Exercise 4 – Look inside the forest

Mathematical Modelling
02526

The goal with this exercise is to use mathematical modelling to solve a practical problem: Finding shots in wood logs. You should:

- Identify significant issues
- Formulate mathematical models, which solves these issues
- Implement the model in a computer program
- Use the implemented model for analysis of data
- Report the results from the analysis and discuss the validity of the model

The report for this exercise must be 5 pages at most, including graphs, tables, and images (excluding frontpage and appendices). In this exercise a number of questions are asked. The report, however, should not be a list of answers, but instead be a coherent documentation and discussion of the analysis performed.

The exercise consists of several parts. The first is understanding Computed Tomography and how we can treat that mathematically. The next is understand what is necessary from a CT system, the solve the customers problem, and design such a CT system. Finally you will either try to validate your system through simulation or work on a better reconstruction method.

To solve the exercise, we suggest you use MATLAB or PYTHON.

Detecting shots

A company that produces hardwood floors wants your help with designing a system, that can detect shots in the logs they cut up. Due to hunting in the forests, shots are often found embedded in the logs.

Historically lead has been used for shots. However, due to the toxicity of lead, lead shots have been banned in Denmark since 1996. Instead, steel shots are often used. This created a problem for the wood industry as 1) Steel is harder than lead and damages the saw blades 2) steel rusts and discolours the wood. Due to these challenges, new shots were developed based on bismuth. Bismuth and lead are much softer and do not discolour the wood, but bismuth is much more costly than steel. Shots come in sizes ranging from 4mm diameter (Size 1) down to 2 mm diameter (Size 8).

This means that we want a system that can detect steel shots down to 2 mm in diameter. Ideally, the system can also tell us if the shot is steel or lead/bismuth, as less care has to taken around the latter.

If they can be detected and located before cutting, the cuts can be changed to maximise the amount of useable wood.

For detection and location they want to use Computed Tomography (CT). It is a technique where x-ray images are taken from several different angles and then reconstructed into a tomographic image of the inside of the object.

X-ray and Computed Tomography

In tomography we project a number of X-ray beams through a sample, and measure the attenuation of the radiation due to the material that the ray passes through. The attenuation of the beam depends on the properties of the material along the ray, and we can thus – given enough beams – “calculate backwards” from the measured attenuation and reconstruct the material properties inside the sample.

This an example of of an inverse problem[1] where we have data (measurements) and a mathematical model, and want to estimate the object that gave rise to the data. We will here derive the underlying model for the tomography problem.

Attenuation of a X-ray beam We consider a X-ray beam which passes through a sample along a straight line, and we let ℓ denote the length of the ray inside the sample. The material has the spatially dependant attenuation coefficient x, and we let $I(\ell)$ denote the intensity of the beam as a function of ℓ. It can then be shown [Buzug, p. 32] that $I(\ell)$ satisfies **Lambert-Beer's law** in the form of this differential equation:

$$\frac{\mathrm{d}I}{\mathrm{d}\ell} = -x\, I(\ell)\ . \tag{1}$$

After the beam has passed through the sample it therefore has the intensity

$$I = I_0\, e^{-\int_0^{\ell_{\max}} x\, \mathrm{d}\ell}\ , \tag{2}$$

where I_0 is the intensity of the beam before it passes through the material and $\ell_{\max}$ is the total length of the beam through the sample. In the special case where the material is homogeneous with a spatially independent attenuation coefficient x_0 we get

$$I = I_0\, e^{-x_0\, \ell_{\max}}\ . \tag{3}$$

Consider using what you have learned in previous courses to derive the results in equations (2) and (3) to check your understanding

Simplification of the model We will now simplify the model in such a way that we can use it in our numerical computations. Let us assume that the X-ray beam passes through p different homogeneous areas with attenuation coefficients $x_1, x_2, \ldots, x_p$. The intensity of the beams after exiting the sample is then given by

$$I = I_0\, e^{-\sum_{j=1}^{p} x_j\, \ell_j}\ , \tag{4}$$

where $\ell_1, \ell_2, \ldots, \ell_p$ are the lengths of the ray's passage through each of the areas. Can you explain how this follows from equation (2).

If we take the logarithm on both sides, we obtain the following equation which is linear in the attenuation coefficients:

$$\sum_{j=1}^{p} x_j\, \ell_j = \log(I_0/I)\ . \tag{5}$$

The quantity $b = \log(I_0/I)$ is easy to calculate from the measurements, and below we will consider it as the data for the given ray. By transforming the problem and thus linearising it, we simplify the reconstruction problem and make it easier to solve.

Reconstruction model In tomographic reconstruction the attenuation coefficients of the material are unknown – they must be determined from the measurements. We must therefore formulate a reconstruction model that links these attenuation coefficients to the measured data.

We will start by defining the geometry. We wish to calculate the material properties in a thin slice through the material sample. It thus makes sense to only project the X-ray beams through this slice. We will further assume an infinitesimal thickness of the slice, thus reducing the problem to a square 2-dimensional one, with side lengths $L \times L$.

[1]The theory of inverse problems, is outside the scope of this exercise; more details can be found in [1] and [2].

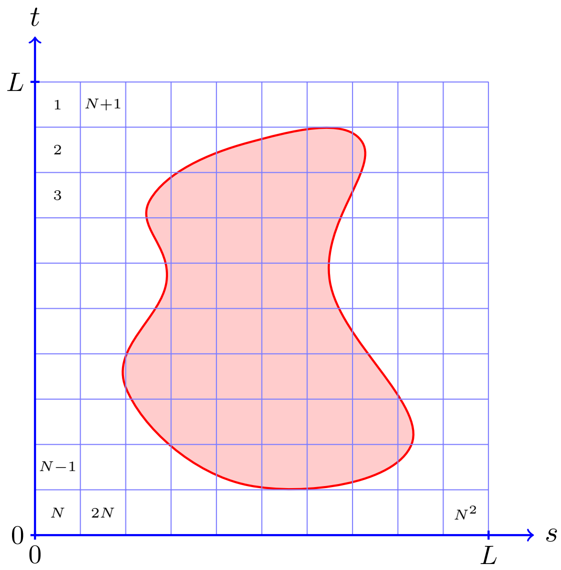

Figure 1: Coordinate system and grid with indexing of the unknowns.

We now discretize the problem from equation (2). We divide the slice into a $N \times N$ grid of pixels, each with the length $D = L/N$, and we assume that each pixel is so small that we in each of them can consider the attenuation coefficient as a constant. Our unknowns are thus the values of the $n = N^2$ attenuation coefficients in the respective pixels, which we will index as shown in Figure 1, such that our unknowns become a vector $\boldsymbol{x}$ with $n = N^2$ elements.

If we consider ray number i with associated data b_i, it follows from (5) that

$$\sum_{j \in \mathcal{S}_i} x_j \, \ell_{ij} = b_i \,, \qquad i = 1, 2, \ldots, m \,, \tag{6}$$

where m is the number of ray, $\mathcal{S}_i$ is the set of indices for the pixels that are hit by ray i, and ℓ_{ij} is the length of this ray through the jth pixel. Can you explain the meaning of this equation?

We can now simplify our notation considerably by defining that all pixels which are not hit by the ray are assigned the ray length 0. It then follows that for each of the m rays we can write

$$\sum_{j=1}^{n} a_{ij} \, x_j = b_i \,, \qquad i = 1, 2, \ldots, m \,, \tag{7}$$

where we have defined

$$a_{ij} = \begin{cases} \ell_{ij} & \text{if beam } i \text{ hits pixel } j \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Equation (7) is in reality a system of equations $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$. Consider why.

Simple configuration with two angles In our apparatus we use a CCD camera which we can assume to be a 1-dimensional array with N pixels placed outside the slice. We first consider a very simple measurement situation with only two projections, where the CCD is placed along the two edges of the slice as shown in Figure 2. This results in $m = 2N$ data values. We assume that the pixel size of the CCD is such that each pixel is hit exactly by one X-ray. Moreover, each X-ray is parallel with the edges and passes through exactly N pixels, either along the s-axis or the t-axis as shown in Figure 2. This results in a linear system of equations $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$ where the matrix $\boldsymbol{A}$ is rectangular with $m = 2N$ rows and $n = N^2$ columns – i.e. the system of equations is severely underdetermined.

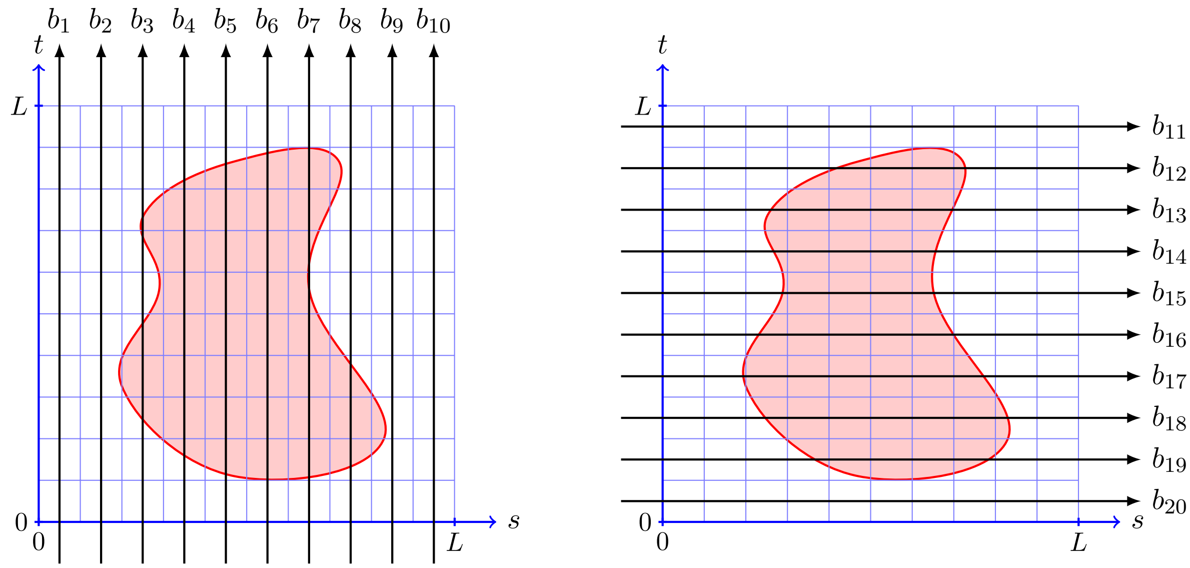

Figure 2: Configuration with $N = 10$ and two projections.

Consider a ray parallel to the t-axis through $s = \frac{1}{2}\Delta L$, i.e., it passes through pixels corresponding to $j = 1, 2, \ldots, N$. The corresponding row in the $\boldsymbol{A}$ matrix thus has the elements

$$\underbrace{\Delta L, \Delta L, \ldots, \Delta L}_{N \text{ elements}}, \underbrace{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,\ldots,0}_{N^2 - N \text{ elements}}$$

The next ray, which is also parallel to the t-axis through $s = \frac{3}{2}\Delta L$, hits pixels corresponding to $j = N+1, N+2, \ldots, 2N$ and the corresponding row in the $\boldsymbol{A}$ matrix thus has the elements

$$\underbrace{0,0,0,0,0,\ldots,0}_{N \text{ elements}}, \underbrace{\Delta L, \Delta L, \ldots, \Delta L}_{N \text{ elements}}, \underbrace{0,0,0,0,0,0,0,0,0,0,0,0,0,\ldots,0}_{N^2 - 2N \text{ elements}}$$

Consider what the entire $\boldsymbol{A}$ matrix looks like

Configuration with arbitrary CCD and arbitrary angles We need to send more rays through the slice to obtain more measurements.

The computational model thus needs to be extended, such that

1. it allows for an arbitrary number of projections (corresponding to a number of arbitrary angles for the rays),
2. an arbitrary number of rays P (one for each pixel on the CCD), and
3. a fixed distance of $(N-1)\Delta L$ between the two outer beams

This functionality is already implemented. Check out Paralleltomo (PYTHON version on Learn) and make sure you understand how it works by comparing it to the two-angle situation above.

As an alternative for the direct solution of $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$ - which can be very time consuming - many methods exist. Have a look at AIRToolsII and the Python version.

Part 1 - Shot detection and differentation

You have been supplied with a simple test image: testImage.mat / testImage.npy. The image is 5000×5000 pixels, and is meant to simulate a 50 cm log with a 2 mm steel and a 2 mm lead shot embedded.

I.e., a physical pixel size of 0.1 mm.

Downsample the image to 50×50 pixels. Create a system matrix $\boldsymbol{A}$ with fifty rays $P = 50$ using Paralleltomo. Try with default settings initially.

You can now simulate a forward projection through the test image by $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{A}$ is the system matrix from Paralleltomo and $\boldsymbol{x}$ is the downsampled image unfolded to a vector.

Try to reconstruct the image $\boldsymbol{x}$ from $\boldsymbol{A}$ and $\boldsymbol{b}$ (mldivide in MATLAB and numpy.linalg.solve in PYTHON). In reality, a projection is not perfect. Try to add noise to your projections, i.e., to the vector $\boldsymbol{b}$, before reconstructing. Does it change the results?

Try out different settings of Paralleltomo (angles, number of rays, etc.), different noise levels, and different resolutions of the test image.

Energies and Resolution

Assume a maximum log size of 0.5 m, i.e., $L = 0.5\,m$. *For ease of computation, assume parallel rays.* The slice thickness can be assumed to be N/L.

Commercial x-ray sources are typically between 10 keV - 200 keV. Consider the attenuation of wood and the size of the log. What x-ray energy would you use for the system?

Is it also the best energy for detecting shots and distinguishing steel and lead/bismuth?

What should N be if we want to

- detect 2 mm steel pellets
- distinguish steel pellets from lead/bismuth pellets?

Think back to physics and compare the attenuation coefficients of steel (iron), bismuth and wood. NIST is an excellent source for attenuation coefficients. Alternatively, make a test image by inputting the attenuation values corresponding to different energy levels, and you can then try to reconstruct it at various resolutions.

Part 2 - Configuring the system

Both an x-ray source and a detector is expensive. Given your results from Part 1, design a CT system for the customer. I.e., how few angles and how few rays in each angle, can we use and still get acceptable reconstructions?

Consider ways to compare different setups. You may, e.g., use the condition number of $\boldsymbol{A}$, or the reconstruction of a test image with some noise added.

Validate the design on artificial data, such that you know the ground truth. Test the sensitivity to noise. Consider what type of noise and how much noise to use. Explain your choices.

Part 3 - validation or improvement?

You can now choose between two tasks:

1) Validation of your setup by simulating data and making sure you can reliably detect shots.

Simulate as realistic data as possible[2] and simulate the noise levels using different acquisition times (see Poisson noise). Is your system robust, or do you gain new insights that lead to design changes?

[2] See e.g. this link

2) Improving the system through a better reconstruction method.

Perform a literature search. There are many different ways to do tomographic reconstruction, some of which are implemented in AIRToolsII. You may also consider if the solution of the system $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$ is best done by minimising the 2-norm. Could we do something else to see the shots better (see CVX or CVXPY)?

Reporting **Contents of the report**

1. Describe the problem and the background – what is modelled and why?
2. Describe data and experiments
3. Describe the mathematical model - how and why?
4. Describe results
5. Discuss results – how good are the results? To what degree do they reflect the true problem?
6. Conclude – what is the contribution of the analysis?

Hand in The report is uploaded to DTU Learn

1. R. C. Aster, B. Borchers, and C. H. Thurber, *Parameter Estimation and Inverse Problems*, 2nd Edition, Academic Press, 2012.
2. C. W. Groetsch, *Inverse Problems: Activities for Undergraduates*, Mathematical Associated of Americal, 1999.
3. C. B. Moler, *What is the Condition Number of a Matrix*, `https://blogs.mathworks.com/cleve/2017/07/17/what-is-the-condition-number-of-a-matrix/`

Anders Nymark Christensen 2023

Partly based on material by Per Christian Hansen
Special thanks to Hans Martin Kjer for revision and critical comments

References