Computational aspects of Inverse Problems — Extensive Exercise 3
Helsinki University of Technology, January 2009
Samuli Siltanen, Lauri Harhanen and Juho Linna

## 1. Image deconvolution and Tikhonov regularization

Download and study the files `oblur.m`, `ex_conv_2D.m` and `ex_conv_2D_SVD.m`. Routine `ex_conv_2D.m` simulates blurred image data (with inverse crime, unfortunately, but do not mind about that now), and routine `ex_conv_2D_SVD.m` solves the inverse problem using truncated SVD.

**Exercise 1.1.** *Working with images of size* $32 \times 32$*, find out the number of singular values giving the smallest relative error defined by*

$$\frac{\|\text{original} - \text{reconstruction}\|_2}{\|\text{original}\|_2}.$$

**Exercise 1.2.** *Increase the image size gradually and find out how for big images you can perform truncated* SVD *reconstruction before your computer memory is too small for the computation.*

Download and study the file `ex_conv_2D_Tikhonov_CG.m`. There, noisy blurred data is simulated (with inverse crime), and Tikhonov regularization is computed by using the iterative Conjugate Gradient (CG) solution method for the normal equations. Information about CG and the preconditioning method used is available for example in Wikipedia. Just to demonstrate the power of matrix-free computation, perform the computations below with significantly larger image size than the maximal size suitable for SVD found in Exercise 1.2.

**Exercise 1.3.** *Remove the inverse crime from* `ex_conv_2D_Tikhonov_CG.m` *by simulating blurring on a higher resolution image and interpolating the result to coarser level for inversion. Note that the point spread function (*PSF*) is defined as an explicit formula, so approximately the same shape of* PSF *can be used in computations at both resolutions.*

**Exercise 1.4.** *Assume that the image size is* $n_0 \times n_0$*; then our unknown* $x$ *is modelled as a vector in* $\mathbb{R}^n$ *with* $n = n_0^2$*. Let* $A$ *denote the* $n \times n$ *matrix of discrete two-dimensional convolution as explained in Section 2.2.2 of the lecture notes. Show that is the continuum* PSF *has rotational symmetry (*$\psi(t) = \psi(|t|)$*), then* $A^T = A$*.*

**Exercise 1.5.** *How is the fact* $A^T = A$ *utilized in the matrix-free iterative computation in* `ex_conv_2D_Tikhonov_CG.m`*?*

**Exercise 1.6.** *Use the L-curve method (if possible) to determine optimal value for the regularization parameter. What is the relative error of that reconstruction? If the L-curve method does not work, find out by experimenting the regularization parameter giving the smallest relative error.*

## 2. Statistical inversion for 1-D deconvolution

Download and study the file `conv1D_07_stackedform.m`.

**Exercise 2.1.** *Take the dimension of the problem to be $N = 32$ (modify line 43 of routine* `conv1D_07_stackedform.m`*). Choose such a value of the regularization parameter $\delta > 0$ that both bumps in the signal are recovered approximately, and use the same value $\delta$ throughout the rest of the exercise.*

As was shown in the lectures, the Tikhonov regularized solution computed in Exercise 2.1 coincides with the MAP estimate for the posterior distribution

$$p_{X|M}(x|m) = C \underbrace{\exp(-\alpha \|Lx\|_2^2)}_{\text{prior}} \underbrace{\exp(-\frac{1}{2\sigma^2}\|Ax - m\|_2^2)}_{\text{likelihood}} \tag{1}$$

with suitable choice of $\alpha > 0$. Here $\sigma > 0$ is the standard deviation of the independent Gaussian noise components $\varepsilon_j$.

**Exercise 2.2.** *Check from* `conv1D_07_stackedform.m` *what value of $\sigma$ you are using, and determine $\alpha$ so that the Tikhonov regularized solution and the* MAP *estimate coincide.*

**Exercise 2.3.** *Use Metropolis-Hastings algorithm to compute the conditional mean estimate for the posterior distribution (1). Take the proposal distribution to be Gaussian with fixed standard deviation $\sigma_P > 0$; in other words, construct the candidate from the current sample $x^{(\ell)}$ in the chain by adding a Gaussian random number with zero mean and standard deviation $\sigma_P$ to every component of $x^{(\ell)}$. Compute 10000 samples and keep track of how many candidates are accepted. Can you find such a value for $\sigma_P$ that the percentage of accepted candidates is between 20% and 40%?*

The approach of Exercise 2.3 becomes very slow when the dimension $N$ of the problem becomes bigger. Also, it may be quite difficult to find such a value for $\sigma_P$ that the percentage of accepted candidates is between 20% and 40%. One way to overcome those problems is described in the following exercise.

**Exercise 2.4.** *Modify the algorithm of Exercise 2.3 as follows. When constructing the candidate from the current sample $x^{(\ell)}$ in the chain, do not modify every component of $x^{(\ell)}$. Instead, choose $1 \leq n_0 \leq N$ components randomly, and add a Gaussian random number with zero mean and standard deviation $\sigma_P$ to the chosen components only. Compute 10000 samples and keep track of how many candidates are accepted. Can you find such values for $n_0$ and $\sigma_P$ that the percentage of accepted candidates is between 20% and 40%?*

**Exercise 2.5.** *Show that the transition probabilities given by the proposal distribution of Exercise 2.4 are symmetric.*

It can be shown that for Gaussian posterior distributions the MAP estimate coincides with the conditional mean estimate. So the result of Exercise 2.4 should be the same than the result of Exercise 2.1, provided that $\alpha$, $\delta$ and $\sigma$ are related as in Exercise 2.2.

**Exercise 2.6.** *Find out (by numerical experiments) how many samples you have to draw to make the relative difference between results from Exercises 2.1 and 2.4 less than 10%.*

## 3. Image deconvolution with approximate TV regularization

Here we study total variation regularization using approximate absolute value function and derivative-based (Barzilai-Borwein) minimization. This approach is suitable for very large scale problems.

Download and study the files `deblur_BarzilaiBorwein.m, deblur_aTV_fgrad.m, deblur_misfit_grad.m, deblur_aTV.m, deblur_aTV_grad.m, deblur_aTV_feval.m deblur_misfit.m`. The main file `deblur_BarzilaiBorwein.m` will not just run in Matlab as it is, you need take the data simulation step from exercise set 1 and include it here. There may be a need for further debugging as well; consider it as part of the exercise.

**Exercise 3.1.** *Implement Barzilai-Borwein minimization by modifying the above files. Can you get a smaller relative error than with the Tikhonov approach in exercise set 1?*