Project 2 - X-ray Tomography

1 Introduction

Computational tomography consists of determining an image of an object from its projections. In X-ray tomography, properties related to the object's density (more accurately, the product of the density and the absorption cross-section of the material) are inferred from multiple projections of the object being scanned. These projections are essentially the Radon transform of the structure of the object along the line γ, which is given by

$$y \equiv -\log\left(\frac{I}{I_0}\right) = \int_{\gamma(l)} \rho(\gamma(l))dl, \tag{1}$$

where we actually measure the attenuated intensity I of the projected intensity I_0 of X-rays, which is related to the (unknown) object density ρ by Beer-Lambert law.

2 Toy problem

Let us start by considering a toy problem of an X-ray tomography experiment. The toy problem consists of a 5×5 grid to model a rectangular cross-section of an object, with sources that generate X-ray energy, and receivers to collect the energy, located around the grid's boundary as depicted in Figure 1.

We shall denote by $\mathbf{X}$ the sampled density on a $(\tilde{x}, \tilde{y})$-grid (i.e., $X_{ij} = \rho(\tilde{x}_j, \tilde{y}_i)$), and by $\mathbf{x}$ its column stack representation, according to the following convention for cell numbering:

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & X_{13} & \dots & X_{1N} \\ X_{21} & X_{22} & X_{23} & \dots & X_{2N} \\ \dots & \dots & \dots & \dots & \dots \\ X_{M1} & X_{M2} & X_{M3} & \dots & X_{MN} \end{bmatrix}$$

$$\mathbf{x} = \left[X_{11}, X_{21}, X_{31}, \quad \dots \quad, X_{12}, X_{22}, X_{32}, \quad \dots \quad, X_{1N}, X_{2N}, X_{3N}, \quad \dots \quad, X_{NN}\right]^{\mathrm{T}}$$

The model parameters are the entries of $\mathbf{X}$. In order to evaluate the integral in (1), we will apply the simple midpoint rule:

$$\int_a^b f(\tilde{x})d\tilde{x} \approx \sum_{j=1}^{n} f\left(\tilde{x}_j\right)\Delta\tilde{x} \tag{2}$$

$$\tilde{x}_j = a + \frac{\Delta\tilde{x}}{2} + (j-1)\Delta\tilde{x}$$

$$\Delta\tilde{x} = \frac{b-a}{n}$$

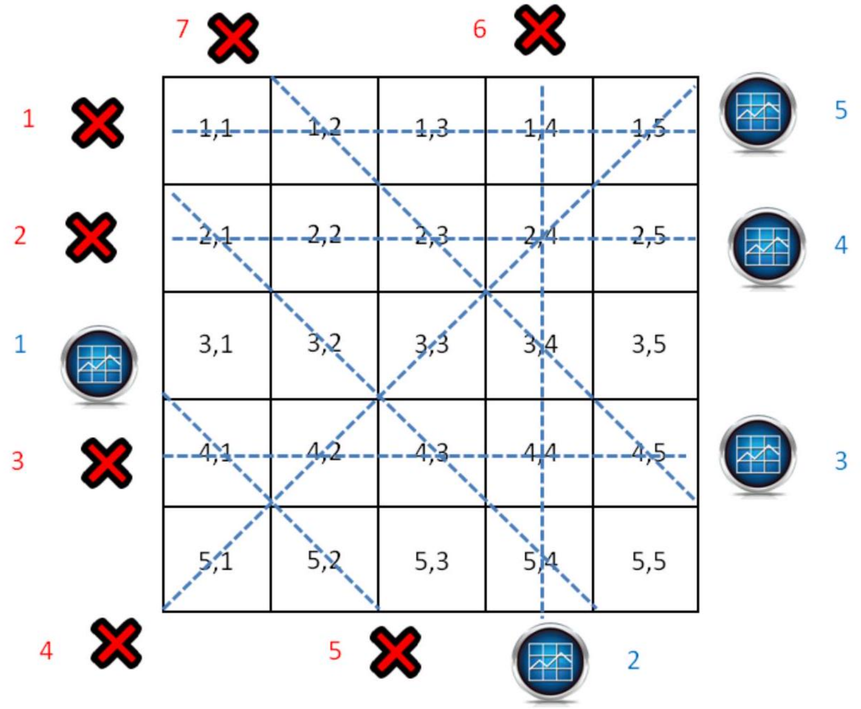

Figure 1: Schematic for the toy problem. X-ray energy is generated at source points (X icons) and collected at the receivers (Circle icons). Sources are numbered counter-clockwise in red numbers. Receivers are numbered counter-clockwise in blue numbers. The ray path matrix $\mathbf{A}$ can be inferred from the schematic. For example, the element of $\mathbf{A}$ that corresponds to the path traversed in cell [2, 1] by the ray that goes from source 2 to receiver 2 is $\sqrt{(\Delta\tilde{x})^2 + (\Delta\tilde{y})^2}$.

Q1. Apply the midpoint rule to discretize the integral in (1). Remember that the integration is performed along straight lines from source to receiver that can be horizontal, vertical, or diagonal (define Δl accordingly in terms of $\Delta\tilde{x}$ and $\Delta\tilde{y}$). Do not confuse the spatial coordinates notation $\tilde{x}, \tilde{y}$ with the measurements $\mathbf{y}$ and the sampled density field $\mathbf{x}$.

Q2. Using the approximation you obtained, write the relation between the vector of measurements $\mathbf{y}$, the *ray-path matrix* $\mathbf{A}$, and the density field $\mathbf{x}$ (in its column stack representation). Refer to Figure 1 for the placements of sources, receivers, and ray paths, and build the matrix $\mathbf{A}$ (the element $A_{i,j}$ in $\mathbf{A}$ is the distance traversed by the ray that produces measurement i, in the cell that corresponds to element j in $\mathbf{x}$). Since $\mathbf{A}$ is known, of what type is the relation between $\mathbf{x}$ and $\mathbf{y}$?

A very frequently used operation on scalar fields sampled on regular grids (e.g., density field) is the calculation of (finite difference approximation of) its partial derivatives along the horizontal and vertical dimensions. For simplicity, consider the **forward difference approximation** of the derivative in the vertical direction:

$$(\partial_{\tilde{y}}\mathbf{X})_{i,j} = X_{i+1,j} - X_{i,j}$$

for $i \in \{1, \dots, M-1\}$. You can assume that the derivative at $i = M$ is zero. The horizontal derivative $\partial_{\tilde{x}}\mathbf{X}$ is obtained in the same way by switching the roles of the indices i and j.

Q3 (code). Given the dimension of $\mathbf{X}$, i.e., (M, N), constructs two matrices $\mathbf{D}_{\tilde{x}}$ and $\mathbf{D}_{\tilde{y}}$ such that the multiplications $\mathbf{D}_{\tilde{x}}\mathbf{x}$ and $\mathbf{D}_{\tilde{y}}\mathbf{x}$ produce the column-stack representation of $\partial_{\tilde{x}}\mathbf{X}$ and $\partial_{\tilde{y}}\mathbf{X}$, respectively.

Download from the website the following files, which are examples of density fields:

- X1.mat
- X2.mat
- X3.mat

Q4 (code). Apply the derivatives to the set of example density fields. Visualize the images representing $\partial_{\tilde{x}}\mathbf{X}$ and $\partial_{\tilde{y}}\mathbf{X}$ and make sure that they make sense. Also, compute the magnitude of the gradient $G_{ij} = \sqrt{(\partial_{\tilde{x}}\mathbf{X})_{ij}^2 + (\partial_{\tilde{y}}\mathbf{X})_{ij}^2}$. What can you see in this image?

One of the challenges in tomographic reconstruction is that the number of unknowns highly exceeds the number of observations (projections along straight lines). Such reconstruction tasks are called ill-posed inverse problems.

Q5. Referring to Figure 1, what is the number of unknown parameters? How many observations are available? In this setting, what can you say about the set of solutions (minimizers) of the problem

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2,$$

Is it a singleton (has a single element)? Is it convex?

A common way to tackle ill-posed problems is to use a regularization term, such as the classical **Tikhonov regularization**. This technique is based on formulating the problem of estimating $\mathbf{x}$ from $\mathbf{y}$ as the unconstrained minimization of

$$\min_{\mathbf{x} \in \mathbb{R}^{MN}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \frac{\lambda}{2}\|\mathbf{L}\mathbf{x}\|_2^2, \qquad (3)$$

where $\mathbf{L}$ is a linear operator, and $\lambda > 0$ is a parameter. A common choice for $\mathbf{L}$ is

$$\mathbf{L} = \begin{bmatrix} \mathbf{D}_{\tilde{x}} \\ \mathbf{D}_{\tilde{y}} \end{bmatrix}.$$

The first term is interpreted as the data fitting term, while the second one serves as a regularization. The regularization term introduces bias in order to favor solutions that correspond

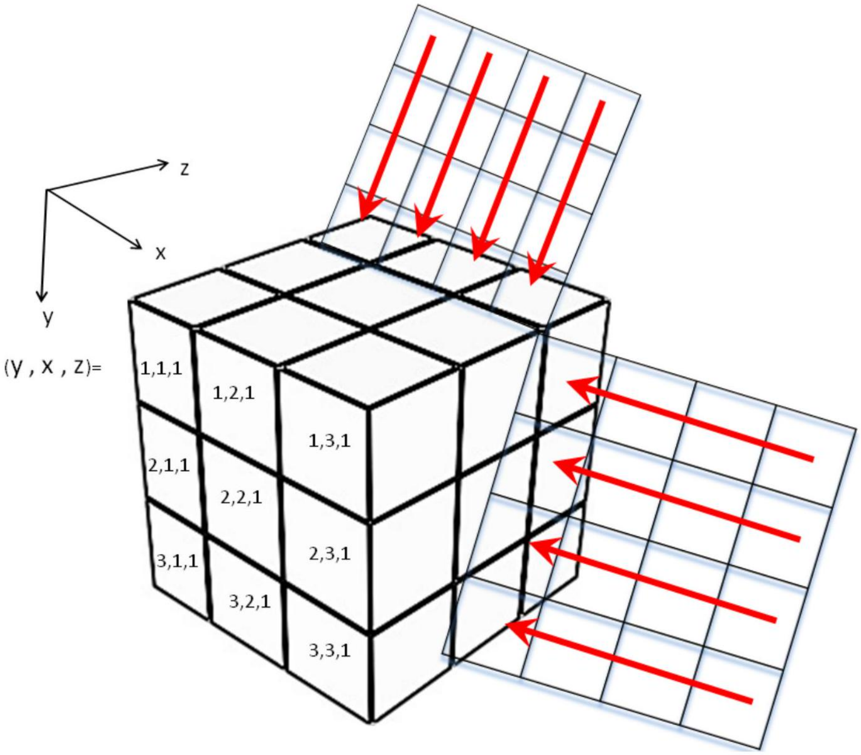

Figure 2: Schematic for the X-ray tomography experiment. The volume is discretized on a grid of $n \times n \times n$ voxels, and the projections are taken along slices, discretized into $m \times m$ pixels. Each slice produces m observations. For our experiment $m = n$. The figure shows example of two such slices, each one of them gives 4 projections/observations. The numbering is again according to the aforementioned convention, where for the column stack representation we stack slices along the $\tilde{z}$ direction.

to smaller values of $\|\mathbf{L}\mathbf{x}\|_2$, which for the above choice of $\mathbf{L}$ means more smoothness. The amount of bias can be controlled by the parameter λ.

Problem (3) can be compactly written as the following least squares problem:

$$\min_{\mathbf{x} \in \mathbb{R}^{MN}} \frac{1}{2} \left\| \begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{L} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 \tag{4}$$

Q6. Apply the first order optimality condition to problem (3) to get a set of linear equations (these are known as the *normal equations*). Find a simple closed-form solution to (3) expressed in terms of λ, $\mathbf{A}$ and $\mathbf{L}$.

3 Large scale unconstrained optimization

A typical application of an X-ray scanner requires scanning from multiple directions, where as with the toy problem above, the number of observations is much smaller than the number of density values you would like to infer (otherwise you would spend several minutes on scanning a single object). This means that without appropriate regularization, there is no chance to get a meaningful solution.

The scanning takes place in a 3D environment, and the object is reconstructed as a volume on a 3D grid of *voxels* (see Figure 2). Unfortunately, computing the closed form solution you devised above is not feasible due to the size of the problem, and one has to resort to an iterative procedure.

Q7. Bring (4) to the form

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c, \quad \mathbf{Q} \succ 0. \tag{5}$$

Explain why the expression that equals $\mathbf{Q}$ in our case obeys $\mathbf{Q} \succ 0$ (rather than just $\mathbf{Q} \succeq 0$). What does it imply on the number of solutions?

Q8. State the gradient descent (GD) algorithm for minimizing (5). Explicitly write the gradient in terms of λ, $\mathbf{A}$ and $\mathbf{L}$, but you can keep the step-size undefined.

Q9. Consider $\mathbf{A}$ associated with the toy problem with $\Delta\tilde{x} = \Delta\tilde{y} = 1$, $\lambda = 10^{-5}$ and $\mathbf{L} = \begin{bmatrix} \mathbf{D}_{\tilde{x}} \\ \mathbf{D}_{\tilde{y}} \end{bmatrix}$. Explain what values of the step-size ensure convergence of GD that is applied on (5). What is the condition number of $\mathbf{Q}$? How many GD iterations will be required to reduce the objective by a factor of 10? **You are allowed to use exiting functions that compute eigenvalues.**

The method of Conjugate-Gradients (CG) can be used to efficiently solve linear least squares problems like (4). Although you can plug (5) as is into the CG method you learned in class, the special structure of a least-squares problem suggests a modification of CG that is better in both time and memory aspects.

Q10 (code). Conjugate Gradient Least Squares (CGLS) — Adapt the method of Conjugate-Gradients to minimize problem (4) with $\Delta\tilde{x} = \Delta\tilde{y} = 1$. In your adaptation, keep in mind the following guidelines:

- In a typical tomography problem, the matrix $\mathbf{A}$ is very sparse. On the other hand, $\mathbf{A}^T\mathbf{A}$ might not be sparse. Avoid explicitly computing $\mathbf{A}^T\mathbf{A}$ (same goes for $\mathbf{L}$ and $\mathbf{L}^T\mathbf{L}$)
- In order to decrease round off errors when computing residuals of the form

  $$\mathbf{A}^T\mathbf{A}\mathbf{x} - \mathbf{A}^T\mathbf{y}$$

  factor $\mathbf{A}^T$ out first and compute:

  $$\mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{y})$$

- Use the following recursive relation:

$$\mathbf{s}_{k+1} \equiv \mathbf{A}\mathbf{x}_{k+1} - \mathbf{y} = \mathbf{A}\left(\mathbf{x}_k + \alpha_k \mathbf{d}_k\right) - \mathbf{y} = \left(\mathbf{A}\mathbf{x}_k - \mathbf{y}\right) + \alpha_k \mathbf{d}_k = \mathbf{s}_k + \alpha_k \mathbf{A}\mathbf{d}_k$$

- For large scale problems, you don't want the computation to proceed endlessly (plus, due to inevitable round-off errors, you may not find an exact solution anyway). Introduce a tolerance τ and a bound $i_{\max}$ on the number of iterations such that the algorithm will stop ahead of time if it reached the required tolerance or the maximum number of iterations.

Using the above guidelines, you should end up with an algorithm that requires only one vector-matrix multiplication with $\mathbf{A}$ and one vector-matrix multiplication with $\mathbf{A}^T$ per iteration.

Download from the website the following file:

- `Y.mat` - A matrix containing the observations for the toy problem. $\mathbf{Y}[i, j]$ is the measurement at receiver j of the energy generated by source i. Refer to Figure 1.

Apply the method of conjugate gradients to solve the equations you got in the toy problem, with

$$\mathbf{L} = \begin{bmatrix} \mathbf{D}_{\tilde{x}} \\ \mathbf{D}_{\tilde{y}} \end{bmatrix}$$

and $\lambda = 10^{-5}$ (you need to rearrange $\mathbf{Y}$ according to the way you constructed $\mathbf{A}$). How many iterations did you expect, and how many were required?

It is now time to test your algorithm on real data. Think about this as giving your prototypical method to security personnel in a test site airport to perform security screening "on the fly" when suspicion arises. A suspicious gentleman with two bags was chosen as the first candidate for the test.

In the website you will find two ZIP files:

- `Small.zip` - for the small bag, which is a volume of size 19 × 19 × 19
- `Large.zip` - for the large bag, which is a volume of size 49 × 49 × 49

Each file contains the following Matlab data files:

- `y.mat` – Normalized attenuation, which corresponds to path integrals of the density through the volume, contaminated with zero mean relative Gaussian noise level of 5% (i.e., if $\mathbf{e}$ is the noise vector and $\mathbf{x}$ is the true solution, then $\|\mathbf{e}\|_2 = 0.05 \cdot \|\mathbf{x}\|_2$). Element y_i contains the projection of the density field along ray i.

- `A.mat` – Ray path matrix. Each row represents a projection direction parallel to a slice, whose normal direction was randomly generated from a uniform distribution on the unit sphere. Multiplying by the matrix produces projections along 30 2D slices, each slice is discretized into $n \times n$ pixels (see Fig. 2).

For the next sections, test your implementations on the data within `Small.zip`.

Before solving the 3D problem, you will need to construct the derivative matrices to account for the 3D space. Construct three matrices $\mathbf{D}_{\tilde{x}}, \mathbf{D}_{\tilde{y}}, \mathbf{D}_{\tilde{z}}$ such that the multiplication $\mathbf{D}_{\tilde{x}}\mathbf{x}, \mathbf{D}_{\tilde{y}}\mathbf{x}$ and $\mathbf{D}_{\tilde{z}}\mathbf{x}$ produces the column-stack representation of $\partial_{\tilde{x}}\mathbf{X}$, $\partial_{\tilde{y}}\mathbf{X}$, and $\partial_{\tilde{z}}\mathbf{X}$, respectively. You can use the forward difference scheme for simplicity. For example:

$$\left(\partial_{\tilde{y}}\mathbf{X}\right)_{i,j,k} = X_{i+1,j,k} - X_{i,j,k}$$

and so on. You can assume that the derivative along the boundary is zero.

Q11. Reconstruct the content of the small bag by solving problem (3) (in its equivalent form) with the CGLS algorithm. Use

$$\mathbf{L} = \begin{bmatrix} \mathbf{D}_{\tilde{x}} \\ \mathbf{D}_{\tilde{y}} \\ \mathbf{D}_{\tilde{z}} \end{bmatrix}$$

Show the solution and state its objective value for several values of λ. (See the Tips sections for display options).

4 A better regularization

Since contents of bags are usually discrete objects, the density within the bag can be modeled as a piecewise smooth function. Therefore it is beneficial to use a different regularization scheme, that will bias your solution towards piecewise-smooth density functions.

Consider the following nonlinear optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^{n^3}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \alpha\|\mathbf{L}\mathbf{x}\|_1, \quad \alpha > 0 \tag{6}$$

When $\mathbf{L}$ is the approximation of the gradient operator ∇, the second term is known as the *anisotropic* Total Variation, and this regularization scheme is a variant of **Total Variation Regularization**.

In order to motivate the use of this more complicated regularization scheme, consider the following one dimensional problem of fitting a curve that passes closest to a collection of noisy sample points (see Fig. 3). The marked points are:

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 \end{bmatrix} \\ f_1(\mathbf{x}) &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0.5 & 1 & 1 & 1 & 1 \end{bmatrix} \\ f_2(\mathbf{x}) &= \begin{bmatrix} 0 & 0.0025 & 0.0180 & 0.1192 & 0.5000 & 0.8808 & 0.9820 & 0.9975 & 1 \end{bmatrix} \end{aligned}$$

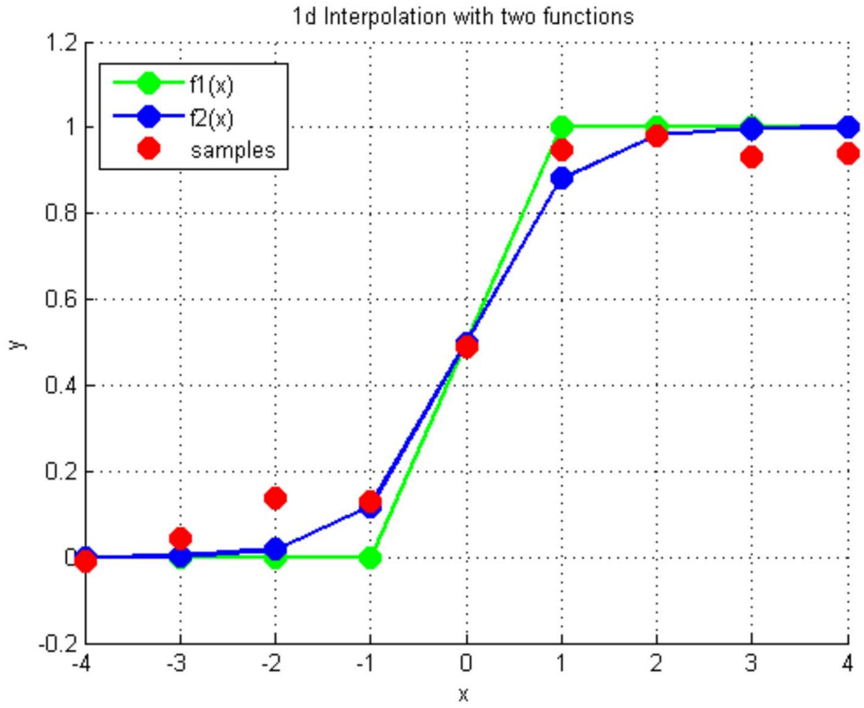

Figure 3: Curve fitting. $f_2(x)$ is smoother than $f_1(x)$, which makes it more favorable than $f_1(x)$ in terms of $\|\nabla f(x)\|_2^2$. This shows that $\|\nabla f(x)\|_2^2$ biases solutions to be smoother, and therefore is less appropriate when the data is piecewise smooth or piecewise constant. On the other hand, $\|\nabla f(x)\|_1$ is the same for either reconstruction.

Q12. Compute $\|\mathbf{D}_{\mathrm{x}}f(\mathbf{x})\|_1$ for both proposed interpolations in the figure, based on the samples at the marked points. Is there a preferable reconstruction (in terms of minimal value of $\|\mathbf{D}_{\mathrm{x}}f(\mathbf{x})\|_1$)? Do the same for $\|\mathbf{D}_{\mathrm{x}}f(\mathbf{x})\|_2$. Is there a preferable reconstruction (in terms of minimal value of $\|\mathbf{D}_{\mathrm{x}}f(\mathbf{x})\|_2$)?

This simple example hints that the term $\|\mathbf{D}_{\tilde{x}}\mathbf{x}\|_2$ favors solutions that vary more smoothly than those favored by $\|\mathbf{D}_{\tilde{x}}\mathbf{x}\|_1$.

Q13. Is problem (6) convex? Strictly convex? Smooth?

We will see how to solve an approximate problem using an **Iteratively Reweighted Least Squares** (IRLS) method.

Q14.

1. Show that for every $\gamma_i \neq 0$
$$\frac{\partial \|\mathbf{Lx}\|_1}{\partial x_k} = \sum_{i=1}^{m} L_{i,k} \frac{\gamma_i}{|\gamma_i|}$$
with $\gamma_i = (\mathbf{Lx})_i$.

2. Show that $\nabla(\|\mathbf{Lx}\|_1) = \mathbf{L}^T\mathbf{WLx}$, with $\mathbf{W}$ a diagonal matrix, and find explicitly the entries of $\mathbf{W}$. In order to accommodate for the non-differentiability at $\gamma_i = 0$, introduce a tolerance ϵ, such that $W_{i,i} = \frac{1}{\epsilon}$ whenever $|\gamma_i| < \epsilon$.

3. Use the expression for $\nabla(\|\mathbf{Lx}\|_1)$ you derived to write the first order optimality condition for problem (6). $\mathbf{W}$ is a function of $\mathbf{x}$ so these equations are not linear, but if you fix $\mathbf{W}$ (i.e., if $\mathbf{W}$ is set constant), these equations can be interpreted as the normal equations for a weighted least squares problem.

4. Assume that $\mathbf{W}$ is fixed and does not depend on $\mathbf{x}$. Write a (weighted) least squares problem, similar to (4), whose first order optimality condition is given by the normal equations that you found above.

Q15 (code). Use IRLS to solve problem (6) with $\alpha = 1/2$, and $\mathbf{L}$ the finite difference approximation of the gradient operator as before. Initialize with the best solution you got from the previous section (does it matter?). In the beginning of iteration $k+1$, set $\mathbf{W} = \mathbf{W}(\mathbf{x}_k)$ and solve the weighted linear least squares problem you got with the CGLS algorithm, to get the solution $\mathbf{x}_{k+1}$. Repeat until convergence with an appropriate stopping criterion, or until a maximum number of iterations has reached. Present the output and the objective value after each IRLS iteration. Compare qualitatively the solution you got to problem (6) with the solution you got to problem (3).

Tip: the best way to observe the different quality of the solutions is to use the `slice` function to plot several slices. Also look at the histograms of the reconstructed density fields.

Q16 (code). Run your code on the large bag (`Large.zip`). What did you find in the bag? Display the output.

5 Tips

- You may find the following Matlab commands useful: `patch`, `slice`, `sparse`, `spdiags`
- You can download the file `displayVolumeSlicegUI.m` from the website to display a volume in slices.
- If you are having a hard time with the implementation of CGLS, try using `pcg` or `lsqr` to see that you are on the right track. Switch that with your code later on.