Math 440 – Computational Inverse Problems Fall 2017

Instructor: Prof. Erkki Somersalo

Exercise 3

1. Image deblurring, or image deconvolution, is a classical inverse problem. The goal is to restore an image as sharp as possible based on a noisy, off-focus version of it.

   Mathematically, we define the original image defined as a gray scale function over a rectangle $Q \subset \mathbb{R}^2$,
   $$f : Q \to \mathbb{R}_+,$$
   where $f(p)$ is the gray scale value of the image at point $p \in Q$. We assume that the blurred image g is obtained by convolving the original image with a blurring kernel
   $$g(q) = \int_Q K(q-p)f(p)dp, \quad q \in Q.$$
   The function K is called the *point spread function* and its characteristics determine how near or far the image g is from the original f.

   We begin by discretizing the image support Q into $N = n_1 \times n_2$ identical pixels of area w_N:
   $$Q = \bigcup_{j=1}^{N} Q_j, \quad |Q_j| = w_N = \frac{|Q|}{N},$$
   where $|Q_j|$ and $|Q|$ denote the areas of the pixel Q_j and the original image Q, respectively. Let $p^{(j)}$ denote the center point of Q_j and approximate the blurred image by
   $$g(p^{(j)}) \approx w_N \sum_{k=1}^{N} K(p^{(j)} - p^{(k)})f(p^{(k)}), \quad 1 \le j \le N,$$
   where we tacitly assume that no contribution to the blurred image from outside the field of view (FOV) Q needs to be taken into account. Mathematically, this is tantamount to saying that the function f extends as zero outside Q. Denoting
   $$a_{jk} = w_N K(p^{(j)} - p^{(k)}), \quad x_k = f(p^{(k)}),$$
   and letting b_j be the noisy observation of $g(p^{(j)})$, we arrive at the linear discrete observation model
   $$b = \mathsf{A}x + \eta.$$
   With this notation, the *deconvolution problem* can be stated as follows: *Estimate the original discretized image x from the noisy observation b.*

   (a) Download the original image, a 100×100 matrix called `TrueTarget.mat` given in the Canvas page, and plot the image. This is a square image, so make sure your image is square. You can use the `imagesc` command for plotting.

   (b) Construction of the convolution matrix: Typically, the convolution kernel has the property
   $$K(p-q) \approx 0, \text{ if } |p-q| > r \text{ for some } r > 0,$$
   which means that the corresponding matrix A contains a lot of zeros. The sparsity of A is a great computational advantage. In this example, we are going to use a truncated Gaussian blurring kernel. Define
   $$K(p-q) = \exp\left(-\frac{1}{2\lambda^2}\|p-q\|^2\right),$$

where $\lambda > 0$ is a parameter controlling the width of the kernel. Without loss of generality, we may assume that the points $p^{(j)}$ are integer lattice points, and $|Q_j| = 1$. To compute the matrix, here are some guidelines:

```
n1 = 100;
n2 = 100;
N = n1*n2; $ Number of pixels
p1 = (1:n1);
p2 = (1:n2);
[P1,P2] = meshgrid(p1,p2);
P = [P2(:)';P1(:)']; % First row = index to the matrix row, second to column
```

Define the width of the convolution kernel: use, e.g., the value $\lambda = 3$. Then define the convolution matrix, which is a $N \times N = 10\,000 \times 10\,000$ matrix. For each pair of columns $P(\,:\,, j)$ and $P(\,:\,, k)$, you need to calculate the entry

$$a_{jk} = \exp\left(-\frac{1}{2\lambda^2}\|P(\,:\,, j) - P(\,:\,, k)\|^2\right).$$

This leads to a full matrix, but lots of the entries are numerically zeros. Therefore, a truncation is recommended. You can do as follows: Calculate the entry a_{jk} only if the distance $\|P(\,:\,, j) - P(\,:\,, k)\|$ is below a given threshold,

$$\|P(\,:\,, j) - P(\,:\,, k)\| < \tau.$$

For instance, you can define τ by solving it from the condition

$$e^{-\tau^2/2\lambda^2} = 10^{-3},$$

meaning that you discard all entries of A that have a value below 10^{-3}. Define A as a sparse matrix, using the Sparse matrix structures in Matlab. Using `imagesc` or other visualization tools of Matlab, visualize the point spread function to give an idea of how strong the blurring is. Also, using the command `spy`, visualize the sparsity structure of the matrix A. What is the fill-in ratio of A, that is, percentage of non-zero entries?

(c) Apply your blurring matrix A on the original image, and plot the blurred image.

(d) Try to reconstruct the original image using Tikhonov regularization,

$$x_\alpha = \text{argmin}\{\|b - \mathsf{A}x\|^2 + \alpha\|x\|^2\}.$$

To get a sense of the effect of the regularization parameter, start with the noiseless blurred image, and compute and plot the reconstruction with $\alpha = 10^{-6}$. Then add low level normally distributed noise to the blurred image,

```
noise = 1e-3*randn(N,1);
```

and compute and plot the reconstruction with the same regularization parameter value.

(e) With the noise vector defined above, find a regularization parameter using the Morozov discrepancy principle. Plot the resulting reconstruction.

Notice that in this example, we are considering only the inverse crime reconstructions, that is, we assume that the point spread function is exactly known, and the data were generated with the same matrix that we used for the image reconstruction. An alternative approach can be based on Fourier analysis that will be discussed later.