Math 262 / CME 372 **Winter 2021**

Homework 3

Due Tuesday, March 9

In this assignment you will explore some inversion techniques for the Radon transform and their relations to the Fourier transform. These have direct applications in reconstruction of images in computerized tomography (CT). This project will be mostly computational.

Since we are interested in evaluating the effectiveness of the methods in recovering a true continuous object, we will use an analytical phantom for which you can evaluate its values, and the values of its Radon transform, up to machine precision. The phantom will be a function f_0 supported on the set $\Omega_X = [-1/2, 1/2] \times [-1/2, 1/2]$ which can be seen in Fig. 1.

Methods for inverting the Radon transform

We will study some basic approaches to the inversion of the Radon transform. Our data consist on samples of the Radon transform of f_0 on an equispaced grid over $\Omega_R = [-1/2, 1/2] \times [0, \pi]$. Note there is no need *a priori* to sample for values of t with $|t| > 1/2$ as f_0 is supported on a ball centered at the origin of radius $1/2$. In what follows we will use a discretization of Ω_R using an equispaced grid with $N_t = 256$ and $N_\theta = 64$ points $\{(t_k, \theta_\ell)\}_{k,\ell}$ given by

$$t_k = \frac{k - 1/2}{N_t - 1}, \quad k = -\frac{N_t}{2}, \ldots, \frac{N_t}{2} - 1 \qquad \text{and} \qquad \theta_\ell = \frac{\pi \ell}{N_\theta - 1}, \quad \ell = 0, \ldots, N_\theta - 1.$$

We will denote as $\boldsymbol{Y}_0$ the matrix obtained by sampling the Radon transform of the analytical phantom on this grid. You can get the data from any of the two following files in Canvas (both containing the same data in different formats):

- `radon_data.mat` which is a Matlab file with the array `Y0` containing the samples.
- `radon_data.dat` which is a dat file with the samples.

In both cases rows represent the samples with θ fixed, and columns represent the samples with the t variable fixed. The values of θ increase top to bottom, and the values of t increase left to right.

The methods we will study attempt to reconstruct an $N \times N$ image with $N = 128$. This image will correspond to a discretization over the equispaced grid $\mathcal{C}$ with points

$$\boldsymbol{x}_{n_1, n_2} = \left(\frac{n_1 + 1/2}{N}, \frac{n_2 + 1/2}{N} \right), \qquad n_1, n_2 = -\frac{N}{2}, \ldots, \frac{N}{2} - 1.$$

1 Direct Fourier inversion

Here we will use the fact that sampling the Radon transform allows us to estimate the Fourier transform of f_0. Recall the projection-slice theorem relates the Radon transform of f_0 to its Fourier transform as follows

$$\int \mathcal{R} f_0(t, \theta) e^{-irt} \, dt = \hat{f}_0(r \cos\theta, r \sin\theta).$$

Therefore, one could try to estimate the value of $\hat{f}_0$ over some non-equispaced grid on the frequency domain, and then attempt to use a NUFFT to invert the Radon transform. We will explore this approach in this section.

(a) What is the shape of the non-equispaced grid $\mathcal{P}$ over which you can estimate $\hat{f}_0$?

(b) At which points can you approximate $\hat{f}_0$ over a ray defined by any of the N_θ angles used for sampling $\mathcal{R}f_0$? How would you compute these values efficiently? Implement the proposed method. *Hint:* Recall the object is contained on a ball centered at the origin with radius $1/2$ and $\mathcal{R}f_0(t,\theta) = 0$ for $|t| > 1/2$.

(c) We would like to solve the system

$$\hat{f}_0(\boldsymbol{\omega}) = \sum_{\boldsymbol{x}\in\mathcal{C}} X_{\boldsymbol{x}} e^{-i\langle \boldsymbol{\omega}, \boldsymbol{x}\rangle}, \qquad \boldsymbol{\omega} \in \mathcal{P}. \tag{1}$$

In order to do this we need to implement

$$\boldsymbol{X} \mapsto \sum_{\boldsymbol{x}\in\mathcal{C}} X_{\boldsymbol{x}} e^{-i\langle \boldsymbol{\omega}, \boldsymbol{x}\rangle}, \qquad \boldsymbol{\omega} \in \mathcal{P}. \tag{2}$$

with adjoint

$$\boldsymbol{Y} \mapsto \sum_{\boldsymbol{\omega}\in\mathcal{P}} Y_{\boldsymbol{\omega}} e^{i\langle \boldsymbol{\omega}, \boldsymbol{x}\rangle}, \qquad \boldsymbol{x} \in \mathcal{C}. \tag{3}$$

Using the ideas presented in Lecture 12 to compute NUFFTs for 1D signals, implement fast algorithms for the forward (2) and adjoint (3) transforms.

(d) With the fast algorithms implemented in (c) use an iterative method to solve (1) using, for instance, CG. For example, if you want to solve the linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ in Matlab for $\boldsymbol{A}$ symmetric, positive semi-definite matrix, you would use the command

```
x = pcg( Afun, b );
```

where `Afun` is a handle to a function such that `Afun(x)` computes the matrix-vector product $\boldsymbol{A}\boldsymbol{x}$. For instance you can use the sample code

```
n = 10; p = 5;
X = randn(n,p);
A = X'*X;
b = randn(p,1);
x0 = pcg(@(x) A*x, b);
```

If you want to do the same in Python with the packages numpy and scipy, you can use the commands

```
import numpy as np
from scipy.sparse.linalg import LinearOperator, cg
n = 10
p = 5
X = np.random.normal(size=(n,p))
A = X.T@X
b = np.random.normal(size=(p,1))
Aop = LinearOperator((p,p), matvec= lambda x : A@x)
x0, _ = cg(Aop,b)
```

Solve the system (1) for the approximation of $\hat{f}_0$ over $\mathcal{P}$ you obtained from the method proposed in (b). How does the reconstruction look like? What happens to the edges? Discuss.

(e) **Bonus question:** If you are familiar with preconditioning of linear system, can you come up with a preconditioner to solve (1)?

Figure 1: The analytical phantom sampled on a 1024×1024 grid.

(f) Try to perform a reconstruction for $\boldsymbol{Y}$ sampled from a $\textsc{Poisson}(\lambda \boldsymbol{Y}_0)$ distribution, where $\lambda \|\boldsymbol{Y}_0\|_\infty = 10^3$ and

$$\|\boldsymbol{Y}_0\|_\infty = \max_{k,\ell} |Y_{0,k,\ell}|.$$

This means that the entries of $\boldsymbol{Y}$ are independent and distributed $Y_{n_1,n_2} \sim \textsc{Poisson}(\lambda Y_{0,n_1,n_2})$. For instance, you can generate this data in Matlab using the command

```
Y = poissrnd( lambda * Y0 )
```

and you can do the same in Python with the command

```
Y = np.random.poisson( lambda * Y0 )
```

What happens when you try to solve the system? How does the result look like? Try the experiment for $\lambda \|\boldsymbol{Y}_0\|_\infty = 10^6$. What happens in this case?

2 Backprojection and filtered backprojection

Here we will explore reconstruction techniques that rely on the discrete analogues of the adjoint and inverse of the Radon transform. Recall the backprojection formula is

$$\mathcal{R}^* g(\boldsymbol{x}) = \int_0^\pi g(\langle \boldsymbol{x}, \boldsymbol{\theta} \rangle, \theta)\, d\theta.$$

A straightforward discretization of the above integral over θ would not work, as it is very unlikely that $\langle \boldsymbol{x}_{n_1,n_2}, \boldsymbol{\theta}_\ell \rangle = t_k$ for some indices $n_1, n_2 = 1, \ldots, N$, $k = 1, \ldots, N_t$, and $\ell = 1, \ldots, N_\theta$. Therefore, we need to **interpolate** the data in the t variable by using a suitable kernel before discretizing. This will be the approach we will take here.

(a) Discuss how you would choose an interpolation kernel for discretizing the backprojection formula. Can you give a geometric interpretation of the interpolation kernel? *Hint:* What is the geometric interpretation of the backprojection formula?

(b) Write a routine that takes as input the $N_t \times N_\theta$ matrix $\boldsymbol{Y}$ containing the samples $\{g(t_k, \theta_\ell)\}_{k,\ell}$ and computes the values of the backprojected data on a $N \times N$ grid constructed as described before. Apply your routine to the data $\boldsymbol{Y}_0$. How does the reconstruction look like? What happens to the edges?

(c) The filtered backprojection requires convolving $\mathcal{R}f_0$ with a known filter h in the t variable[1]. However, you only know $\boldsymbol{Y}_0$ and the grid over which $\mathcal{R}f_0$ was evaluated. How would you approximate the convolution? Use your approximation, apply your filter, and describe how the filtered data differs from the original data. *Hint:* Recall the filter amplifies high frequencies.

(d) Compute the backprojection of the filtered data. How does the reconstruction look like? What happens to the edges? Discuss.

(e) Repeat the last step with $\boldsymbol{Y}$ sampled from a $\textsc{Poisson}(\lambda \boldsymbol{Y}_0)$ distribution, where $\lambda \|\boldsymbol{Y}_0\|_\infty = 10^3$. What happens when you try to solve the system? How does the result look like? Try the experiment for $\lambda \|\boldsymbol{Y}_0\|_\infty = 10^6$. What happens in this case?

[1] See Theorem 2 in the notes for Lecture 10.