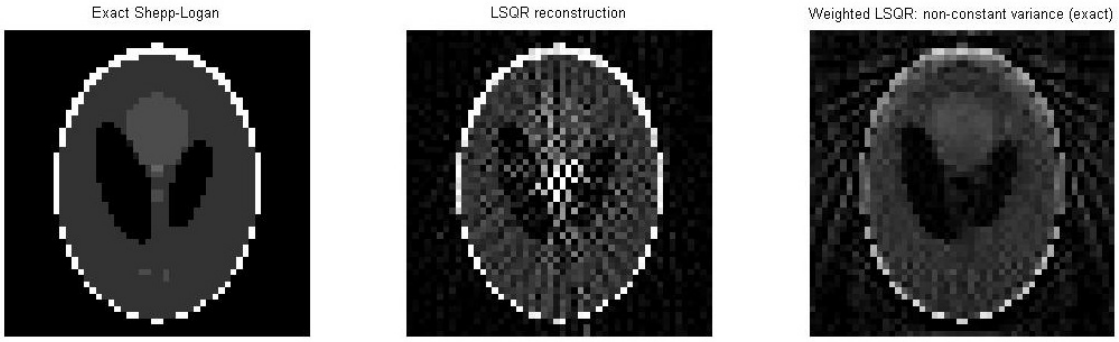


Figure 6.4: Weighted LSQR reconstruction based on the square root of the path dependent variance of the noise, $\sigma_i(\boldsymbol{\phi_{\text{exact}}})$, calculated with the exact solution $\boldsymbol{\phi_{\text{exact}}}$ for each beam $i = \{1,2,\ldots,M\}$.

Fortunately, we can see in Figure 6.4 that the image reconstruction is a lot better when we use the square root of the non-constant path dependent variance $\sigma_i(\boldsymbol{\phi_{\text{exact}}})$ as weights. A relative error of 0.51476 is found for this reconstruction method.

This improvement in the reconstruction gives reason to continue with the weighted least-squares problem from (6.2). In the next section, an iterative algorithm will be used to compute a minimum norm least-squares solution to the weighted least-squares problem.

6.3.1. ITERATIVE ALGORITHM

Starting with the minimum norm least-squares solution $\boldsymbol{\phi_0}$ from the previous reconstruction method, with non-constant exact variance $\sigma_i(\boldsymbol{\phi_{\text{exact}}})^2$ as weights, we want to set up an iterative algorithm which converges to the optimal solution. In the first iteration a new minimum norm solution $\boldsymbol{\phi_1}$ is calculated to the weighted least-squares problem (6.2) with weights $\sigma_i(\boldsymbol{\phi_0})^2$. The solution $\boldsymbol{\phi_1}$ obtained from this problem is used in the next iteration to compute the variance $\sigma_i(\boldsymbol{\phi_1})^2$ and again the minimum norm least-squares solution $\boldsymbol{\phi_2}$ to (6.2) is calculated. This procedure is repeated until the solution converges or until a maximum number of iterations is reached. In Section 5.3 this iterative reconstruction algorithm was described in more detail.
In Figure 6.5 the result of this iterative algorithm is shown. A relative error of 0.40862 is found for this reconstruction method, which is even better than the previous reconstruction method.

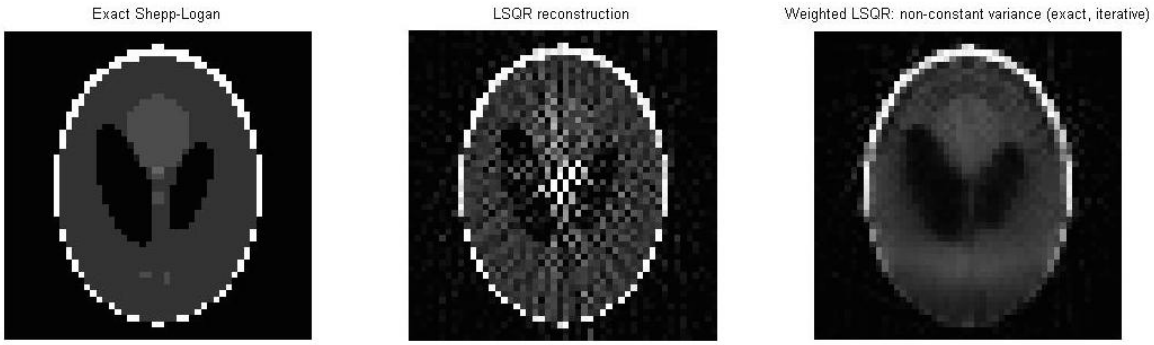


Figure 6.5: Iterative weighted LSQR reconstruction based on the square root of the path dependent variance of the noise, $\sigma_i(\boldsymbol{\phi})$, calculated with the previous solution $\boldsymbol{\phi}$ in each iteration.

To see if this iterative method converges, we will use several 'measures of convergence':

1. The log-likelihood function; see (4.3).
2. The relative error; see (6.1).
3. The weighted sum of squared errors; see (6.2).

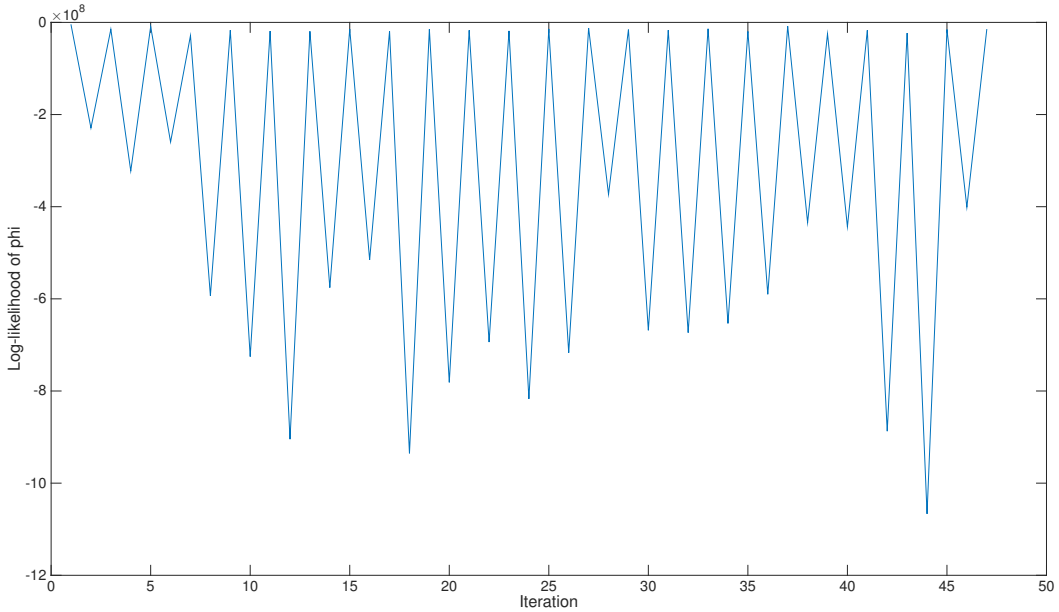


Figure 6.6: Log-likelihood of $\boldsymbol{\phi}$, calculated in each iteration step.

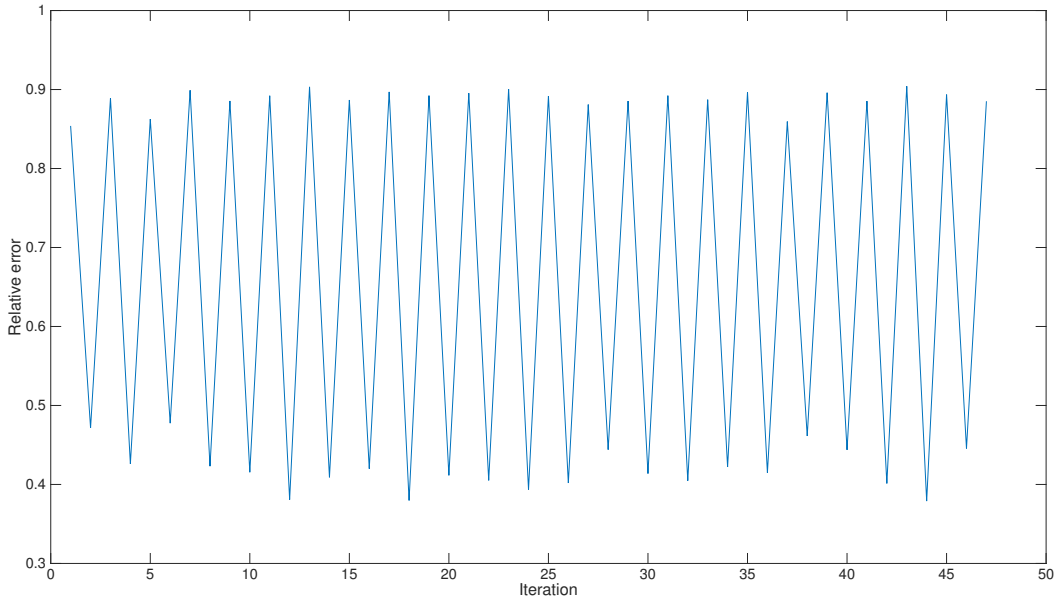


Figure 6.7: Relative error, calculated in each iteration step.

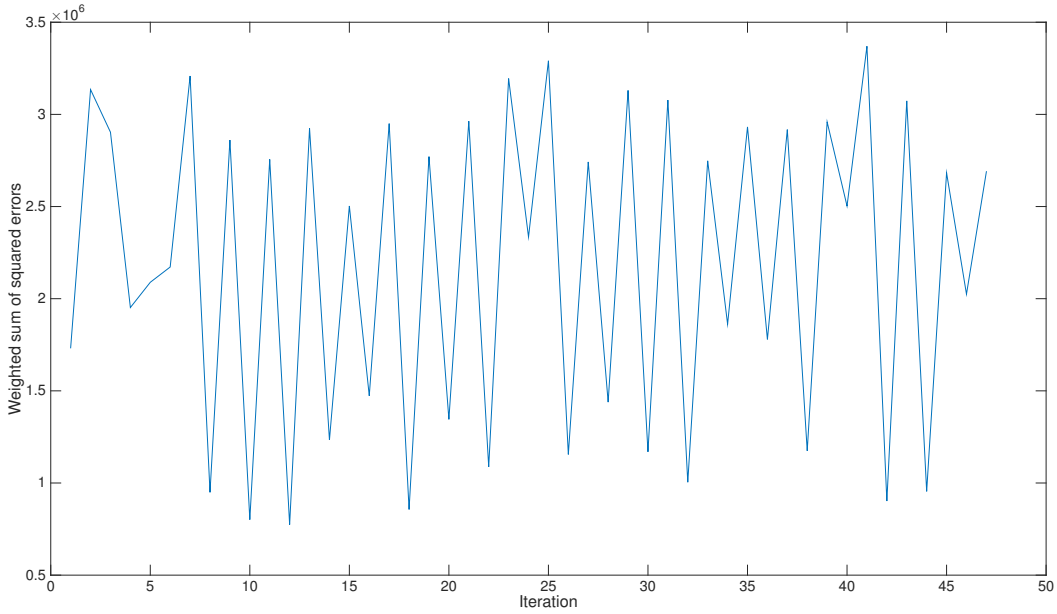


Figure 6.8: Weighted sum of squared errors, calculated in each iteration step.

In Figure 6.6 we can see that the log-likelihood of $\boldsymbol{\phi}$ does not monotonically increase, but is alternating. The same observation can be made for the relative error and the weighted sum of squared errors: for convergence both of these measures should be more or less monotonically decreasing, but from Figure 6.7 and 6.8 we can see that we get the same alternating behaviour as for the log-likelihood.
This leads to the conclusion that our current reconstruction algorithm does not converge and that the good result obtained for the image reconstruction could be just a coincidence. Therefore, we will add relaxation to the reconstruction algorithm to try to get a converging algorithm.

6.3.2. RELAXATION

The alternating behaviour of the 'measures of convergence' from the previous section could be caused by the fact that the minimum norm least-squares solution in each iteration goes past the optimal solution but cannot converge to it. Therefore, relaxation could be used to help establish convergence. We will briefly explain how this relaxation works.

Assume that we can write for the original iteration process

$$\boldsymbol{\phi_{n+1}} = f(\boldsymbol{\phi_n})$$

then the relaxation of this iteration process would be

$$\boldsymbol{\phi_{n+1}} = \alpha\boldsymbol{\phi_n} + (1-\alpha)f(\boldsymbol{\phi_n})$$

for some $\alpha \in (0,1)$.

To find out which value of α is best for the relaxation, several values of α are tested in the relaxation of a similar (but slightly smaller) problem. The three different measures of convergence are then considered to determine the best value of α. The results can be found in Appendix B.1. A value of $\alpha = 0.7$ was chosen as best.

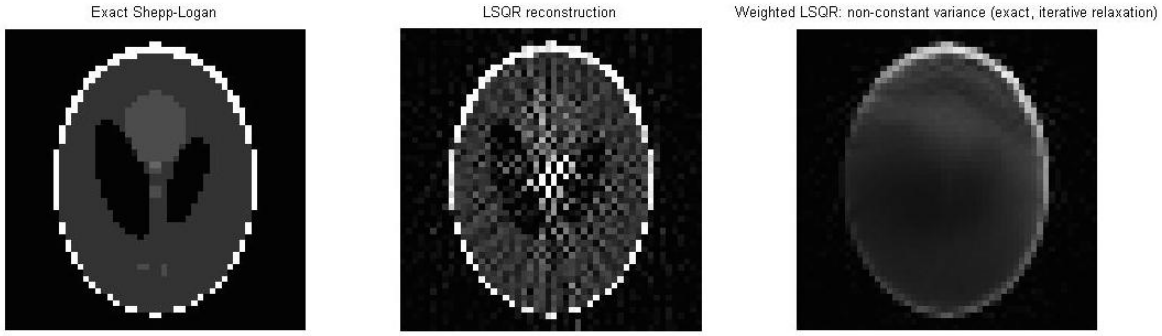


Figure 6.9: Iterative weighted LSQR reconstruction with relaxation and weights based on the square root of the path dependent variance of the noise, $\sigma_i(\boldsymbol{\phi})$, calculated with the previous solution $\boldsymbol{\phi}$ in each iteration.

Adding this relaxation to the reconstruction algorithm and calculating the minimum norm least-squares solution results in Figure 6.9. Although the image reconstruction might look not so good, a relative error of 0.55792 was found, which is smaller than the relative error of the LSQR reconstruction. But is the adjusted algorithm convergent?

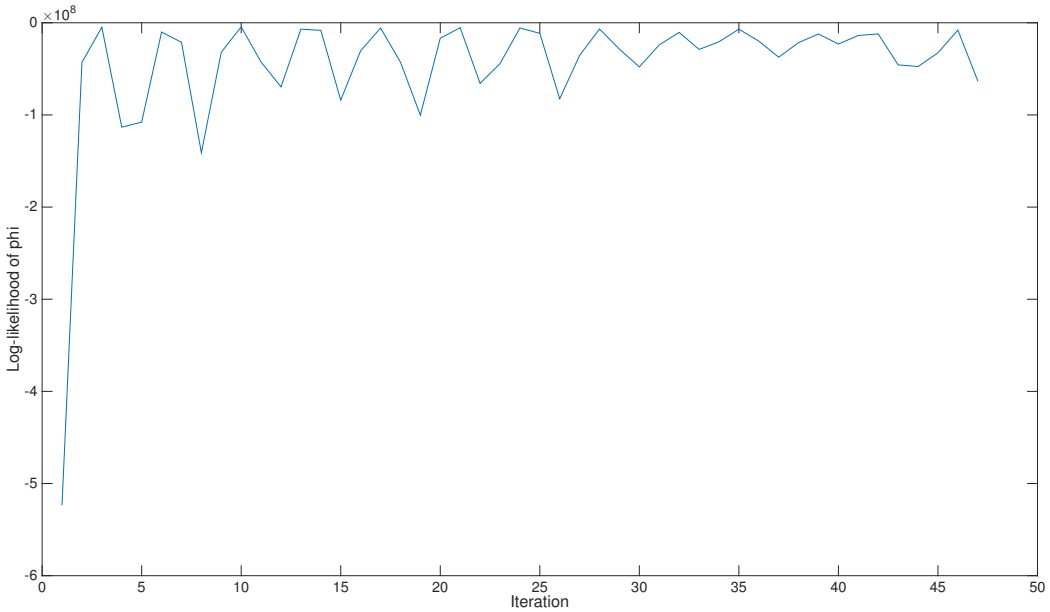


Figure 6.10: Log-likelihood of $\boldsymbol{\phi}$ for the relaxation, calculated in each iteration step.

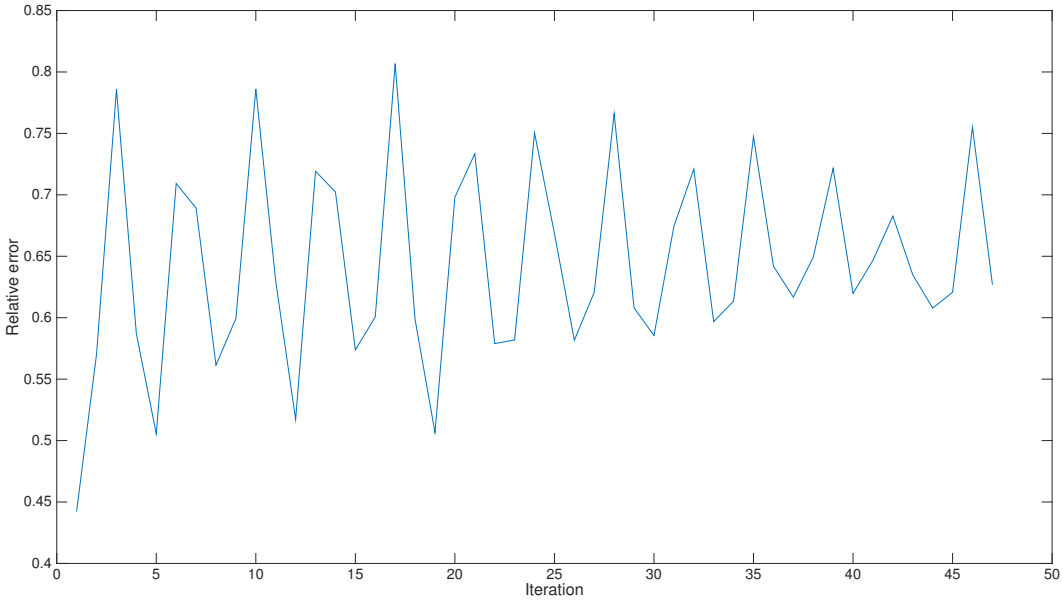


Figure 6.11: Relative error for the relaxation, calculated in each iteration step.

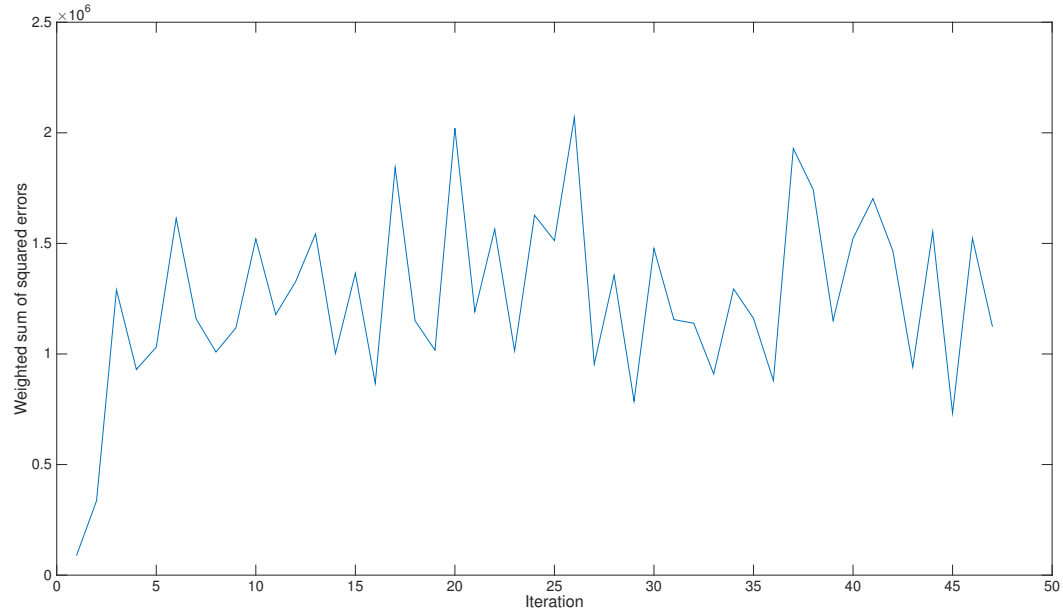


Figure 6.12: Weighted sum of squared errors for the relaxation, calculated in each iteration step.

Again the three 'measures of convergence' are considered for the reconstruction. When we look at the log-likelihood of $\boldsymbol{\phi}$ in Figure 6.10 we see that a better result is obtained: although the log-likelihood still does not increase monotonically, the previous alternating behaviour remains limited.
For the relative error and the weighted sum of squared errors in Figure 6.11 and 6.12 the results are slightly better than without the relaxation, although the alternating behaviour remains.

6.4. NEW RECONSTRUCTION METHOD

In reality, the exact solution is unknown when performing a CT-scan. Therefore, the method described in the previous section cannot be used in its current form. Luckily, only a small adjustment has to be made: instead of using the exact solution $\boldsymbol{\phi}_{\mathbf{exact}}$ to calculate the variance in the first iteration step, we use the solution $\boldsymbol{\phi}_{\mathbf{lsqr}}$ obtained by the LSQR algorithm.

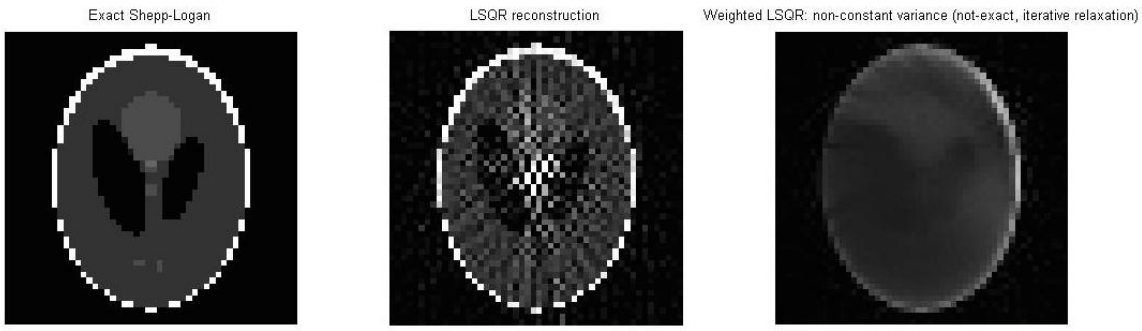


Figure 6.13: Iterative weighted LSQR reconstruction with relaxation and weights based on the square root of the path dependent variance of the noise, $\sigma_i(\boldsymbol{\phi})$, calculated with the LSQR solution $\boldsymbol{\phi}_{\mathbf{lsqr}}$ in the first step and the previous solution $\boldsymbol{\phi}$ in each following iteration.

In Figure 6.13 the image reconstruction can be found obtained by this iterative algorithm, which uses the LSQR solution to compute the variance in the first step of the algorithm and relaxation (with $\alpha = 0.7$) to help establish convergence. The result is almost similar to the image reconstruction shown in Figure 6.9 and again, although the image reconstruction might look not so good, a relative error of 0.59630 was found, which is still smaller than the relative error of the LSQR reconstruction.

For the 'measures of convergence' a similar result is obtained as in Figure 6.10, 6.11 and 6.12. The corresponding plots can be found in Appendix B.2.

7

CONCLUSION

During this research, a stochastic model for attenuation in CT-scans was derived. Discretizing the tomographic image that has to be reconstructed gives rise to the consideration of the image reconstruction as a system of linear equations

$$\mathbf{A}\boldsymbol{\phi} + \boldsymbol{\epsilon} = \boldsymbol{\psi}$$

where

- a_{ij} are the matrix elements of $\mathbf{A}$ that reflect the relation between the area that is traversed by an X-ray beam and the entire area of a pixel in the grid;
- $\boldsymbol{\phi}$ is a column vector with the unknown attenuation coefficients;
- $\boldsymbol{\epsilon}$ is a column vector with the measurement errors;
- $\boldsymbol{\psi}$ is a column vector with the measurements.

It is found that the measurement errors are distributed as

$$\epsilon_i \sim \text{Normal}\left(0, c\,\frac{1-p_i}{p_i}\right) \qquad \text{with } p_i = \exp\left[-\sum_{j=1}^{N} a_{ij}\phi_j\right] \text{ in beam } i$$

for some constant c.

Incorporating the theoretical variance of the measurement errors in the problem gives a more realistic situation, but the image reconstruction obtained from the current reconstruction algorithm is deteriorated. Therefore, new reconstruction methods are constructed based on

1. path length;
2. non-constant exact variance;
3. non-constant not-exact variance.

The image reconstruction based on path length gives a slightly better result than the current reconstruction method. A relative error of 0.53277 is found, while the relative error of the current image reconstruction method is 0.66723.

When the exact solution is used to compute the variance of the measurement errors in the image reconstruction, another improvement is made: now a relative error of 0.51476 is found. Of course, in reality the exact solution is unknown and thus does this reconstruction method not meet our requirements.
An iterative algorithm is constructed which uses the exact solution only in the first step to compute the variance and uses the previous solution in the next step. Again improvement is made: the relative error of this image reconstruction is only 0.40862.

Using this iterative algorithm could be profitable, but unfortunately the algorithm does not converge. Therefore, relaxation is added to the reconstruction algorithm. This makes the convergence a little better, but the image reconstruction slightly worse: a relative error of 0.55792 is found.

To make a realistic new reconstruction method, the iterative algorithm with relaxation is adjusted such that the solution of the current algorithm is used to compute the variance in the first step instead of the exact solution. This results in a similar outcome: the relative error of this image reconstruction is 0.59630.

Because the best result is obtained by the iterative reconstruction algorithm, we conclude that it is beneficial for the image reconstruction to use the non-constant path dependent variance of the measurement errors in an iterative algorithm.

8

DISCUSSION

To obtain a stochastic model for attenuation in CT-scans, several assumptions were made. First of all, we assume that an X-ray photon travels through the body part of interest in a straight line, while in reality its path could change due to obstacles. We also assume that the probability that an X-ray photon decays anywhere on its path is not changed given the part of the path it has already travelled. However, an X-ray photon could be weaker because of the obstacles it already traversed which would increase the probability of decay.

Furthermore, the number of photons transmitted in an X-ray beam is unknown. In the reconstruction algorithms this number is arbitrarily chosen as 10,000, but it might be way larger in reality. We should also keep in mind that for the image reconstruction here only a standard test image is used and that the noise is added manually. The developed reconstruction algorithms could work a lot worse for realistic data, because the image that has to be constructed is far more complex than this test image and the variance of the noise could differ from the theoretical distribution.

To measure the convergence of the reconstruction methods, the log-likelihood function, the relative error and the weighted sum of squared errors are used. It could be argued that other measures of convergence should be taken into account. Also, the relaxation parameter was chosen based on subjective observations and not objectively. The new reconstruction algorithm does not give the desired result, but great improvements could be made when an iterative reconstruction algorithm which does converge is found.

In future work it would be interesting to investigate if the theoretical variance of the measurement errors corresponds to the real behaviour of the errors. There could be experimented with the number of X-ray photons to see what the effects are for the image reconstruction and to see if using the exact number of photons results in a more accurate reconstruction.

Alternative modeling of the variance of the measurements could be done by using available data or standard knowledge on anatomy. These known solutions could be combined to an estimation of the solution from which we can compute the variance. Then the same approach as with the exact solution could be used to see if the image reconstruction can be improved.

More research on numerical methods is needed to improve the convergence of the iterative reconstruction algorithm and to find a better solution for the image reconstruction. Also, it could be considered to estimate the attenuation coefficient by minimizing over both the mean as the variance as stated in Section 4.4.

BIBLIOGRAPHY

[Buzug, 2008] T. M. Buzug, *Computed tomography: from photon statistics to modern cone-beam ct*, (Springer Science & Business Media, 2008) Chap. 6, pp. 201-204.

[CyberPhysics] `http://www.cyberphysics.co.uk/topics/medical/CTScanner.htm`.

[Golub and Kahan, 1965] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, Journal of the Society for Industrial & Applied Mathematics, Series B: Numerical Analysis **2**, 205 (1965).

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

[Paige and Saunders, 1982] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Transactions on Mathematical Software (TOMS) **8**, 43 (1982).

[Papanicolaou, 2009] A. Papanicolaou, *Taylor approximation and the delta method*, `http://www.stanford.edu` (2009).

[Schechter] E. Schechter, *The cubic formula (solve any 3rd degree polynomial equation)* `http://www.math.vanderbilt.edu/~schectex/courses/cubic/`.

[Ter-Pogossian, 1977] M. M. Ter-Pogossian, *Basic principles of computed axial tomography*, in *Seminars in nuclear medicine*, Vol. 7 (Elsevier, 1977) pp. 109-127.

[Waseda et al., 2011] Y. Waseda, E. Matsubara, and K. Shinoda, *X-ray diffraction crystallography: introduction, examples and solved problems*, (Springer Science & Business Media, 2011) Chap. 1, pp. 1-2.

A

Maximum-Likelihood Estimation

A.1. Maximum-Likelihood Estimation of $\theta_b = \int_0^{l_b} \phi_b(x)\, dx$

Recall the normal distribution:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

Because $\theta_b = \int_0^{l_b} \phi_b(x)\, \mathrm{d}x$ we have $p_b = e^{-\theta_b}$. Hence, $Y_b \sim \text{Normal}(\,\mu_b,\, \sigma_b^2\,)$ with

$$\mu_b = \mu(\theta_b) = e^{-\theta_b}$$

$$\sigma_b^2 = \sigma(\theta_b)^2 = \frac{1}{m}\, e^{-\theta_b}(1 - e^{-\theta_b}) = \frac{1}{m}\, e^{-\theta_b} - \frac{1}{m}\, e^{-2\theta_b}.$$

Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M)^T$ be a vector that contains the unknown θ_b for all beams $b = \{1, 2, \ldots, M\}$. Then the corresponding likelihood of $\boldsymbol{\theta}$ is equal to

$$\begin{aligned}
\mathscr{L}(\boldsymbol{\theta};\, y_1, y_2, \ldots, y_M) &= f(y_1, y_2, \ldots, y_M \mid \boldsymbol{\theta}) = \prod_{i=1}^{M} f(y_i \mid \theta_i) \\
&= \prod_{i=1}^{M} \frac{1}{\sqrt{2\pi\sigma(\theta_i)^2}} \exp\left[-\frac{(y_i - \mu(\theta_i))^2}{2\sigma(\theta_i)^2}\right]
\end{aligned}$$

Taking the natural logarithm of the likelihood results in the log-likelihood

$$\begin{aligned}
\ell(\boldsymbol{\theta};\, y_1, y_2, \ldots, y_M) &= \ln\{\mathscr{L}(\boldsymbol{\theta};\, y_1, y_2, \ldots, y_M)\} \\
&= \sum_{i=1}^{M} \left(\ln\left\{ \frac{1}{\sqrt{2\pi\sigma(\theta_i)^2}} \right\} - \frac{(y_i - \mu(\theta_i))^2}{2\sigma(\theta_i)^2} \right) \\
&= \sum_{i=1}^{M} \left(-\frac{1}{2} \ln\{2\pi\sigma(\theta_i)^2\} - \frac{(y_i - \mu(\theta_i))^2}{2\sigma(\theta_i)^2} \right)
\end{aligned}$$

Substituting $\mu(\theta_i)$ and $\sigma(\theta_i)^2$ in the log-likelihood gives:

$$\begin{aligned}
\ell(\boldsymbol{\theta};\, y_1, y_2, \ldots, y_M) = \sum_{i=1}^{M} \Bigg(&-\frac{1}{2} \ln\{2\pi\} - \frac{1}{2} \ln\left\{\frac{1}{m}\right\} + \frac{1}{2}\, \theta_i - \frac{1}{2} \ln\{1 - e^{-\theta_i}\} \\
&- \frac{m(y_i - e^{-\theta_i})^2}{2e^{-\theta_i} - 2e^{-2\theta_i}} \Bigg)
\end{aligned}$$

Taking the derivative of the log-likelihood with respect to θ_i results in

$$\begin{aligned}
\frac{\partial \ell}{\partial \theta_i} &= \frac{1}{2} - \frac{1}{2}\, \frac{e^{-\theta_i}}{1 - e^{-\theta_i}} + \frac{m(4e^{-2\theta_i} - 2e^{-\theta_i})(y_i - e^{-\theta_i})^2 - 2me^{-\theta_i}(y_i - e^{-\theta_i})(2e^{-\theta_i} - 2e^{-2\theta_i})}{(2e^{-\theta_i} - 2e^{-2\theta_i})^2} \\
&= \frac{1}{2} - \frac{1}{2}\, \frac{1}{e^{\theta_i} - 1} - \frac{my_i^2 e^{3\theta_i} - 2my_i^2 e^{2\theta_i} + m(2y_i - 1)e^{\theta_i}}{2(e^{\theta_i} - 1)^2}
\end{aligned}$$

The extreme values of the log-likelihood can be found by setting the derivative equal to zero:

$$\frac{\partial \ell}{\partial \theta_i} = 0$$

$$\Longleftrightarrow$$

$$\frac{1}{2} - \frac{1}{2}\frac{1}{e^{\theta_i}-1} - \frac{m y_i^2 e^{3\theta_i} - 2m y_i^2 e^{2\theta_i} + m(2y_i - 1)e^{\theta_i}}{2(e^{\theta_i}-1)^2} = 0$$

$$\Longleftrightarrow$$

$$\frac{-(e^{\theta_i}-1) - m y_i^2 e^{3\theta_i} + 2m y_i^2 e^{2\theta_i} - m(2y_i-1)e^{\theta_i}}{2(e^{\theta_i}-1)^2} = -\frac{1}{2}$$

$$\Longleftrightarrow$$

$$\frac{-m y_i^2 e^{3\theta_i} + 2m y_i^2 e^{2\theta_i} + (-2m y_i + m - 1)e^{\theta_i} + 1}{e^{2\theta_i} - 2e^{\theta_i} + 1} = -1$$

$$\Longleftrightarrow$$

$$-e^{2\theta_i} + 2e^{\theta_i} - 1 = -m y_i^2 e^{3\theta_i} + 2m y_i^2 e^{2\theta_i} + (-2m y_i + m - 1)e^{\theta_i} + 1$$

$$\Longleftrightarrow$$

$$m y_i^2 e^{3\theta_i} + (-2m y_i^2 - 1)e^{2\theta_i} + (2m y_i - m + 3)e^{\theta_i} - 2 = 0$$

This last expression is a polynomial of third degree for e^{θ_i} and can be solved exact by using Cardano's formula [Schechter]:

$$e^{\theta_i} = \sqrt[3]{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right) + \sqrt{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right)^2 + \left(\frac{c}{3a} - \frac{b^2}{9a^2}\right)^3}}$$
$$+ \sqrt[3]{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right) - \sqrt{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right)^2 + \left(\frac{c}{3a} - \frac{b^2}{9a^2}\right)^3}} - \frac{b}{3a}$$

with

$$a = m y_i^2$$
$$b = -2m y_i^2 - 1$$
$$c = 2m y_i - m + 3$$
$$d = -2$$

where m is the number of X-ray photons transmitted in each beam $b = \{1, 2, \ldots, M\}$.

Which of these three roots of the derivative of the log-likelihood gives the maximum?

With a little help from Maple the three solutions for θ_i are evaluated and it follows that two of the solutions are complex, leaving one solution as the desired maximum:

$$\theta_i = \ln\Bigg\{ \frac{1}{6my_i^2}\Big(-144m^3y_i^5 - 72m^2y_i^3 + 72m^3y_i^4 + 36m^2y_i^2 + 96m^2y_i^4 - 60my_i^2 + 64m^3y_i^6 + 8 \\ + 12my_i^2\sqrt{-36my_i + 18m - 156m^2y_i^2 + 12m^2y_i + 288m^2y_i^3 - 3m^2 - 144m^2y_i^4 + 672m^3y_i^4} \\ \overline{-384m^3y_i^3 + 144m^4y_i^5 - 156m^4y_i^4 - 48m^4y_i^6 + 72m^4y_i^3 - 576m^3y_i^5 + 96m^3y_i^2 - 12m^4y_i^2} \\ \overline{+192m^3y_i^6 + 36my_i^2 - 3}\Big)^{\frac{1}{3}} + \frac{2}{3}\Big(-6m^2y_i^3 + 3m^2y_i^2 - 5my_i^2 + 4m^2y_i^4 + 1\Big) \\ \cdot \frac{1}{my_i^2}\Big(-144m^3y_i^5 - 72m^2y_i^3 + 72m^3y_i^4 + 36m^2y_i^2 + 96m^2y_i^4 - 60my_i^2 + 64m^3y_i^6 + 8 \\ + 12my_i^2\sqrt{-36my_i + 18m - 156m^2y_i^2 + 12m^2y_i + 288m^2y_i^3 - 3m^2 - 144m^2y_i^4 + 672m^3y_i^4} \\ \overline{-384m^3y_i^3 + 144m^4y_i^5 - 156m^4y_i^4 - 48m^4y_i^6 + 72m^4y_i^3 - 576m^3y_i^5 + 96m^3y_i^2 - 12m^4y_i^2} \\ \overline{+192m^3y_i^6 + 36my_i^2 - 3}\Big)^{-\frac{1}{3}} + \frac{2my_i^2 + 1}{3my_i^2}\Bigg\}$$

Because this monstrous solution for θ_i is found, an alternative method will be considered in the next section.

A.2. MAXIMUM-LIKELIHOOD ESTIMATION OF $\boldsymbol{\phi}$

In the preceding section an estimate for $\theta_b = \int_0^{l_b} \phi_b(x)\,\mathrm{d}x$ was computed for every beam b. This gives us information about the corresponding line integral, but it does not take into account that X-ray beams can cross the defined grid diagonally, whereby only a part of each pixel that has to be reconstructed is passed through by the beam. See Figure 2.3.

Considering the two dimensional problem and writing the attenuation function $\phi(x)$ as column vector $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_N)^T$ can solve this problem. As discussed in section 2.3, weights a_{ij} have to be introduced that reflect the relation between the area that is illuminated by the beam and the entire area of the pixel:

$$a_{ij} = \frac{\text{illuminated area of pixel } j \text{ by ray } i}{\text{total area of pixel } j}$$

This leads to the following matrix-vector equation:

$$\mathbf{A}\boldsymbol{\phi} = \boldsymbol{\theta} \quad \text{with } \mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1N} \\ a_{21} & a_{22} & \ldots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \ldots & a_{MN} \end{pmatrix}, \boldsymbol{\phi} = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_N \end{pmatrix} \text{ and } \boldsymbol{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_M \end{pmatrix}.$$

Note that for beam b,

$$\theta_b = \sum_{j=1}^{N} a_{bj}\phi_j.$$

Using this notation, it is possible to compute a Maximum-Likelihood Estimate of $\boldsymbol{\phi}$. Define the log-likelihood of $\boldsymbol{\phi}$ as

$$\tilde{\ell}(\boldsymbol{\phi}; y_1, y_2, \ldots, y_M)$$

Taking the derivative of the log-likelihood with respect to ϕ_j results in

$$\frac{\partial\tilde{\ell}}{\partial\phi_j} = \sum_{k=1}^{M} \frac{\partial\ell}{\partial\theta_k}\frac{\partial\theta_k}{\partial\phi_j} = \sum_{k=1}^{M} \frac{\partial\ell}{\partial\theta_k}\,a_{kj} \\ = \sum_{k=1}^{M}\left(\frac{1}{2} - \frac{1}{2}\,\frac{1}{e^{\theta_k}-1} - \frac{my_i^2e^{3\theta_k} - 2my_i^2e^{2\theta_k} + m(2y_i-1)e^{\theta_k}}{2(e^{\theta_k}-1)^2}\right)a_{kj}$$

Naturally, a_{kj} is zero if a beam k does not contain pixel j, so the summation will be over less than M tubes.

Because solving $\frac{\partial\tilde{\ell}}{\partial\phi_j} = 0$ is even harder than solving $\frac{\partial\ell}{\partial\theta_i} = 0$ from the former section, there is decided to continue with different methods of estimating $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ instead of proceeding in this manner (see Chapter 4).

B

MEASURES OF CONVERGENCE

In Chapter 6 some comments were made on the convergence of the reconstruction algorithm. In this chapter, some additional plots are included to support decisions we have made during the reconstruction and to complete the results.

B.1. RELAXATION PARAMETER α

Several values of the relaxation parameter α are tested in the relaxation of the reconstruction of a slightly smaller phantom. In the Figures B.1, B.2 and B.3, the behaviour of the 'measures of convergence', i.e. the log-likelihood function, the relative error and the weighted sum of squared errors, is shown for different values of α.

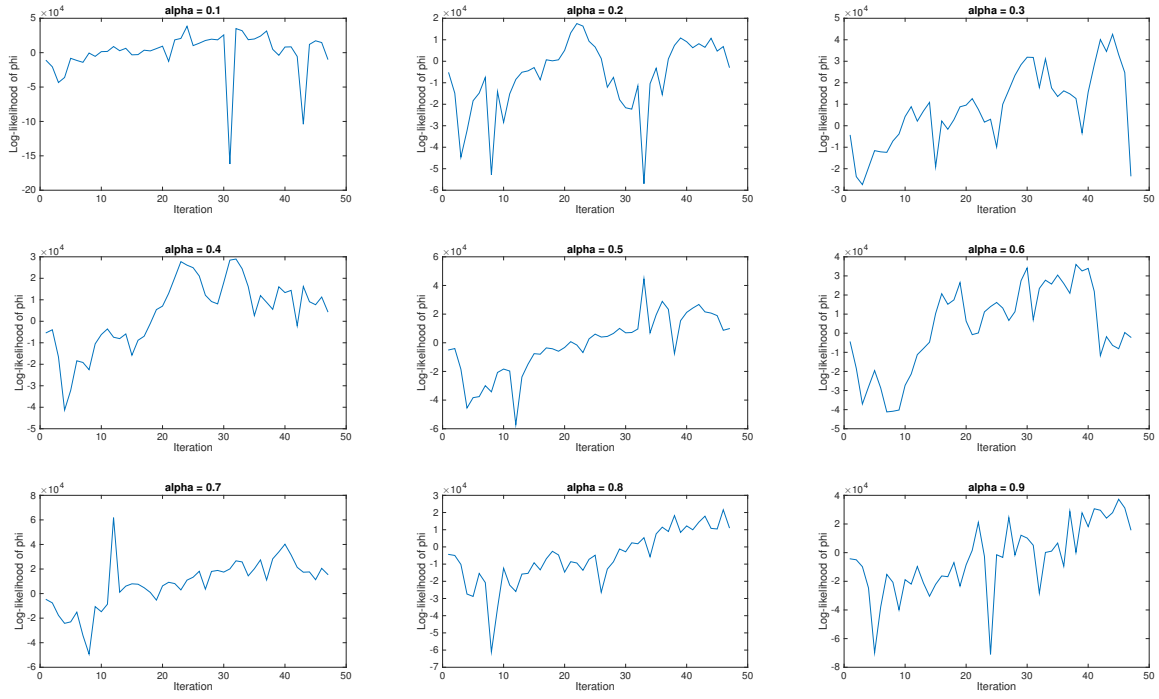


Figure B.1: Log-likelihood of $\boldsymbol{\phi}$, calculated in each iteration step, for different values of the relaxation parameter α.

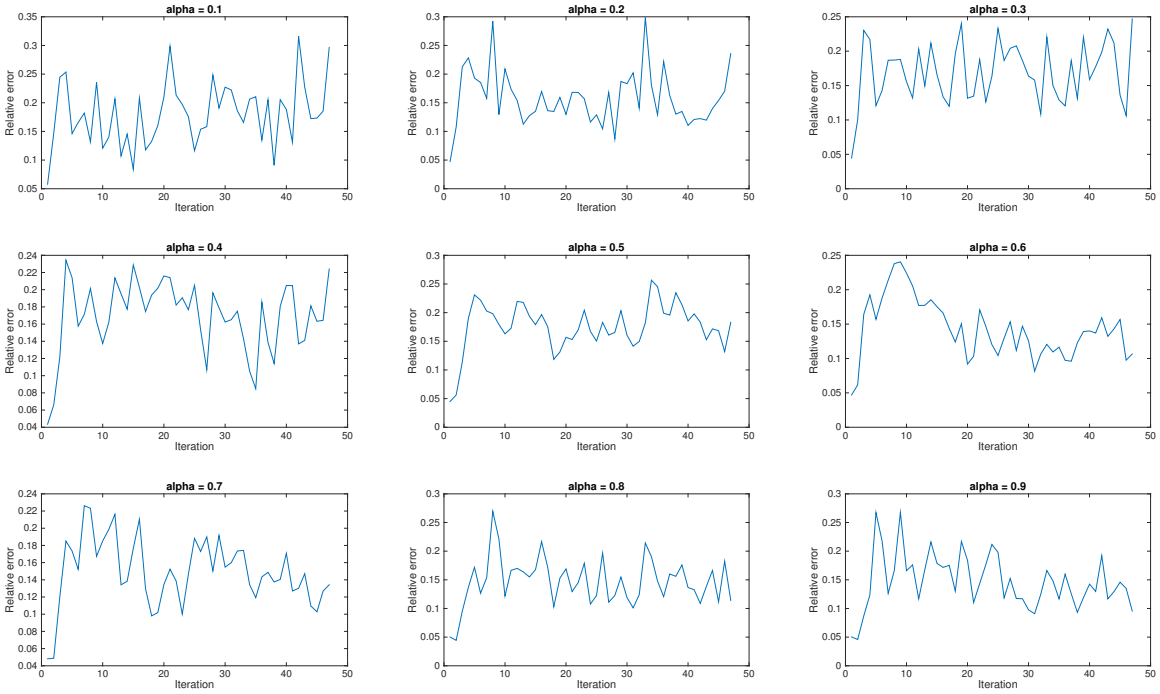


Figure B.2: Relative error, calculated in each iteration step, for different values of the relaxation parameter α.

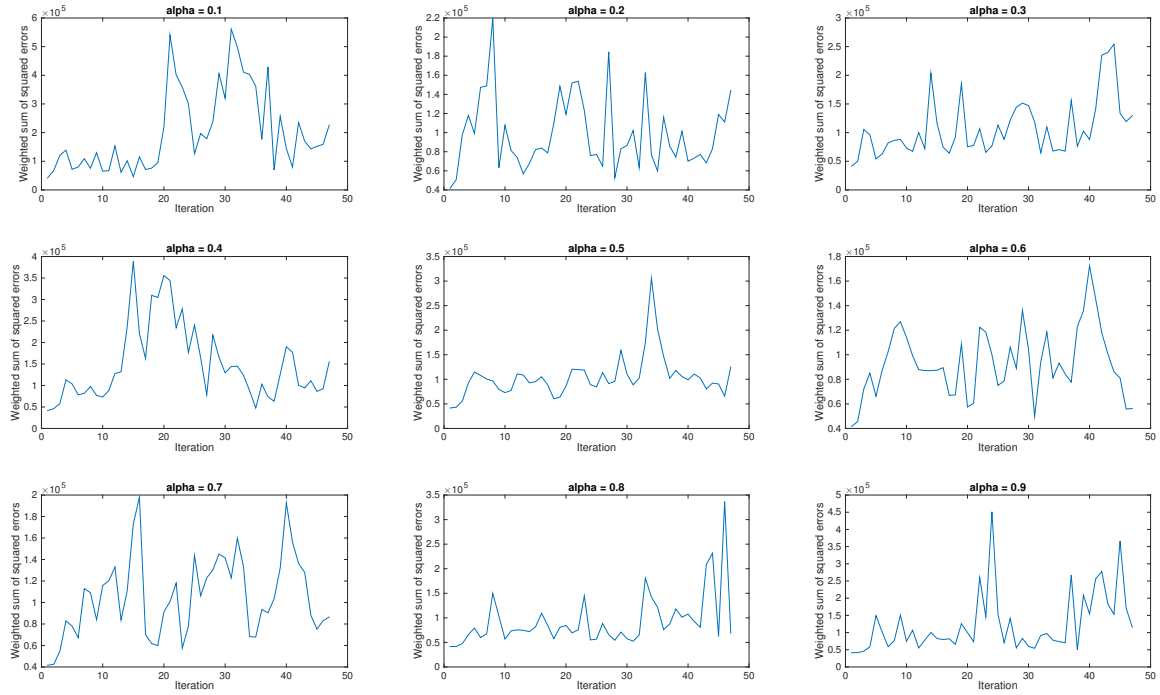


Figure B.3: Weighted sum of squared errors, calculated in each iteration step, for different values of the relaxation parameter α.

For convergence the log-likelihood of $\boldsymbol{\phi}$ should be increasing, while the relative error and the weighted sum of squared errors should be decreasing. Looking at the plots we see that for $\alpha = 0.7$ this more or less holds for the log-likelihood and the relative error. Therefore, this value of α is taken as the relaxation parameter in the reconstruction.

B.2. MEASURES OF CONVERGENCE FOR THE NEW RECONSTRUCTION METHOD

In this section the behaviour of the 'measures of convergence', i.e. the log-likelihood function, the relative error and the weighted sum of squared errors, are shown for the new reconstruction method.

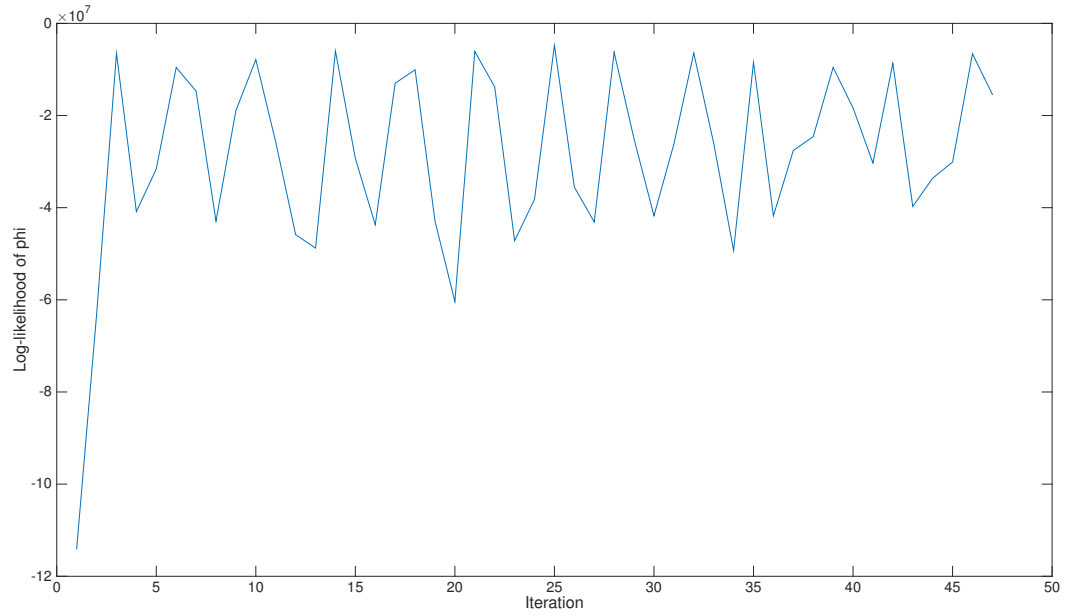


Figure B.4: Log-likelihood of $\boldsymbol{\phi}$ for the new reconstruction method, calculated in each iteration step.

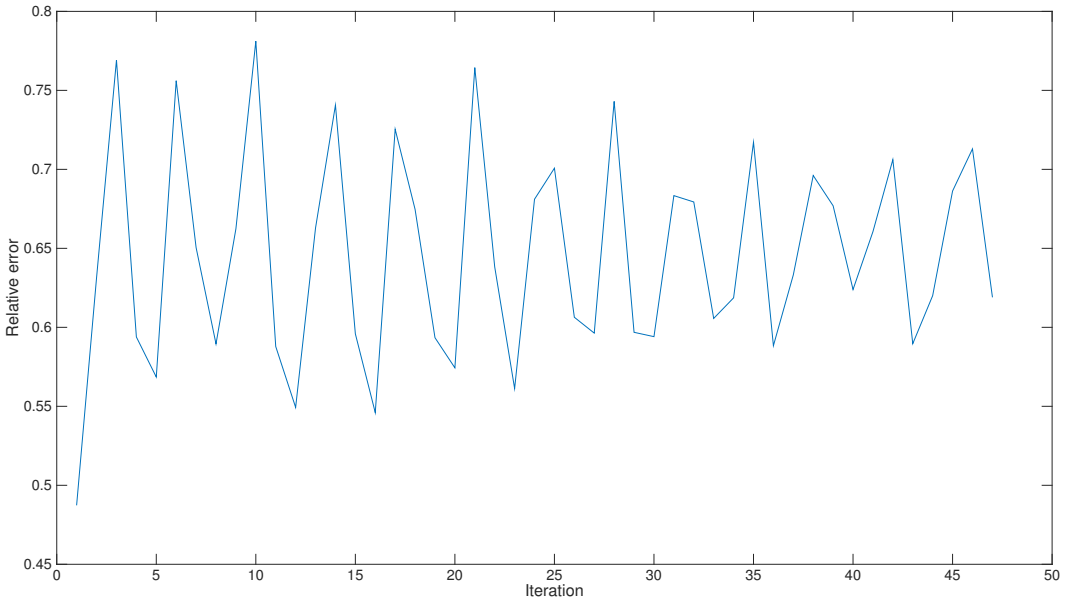


Figure B.5: Relative error for the new reconstruction method, calculated in each iteration step.

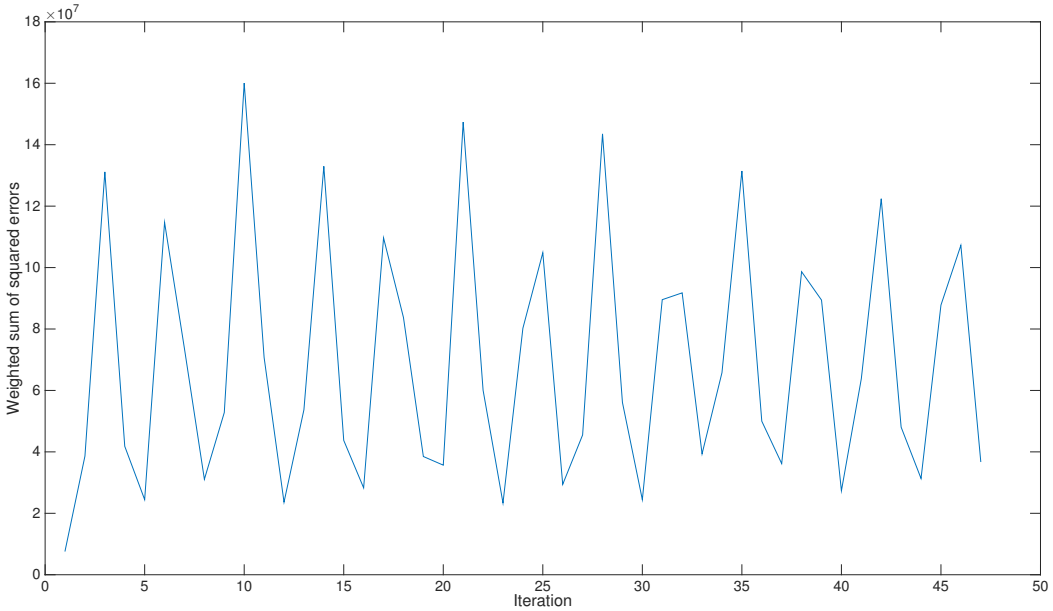


Figure B.6: Weighted sum of squared errors for the new reconstruction method, calculated in each iteration step.

When comparing these figures to Figure 6.10, 6.11 and 6.12, we see that a similar improvement is made on the convergence of the algorithm as in the relaxation method with exact variance.

C

MATLAB CODE

C.1. CURRENT RECONSTRUCTION METHOD

Here the Matlab code that is used as starting point for the experiments is given. Because the basis of the code is not changed in the experiments, only the adjustments are given in the next sections. In the code is specified where the adjustments should be inserted.
For explanation of the reconstruction methods, the reader is referred to Chapter 5 and 6.

```
%
%
% Programmed by Martin van Gijzen, April 29, 2015
%

clear all; close all;

scrsz = get(0,'ScreenSize');

fig1 = figure('Position',[scrsz(1) scrsz(4) scrsz(3) scrsz(4)]);
%fig1 = figure;
subplot(1,2,1)
im = phantom(50);
imshow(im);
title('Exact Shepp-Logan');

[m,n] = size(im);
N = m;
x_mod = reshape(im,N^2,1); % Exact solution for Ax = b

dtheta = 2;
p = round(sqrt(2)*N); % No. of parallel rays.
R = 2;                % Distance source to center
noise    = 0.1;
shift    = 0;
eps      = 1.e-6;
m_iter   = 100;

contin = 1;

old_dtheta = 0; old_p = 0; old_R = 0;
while ( contin )

   Title = 'Parameters';
```

```
   prompt = {'Beam angle spacing:','Rays per beam:','Distance source: ', ...
             'Noise:', 'Maximum iterations:'};
   defaults = {num2str(dtheta),num2str(p),num2str(R), ...
               num2str(noise),num2str(m_iter)};
   lineNo=[1 25];
   params=inputdlg(prompt,Title,lineNo,defaults);
   if ( isempty(params) ) contin = 0; break; end

   dtheta = str2num(char(params(1)));
   p      = str2num(char(params(2)));
   R      = str2num(char(params(3)));
   noise  = str2num(char(params(4)));
   m_iter = str2num(char(params(5)));

   compute_matrix = ( dtheta ~= old_dtheta | p ~= old_p | R ~= old_R );

   if ( compute_matrix )
      old_dtheta = dtheta ; old_p = p; old_R = R;
      theta = 0:dtheta:360-dtheta;      % No. of used angles.

      %Create the test problem.
      A = fanbeamtomo(N,theta,p,R);
   end

   [m,n] = size(A);

   d = A*x_mod;

   disp('Fanbeam projections');
   disp(['Number of rays = ',num2str(m)]);
   disp(['Number of pixels = ',num2str(n)]);
   disp(['Noise = ',num2str(noise)]);

   %%% INSERT ADJUSTMENTS HERE %%%

   if ( noise > 0 )
      randn('state',0);
      db = randn(size(d));
      pert = noise*norm(d)/norm(db);
      b = d + pert*db;
   else
      b = d;
   end

   [x_lsqr, flag, relres, iter] = lsqr( A, b, eps, m_iter );
   im = reshape(x_lsqr,N,N);
   subplot(1,2,2);
   imshow(im);
   title('LSQR reconstruction')
   disp(['LSQR: Relative error = ',num2str(norm(x_mod-x_lsqr)/norm(x_mod)), ...
         ' Iterations = ', num2str(iter)]);

   %%% END OF ADJUSTMENTS %%%

end
```

C.2. THEORETICAL NOISE DISTRIBUTION

```
%%% BEGIN OF ADJUSTMENTS %%%

np = 10000; % number of photons

if ( noise > 0 )
    mu = zeros(length(d),1);
    sigma2 = 1/np*(1-exp(-d))./exp(-d);
    randn('state',0);
    db = normrnd(mu,sigma2);
    pert = noise*norm(d)/norm(db);
    b = max(d + pert*db,0);
else
    b = d;
end

[x_lsqr, flag, relres, iter] = lsqr( A, b, eps, m_iter );

% image reconstruction
im = reshape(x_lsqr,N,N);
subplot(1,2,2);
imshow(im);
title('LSQR reconstruction')
disp(['LSQR: Relative error = ',num2str(norm(x_mod-x_lsqr)/norm(x_mod)), ...
      ' Iterations = ', num2str(iter)]);

%%% END OF ADJUSTMENTS %%%
```

C.3. EXPERIMENT 1: PATH LENGTH

```
%%% BEGIN OF ADJUSTMENTS %%%

np = 10000; % number of photons

if ( noise > 0 )
    mu = zeros(length(d),1);
    sigma2 = 1/np*(1-exp(-d))./exp(-d);
    randn('state',0);
    db = normrnd(mu,sigma2);
    pert = noise*norm(d)/norm(db);
    b = max(d + pert*db,0);
else
    b = d;
end

b = sparse(b);

% lsqr
[x_lsqr, flag, relres, iter] = lsqr( A, b, eps, m_iter );

% weighted lsqr: path length
path = zeros(m,1);
B = sparse(m,n);
c = zeros(m,1);
```

```
for k = 1:m
    path(k) = sum( A(k,:) );
    if path(k) < 0
        path(k) = 0;
    end
    if path(k) > 0
        B(k,:) = A(k,:)/path(k);
        c(k) = b(k)/path(k);
    end
    if path(k) == 0
        B(k,:) = 0;
        c(k) = 0;
    end
end

[x_pl, flag_pl, relres_pl, iter_pl] = lsqr( B, c, eps, m_iter );

% image reconstruction
im = reshape(x_lsqr,N,N);
subplot(1,3,2);
imshow(im);
title('LSQR reconstruction')
disp(['LSQR: Relative error = ',num2str(norm(x_mod-x_lsqr)/norm(x_mod)), ...
    ' Iterations = ', num2str(iter)]);

im = reshape(x_pl,N,N);
subplot(1,3,3);
imshow(im);
title('Weighted LSQR: path length')
disp(['Weighted LSQR (path length): Relative error = ',
    num2str(norm(x_mod-x_pl)/norm(x_mod)), ... ' Iterations = ', num2str(iter_pl)]);

%%% END OF ADJUSTMENTS %%%
```

C.4. EXPERIMENT 2: NON-CONSTANT EXACT VARIANCE

```
%%% BEGIN OF ADJUSTMENTS %%%

np = 10000; % number of photons

if ( noise > 0 )
    mu = zeros(length(d),1);
    sigma2 = 1/np*(1-exp(-d))./exp(-d);
    randn('state',0);
    db = normrnd(mu,sigma2);
    pert = noise*norm(d)/norm(db);
    b = max(d + pert*db,0);
else
    b = d;
end

b = sparse(b);
```

```
% lsqr
[x_lsqr, flag, relres, iter] = lsqr( A, b, eps, m_iter );

% weighted lsqr: non-constant variance (exact)
j = 1:m;
p = exp(-A(j,:)*x_mod);
p = min(1,p);
p = max(0,p);
var = 1/np*(1-p)./p;
D = sparse(m,n);
e = zeros(m,1);

for k = 1:m
    if var(k) < 0
        var(k) = 0;
    end
    if var(k) > 0
        D(k,:) = A(k,:)/sqrt(var(k));
        e(k) = b(k)/sqrt(var(k));
    end
    if var(k) == 0
        D(k,:) = 0;
        e(k) = 0;
    end
end

[x_ncve, flag_ncve, relres_ncve, iter_ncve] = lsqr( D, e, eps, m_iter );

% image reconstruction
im = reshape(x_lsqr,N,N);
subplot(1,3,2);
imshow(im);
title('LSQR reconstruction')
disp(['LSQR: Relative error = ',num2str(norm(x_mod-x_lsqr)/norm(x_mod)), ...
    ' Iterations = ', num2str(iter)]);

im = reshape(x_ncve,N,N);
subplot(1,3,3);
imshow(im);
title('Weighted LSQR: non-constant variance (exact)')
disp(['Weighted LSQR (exact non-constant variance): Relative error = ',
    num2str(norm(x_mod-x_ncve)/norm(x_mod)), ... ' Iterations = ',
    num2str(iter_ncve)]);

%%% END OF ADJUSTMENTS %%%
```

C.5. EXPERIMENT 3: ITERATIVE ALGORITHM

```
%%% BEGIN OF ADJUSTMENTS %%%

np = 10000; % number of photons

if ( noise > 0 )
    mu = zeros(length(d),1);
    sigma2 = 1/np*(1-exp(-d))./exp(-d);
```

```
    randn('state',0);
    db = normrnd(mu,sigma2);
    pert = noise*norm(d)/norm(db);
    b = max(d + pert*db,0);
else
    b = d;
end

b = sparse(b);

% lsqr
[x_lsqr, flag, relres, iter] = lsqr( A, b, eps, m_iter );

% weighted lsqr: non-constant variance (exact)
j = 1:m;
p = exp(-A(j,:)*x_mod);
p = min(1,p);
p = max(0,p);
var = 1/np*(1-p)./p;
D = sparse(m,n);
e = zeros(m,1);

for k = 1:m
    if var(k) < 0
        var(k) = 0;
    end
    if var(k) > 0
        D(k,:) = A(k,:)/sqrt(var(k));
        e(k) = b(k)/sqrt(var(k));
    end
    if var(k) == 0
        D(k,:) = 0;
        e(k) = 0;
    end
end

[x_ncve, flag_ncve, relres_ncve, iter_ncve] = lsqr( D, e, eps, m_iter );

% weighted lsqr: non-constant variance (exact, iterative)
x1 = [zeros(n,1), x_ncve];
i = 1;
maxi = 50;

while norm( x1(:,i)-x1(:,i+1) ) > eps && i < maxi
    p2 = exp(-A(j,:)*x1(:,i+1));
    p2 = min(1,p2);
    p2 = max(0,p2);
    var1 = 1/np*(1-p2)./p2;
    E = sparse(m,n);
    f = zeros(m,1);

    for k = 1:m
        if var1(k) < 0
            var1(k) = 0;
        end
        if var1(k) > 0
```

```
            E(k,:) = A(k,:)/sqrt(var1(k));
            f(k) = b(k)/sqrt(var1(k));
        end
        if var1(k) == 0
            E(k,:) = 0;
            f(k) = 0;
        end
    end

    [x1(:,i+2), flag_ncvei, relres_ncvei, iter_ncvei] = lsqr( E, f, eps, m_iter );
    i = i+1;

end

% b = log(I_0/I)
y = exp(-b);

pe = exp(-A(j,:)*x_ncve);
pe = min(1,pe);
pe = max(0,pe);
vare = 1/np*(1-pe)./pe;

sumsqr = zeros(maxi-3,1);
loglik = zeros(maxi-3,1);
rel_error = zeros(maxi-3,1);
for l = 1:maxi-3
    for q = 1:m
        if vare(q) > 0
            sumsqr(l) = sumsqr(l) + ( (b(q)-A(q,:)*x1(:,l+3))/sqrt(vare(q)) )^2;
        end
        loglik(l) = loglik(l) + ( -1/2*log(2*pi) - 1/2*log(1/np)
                    + 1/2*( A(q,:)*x1(:,l+3) + 10^(-5) )
                    - 1/2*log( 1-exp( ( -A(q,:)*x1(:,l+3) + 10^(-5) ) ) )
                    - ( np*(y(q) - exp( ( -A(q,:)*x1(:,l+3) + 10^(-5) ) ) )^2 )
                    /( 2*exp( ( -A(q,:)*x1(:,l+3) + 10^(-5) ) )
                    - 2*exp( -2*( A(q,:)*x1(:,l+3) + 10^(-5) ) ) ) );
    end
    rel_error(l) = norm( x_mod - x1(:,l+3) )/norm(x_mod);
end

% image reconstruction
im = reshape(x_lsqr,N,N);
subplot(1,3,2);
imshow(im);
title('LSQR reconstruction')
disp(['LSQR: Relative error = ',num2str(norm(x_mod-x_lsqr)/norm(x_mod)), ...
    ' Iterations = ', num2str(iter)]);

im = reshape(x1(:,end),N,N);
subplot(1,3,3);
imshow(im);
title('Weighted LSQR: non-constant variance (exact, iterative)')
disp(['Weighted LSQR (exact iterative non-constant variance): Relative error = ',
    num2str(norm(x_mod-x1(:,end))/norm(x_mod)), ... ' Iterations = ',
    num2str(iter_ncvei)]);
```

```
%%% END OF ADJUSTMENTS %%%
```

C.6. EXPERIMENT 4: RELAXATION

```
%%% BEGIN OF ADJUSTMENTS %%%

np = 10000; % number of photons

if ( noise > 0 )
    mu = zeros(length(d),1);
    sigma2 = 1/np*(1-exp(-d))./exp(-d);
    randn('state',0);
    db = normrnd(mu,sigma2);
    pert = noise*norm(d)/norm(db);
    b = max(d + pert*db,0);
else
    b = d;
end

b = sparse(b);

% lsqr
[x_lsqr, flag, relres, iter] = lsqr( A, b, eps, m_iter );

% weighted lsqr: non-constant variance (exact)
j = 1:m;
pe = exp(-A(j,:)*x_mod);
pe = min(1,pe);
pe = max(0,pe);
var = 1/np*(1-pe)./pe;
D = sparse(m,n);
e = zeros(m,1);

for k = 1:m
    if var(k) < 0
        var(k) = 0;
    end
    if var(k) > 0
        D(k,:) = A(k,:)/sqrt(var(k));
        e(k) = b(k)/sqrt(var(k));
    end
    if var(k) == 0
        D(k,:) = 0;
        e(k) = 0;
    end
end

[x_ncve, flag_ncve, relres_ncve, iter_ncve] = lsqr( D, e, eps, m_iter );

% weighted lsqr: non-constant variance (exact, iterative relaxation)
x1 = [zeros(n,1), x_ncve];
x_ncvei = [zeros(n,1), x_ncve];
i = 1;
maxi = 50;
alpha = 0.7;
```

```
while norm( x_ncvei(:,i)-x_ncvei(:,i+1) ) > eps && i < maxi
    p2 = exp(-A(j,:)*x_ncvei(:,i+1));
    p2 = min(1,p2);
    p2 = max(0,p2);
    var1 = 1/np*(1-p2)./p2;
    E = sparse(m,n);
    f = zeros(m,1);

    for k = 1:m
        if var1(k) < 0
            var1(k) = 0;
        end
        if var1(k) > 0
            E(k,:) = A(k,:)/sqrt(var1(k));
            f(k) = b(k)/sqrt(var1(k));
        end
        if var1(k) == 0
            E(k,:) = 0;
            f(k) = 0;
        end
    end

    [x1(:,i+2), flag_ncvei, relres_ncvei, iter_ncvei] = lsqr( E, f, eps, m_iter );
    x_ncvei(:,i+2) = alpha*x1(:,i+1) + (1-alpha)*x1(:,i+2);
    i = i+1;

end

% b = log(I_0/I)
y = exp(-b);

pe = exp(-A(j,:)*x_ncve);
pe = min(1,pe);
pe = max(0,pe);
vare = 1/np*(1-pe)./pe;

sumsqr = zeros(maxi-3,1);
loglik = zeros(maxi-3,1);
rel_error = zeros(maxi-3,1);
for l = 1:maxi-3
    for q = 1:m
        if vare(q) > 0
            sumsqr(l) = sumsqr(l) + ((b(q)-A(q,:)*x_ncvei(:,l+3))/sqrt(vare(q)))^2;
        end
        loglik(l) = loglik(l) + ( -1/2*log(2*pi) - 1/2*log(1/np)
                    + 1/2*( A(q,:)*x_ncvei(:,l+3) + 10^(-5) )
                    - 1/2*log( 1-exp( ( -A(q,:)*x_ncvei(:,l+3) + 10^(-5) ) ) )
                    - ( np*(y(q) - exp(( -A(q,:)*x_ncvei(:,l+3) + 10^(-5) )))^2 )
                    /( 2*exp( ( -A(q,:)*x_ncvei(:,l+3) + 10^(-5) ) )
                    - 2*exp( -2*( A(q,:)*x_ncvei(:,l+3) + 10^(-5) ) ) );
    end
    rel_error(l) = norm( x_mod - x_ncvei(:,l+3) )/norm(x_mod);
end
```

```
% image reconstruction
im = reshape(x_lsqr,N,N);
subplot(1,3,2);
imshow(im);
title('LSQR reconstruction')
disp(['LSQR: Relative error = ',num2str(norm(x_mod-x_lsqr)/norm(x_mod)), ...
    ' Iterations = ', num2str(iter)]);

im = reshape(x_ncvei(:,end),N,N);
subplot(1,3,3);
imshow(im);
title('Weighted LSQR: non-constant variance (exact, iterative relaxation)')
disp(['Weighted LSQR (exact iterative relaxation non-constant variance):
    Relative error = ',num2str(norm(x_mod-x_ncvei(:,end))/norm(x_mod)), ...
    ' Iterations = ', num2str(iter_ncvei)]);

%%% END OF ADJUSTMENTS %%%
```

C.7. EXPERIMENT 5: NEW RECONSTRUCTION METHOD

```
%%% BEGIN OF ADJUSTMENTS %%%

np = 10000; % number of photons

if ( noise > 0 )
    mu = zeros(length(d),1);
    sigma2 = 1/np*(1-exp(-d))./exp(-d);
    randn('state',0);
    db = normrnd(mu,sigma2);
    pert = noise*norm(d)/norm(db);
    b = max(d + pert*db,0);
else
    b = d;
end

b = sparse(b);

% lsqr
[x_lsqr, flag, relres, iter] = lsqr( A, b, eps, m_iter );

% weighted lsqr: non-constant variance (not-exact)
j = 1:m;
pe = exp(-A(j,:)*x_lsqr);
pe = min(1,pe);
pe = max(0,pe);
var = 1/np*(1-pe)./pe;
D = sparse(m,n);
e = zeros(m,1);

for k = 1:m
    if var(k) < 0
        var(k) = 0;
    end
    if var(k) > 0
        D(k,:) = A(k,:)/sqrt(var(k));
```

```
        e(k) = b(k)/sqrt(var(k));
    end
    if var(k) == 0
        D(k,:) = 0;
        e(k) = 0;
    end
end

[x_ncvne, flag_ncvne, relres_ncvne, iter_ncvne] = lsqr( D, e, eps, m_iter );

% weighted lsqr: non-constant variance (not-exact, iterative relaxation)
x1 = [zeros(n,1), x_ncvne];
x_ncvnei = [zeros(n,1), x_ncvne];
i = 1;
maxi = 50;
alpha = 0.7;

while norm( x_ncvnei(:,i)-x_ncvnei(:,i+1) ) > eps && i < maxi
    p2 = exp(-A(j,:)*x_ncvnei(:,i+1));
    p2 = min(1,p2);
    p2 = max(0,p2);
    var1 = 1/np*(1-p2)./p2;
    E = sparse(m,n);
    f = zeros(m,1);

    for k = 1:m
        if var1(k) < 0
            var1(k) = 0;
        end
        if var1(k) > 0
            E(k,:) = A(k,:)/sqrt(var1(k));
            f(k) = b(k)/sqrt(var1(k));
        end
        if var1(k) == 0
            E(k,:) = 0;
            f(k) = 0;
        end
    end

    [x1(:,i+2), flag_ncvnei, relres_ncvnei, iter_ncvnei] = lsqr(E,f,eps,m_iter);
    x_ncvnei(:,i+2) = alpha*x1(:,i+1) + (1-alpha)*x1(:,i+2);
    i = i+1;

end

% b = log(I_0/I)
y = exp(-b);

% for weighted sum of squared errors
pe = exp(-A(j,:)*x_ncvne);
pe = min(1,pe);
pe = max(0,pe);
vare = 1/np*(1-pe)./pe;

sumsqr = zeros(maxi-3,1);
loglik = zeros(maxi-3,1);
```

```
rel_error = zeros(maxi-3,1);
for l = 1:maxi-3
    for q = 1:m
        if vare(q) > 0
            sumsqr(l) = sumsqr(l) + ((b(q)-A(q,:)*x_ncvnei(:,l+3))/sqrt(vare(q)))^2;
        end
        loglik(l) = loglik(l) + ( -1/2*log(2*pi) - 1/2*log(1/np)
                    + 1/2*( A(q,:)*x_ncvnei(:,l+3) + 10^(-5) )
                    - 1/2*log( 1-exp( ( -A(q,:)*x_ncvnei(:,l+3) + 10^(-5) ) ) )
                    - ( np*(y(q) - exp(( -A(q,:)*x_ncvnei(:,l+3) + 10^(-5) )))^2 )
                    /( 2*exp( ( -A(q,:)*x_ncvnei(:,l+3) + 10^(-5) ) )
                    - 2*exp( -2*( A(q,:)*x_ncvnei(:,l+3) + 10^(-5) ) ) ) );
    end
    rel_error(l) = norm( x_mod - x_ncvnei(:,l+3) )/norm(x_mod);
end

% image reconstruction
im = reshape(x_lsqr,N,N);
subplot(1,3,2);
imshow(im);
title('LSQR reconstruction')
disp(['LSQR: Relative error = ',num2str(norm(x_mod-x_lsqr)/norm(x_mod)), ...
    ' Iterations = ', num2str(iter)]);

im = reshape(x_ncvnei(:,end),N,N);
subplot(1,3,3);
imshow(im);
title('Weighted LSQR: non-constant variance (not-exact, iterative relaxation)')
disp(['Weighted LSQR (not-exact iterative relaxation non-constant variance):
    Relative error = ',num2str(norm(x_mod-x_ncvnei(:,end))/norm(x_mod)), ...
    ' Iterations = ', num2str(iter_ncvnei)]);

%%% END OF ADJUSTMENTS %%%
```