MEASURES AND LINEAR INVERSE PROBLEMS | MScT CSE
Martin Rodriguez-Vega | Year: 2025–2026

PRACTICAL WORK 2

1 Introduction

During the practical work sessions, the use of Python to solve the proposed exercises is recommended. Remember to save your work, as it may be used in the next sessions.

Documents and useful scripts can be found in the course repository:

https://github.com/m93rodriguez/BioMed-Inversion-Course

A report were you describe your methodology and provided answers for the exercises for all practical work sessions is expected after the end of the ourse. Send your work to:

martin.RODRIGUEZ-VEGA@univ-amu.fr

Please abstain to use LLM or AI tools. Previous years have shown that false responses are very common. Do not hesitate to ask questions during the session.

2 X-Ray Tomography

In this section, we are analyze the absorption coefficient in 2D: $\mu_a(\mathbf{r}) = \mu(x, y)$. In X-ray tomography, scattering in negligible, so the steady-state light propagation model is

$$\mathbf{u} \cdot \nabla L(\mathbf{r}, \mathbf{u}) + \mu_a(\mathbf{r}) L(\mathbf{r}, \mathbf{u}) = 0 \tag{1}$$

which has analytical solution

$$L(\mathbf{r}_0 + \ell \mathbf{u}, \mathbf{u}) = L(\mathbf{r}_0, \mathbf{u}) \exp\left(- \int_0^{\ell} \mu(\mathbf{r}_0 + x\mathbf{u}) \mathrm{d}x \right) \tag{2}$$

where the Radiance is known at some point $\mathbf{r}_0$.

EXERCISE 2.1 : Prove that (2) satisfies (1) for the 1D case.

HINTS : Recall that $\mathbf{u} \cdot \nabla$ is equivalent to the directional derivative, and that in 1D there is only one direction. Use the Fundamental Theorem of Calculus.

The Radon Transform is the analytical method to obtain the sinogram of an object of interest. It is defined as

$$R_\mu(\theta, s) = \int_{-\infty}^{\infty} \mu(s\cos\theta - t\sin\theta, s\sin\theta + t\cos\theta) \mathrm{d}t \tag{3}$$

Let the sinogram be denoted as $p(\theta, t) = R_\mu(\theta, s)$. Suppose we discretize the sinogram into a matrix of dimensions n_s, n_θ, and that we discretize space into n_x, n_y points. Then, we can approximate the Radon transform with a matrix:

$$\mathbf{p} = W\boldsymbol{\mu} \tag{4}$$

where $\mathbf{p}$ is an unwrapped version of the sinogram, and $\boldsymbol{\mu}$ is an unwrapped version of the absorption coefficient.

EXERCISE 2.2 : Is W a constant matrix of does it depend on μ? Why? Show that the Radon Transform is a linear operator.

EXERCISE 2.3 : What are the dimension of W in terms of n_s, n_θ, n_x, n_y?

In the repository, in `Scripts`, you will find Python module `xray.py` which contains functions `make_sinogram` and `radon_matrix`.

EXERCISE 2.4 : Generate some sample domains by defining the absorption coefficient in a 2D plane. Use `make_sinogram` function to generate its measure. Plot and compare the 2D matrices.

EXERCISE 2.5 : Function `radon_matrix` constructs matrix W in (4). Compute the sinogram using (4) and compare the outputs with was obtained in the previous exercise.

HINTS : In Python, see the documentation of `numpy` methods `ravel, flatten, ravel_multi_index, unravel_index, reshape`.

3 Linear Inversion problems

Inverse problems can often be written as an optimization problem

$$\min_{\mathbf{x}} \quad d(\mathbf{y}, f(\mathbf{x})) \tag{5}$$

where $\mathbf{y}$ is a vector of measurements, $\mathbf{x}$ is a vector of parameters, f is the forward model, and d is a distance or error function. When f is a linear function and d is the square norm of the error we get the least squares problem

$$\min_{\mathbf{x}} \quad ||\mathbf{y} - A\mathbf{x}||^2 \tag{6}$$

Although this has analytical solution $\mathbf{x} = A^+\mathbf{y}$ where A^+ is the pseudo-inverse, for very large systems this is too computationally expensive to solve analytically.

Iterative methods give an approximate solution to this problem. Here, we consider two methods:

- Algebraic Reconstruction Technique (ART). Developped for X-ray tomography, finds successive solutions

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda \frac{y_i - A_i \cdot \mathbf{x}^k}{||A_i||^2} A_i^\top \tag{7}$$

  where A_i is the i-th row of A

- Gradient descent. Finds successive solutions to minimize some function $g(\mathbf{x})$:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \eta \nabla g(\mathbf{x}^k) \tag{8}$$

  For the case of linear least squares, we have $\nabla g(\mathbf{x}) = 2A^\top(A\mathbf{x} - \mathbf{y})$

EXERCISE 2.6 : Write a Python function that implements the ART algorithm.

EXERCISE 2.7 : Write a Python function that implements the Gradient descent algorithm.

EXERCISE 2.8 : In the `Scripts` directory, you will find two sinograms, either in image or `numpy` array form. Test your inversion algorithms to reconstruct the corresponding $\mu(x, y)$ and show your results.

HINTS : For `sinogram1`, you can use a reconstruction size of between 50 and 100 pixels per dimension, and for `sinogram2` between 100 and 200 pixels. Use `radon_matrix` to generate the W matrix. If needed, you can reduce the number of θ and s by taking 1 every 2 or 3 samples instead.