Math 76 Homework 2: Due July 14 at the beginning of class

## Tools needed for homework

This is *modified* from the original posted homework. Things you could use for this assignment:

1. There is some helpful code and information at `http://www.cs.umd.edu/~oleary/SCCS/cs_deblur/index.html`. I encourage you to download the sample solution code to help you get started. You can also use the sample solution code to compare against your own results. However, you may not simply copy this code to complete the homework assignment.

2. **MATLAB CVX**: `http://web.cvxr.com/cvx/doc/quickstart.html#an-optimal-trade-off-curve` and `http://web.cvxr.com/cvx/doc/CVX.pdf`. There are a lot of ways to use CVX. One of the goals in this assignment is to see the effect of different regularization parameters $\lambda$, so make sure you choose the CVX options that allow you to vary this. You will also want to vary the number of iterations and error tolerance.

- You do **not** need to program your own optimization method from scratch. If you are interested in what is going on "under the hood" you can check out the material in the convex optimization section of your textbook.

- You may certainly work with each other on code development, but each student must submit his/her own homework and do his/her own analysis of the results.

- Do not use very large data sets, as the cvx program is not well suited for this. Eventually if you want to use a larger image, you are free to find a better software package. CVX will be fine for a one-dimensional problem.

## Problem set up

As described in **Challenge 6.5** in the O'Leary et. al. textbook, your goal is to write a program that takes matrices $\mathbf{A}$ and $\mathbf{B}$ and an image $\mathbf{G}$ and computes an image $\mathbf{F}$ using different regularization techniques. We are going to simplify this problem and consider $\mathbf{G} = \mathbf{g}$, where $\mathbf{g} \in \mathbb{R}^{m\times 1}$, that is $\mathbf{g}$ is an $m$ vector. The solution we seek is a function $f$ which we will calculate at a set of $n$ grid points.

Let $f$ be a piecewise continuous function on the interval $[0, 1]$. (You can use any interval $[a, b]$ you like.) For example choose $f$ to be piecewise constant, piecewise linear, piecewise polynomial, or a piecewise trigonometric polynomial. There should be a few points of discontinuity in the function. Use your imagination. Suppose we have data $\mathbf{g}$ which is obtained by blurring $f$. That is, we have $g(x) = (K * f)(x)$, which we put on a set of $m$ grid points $x_j = a + (b - a)(j - 1)/(m - 1)$, $j = 1, \cdots, m$. You can choose different blurring functions and you should choose different choices for $m$, say $16, 32, 64, 128$ and $m = \alpha n$, where $\alpha > 1$ for an overdetermined system and $\alpha < 1$ for an underdetermined system.

Now we consider the following regularizations:

1. Tikhonov regularization,

2. truncated SVD,

3. $\ell_1$ regularization.

In all cases, you can write the solution to the problem as

$$\mathbf{f}^* = argmin_f(||K\mathbf{f} - \mathbf{g}||_2^2 + \lambda||L\mathbf{f}||_p^p),$$

where we choose either $p = 1$ or $p = 2$. The solution $\mathbf{f}^*$ is the approximation of $f$ at grid points $x_l$, $l = 1, \cdots, n$ on $[a, b]$.

Depending on the underlying function (signal), $L$ could be either $I$ (as in the Tikhonov method described in the book) or it could be the first derivative operator (this is related to *total variation* (TV), which is especially useful in images that take on piecewise constant appearance). You should also consider the following scenarios:

- The system is ill conditioned: The blurring matrix is $n \times n$ but has at least one singular value that is close to zero. This is consistent with blurring.
- The system has noise: Play with various noise levels and instead of $\mathbf{g}$, choose $\mathbf{g} + \eta$, where $\eta$ is a vector of white noise.
- The system is underdetermined: In this case we have fewer measurements of $\mathbf{g}$ then needed, so we need the regularization to choose the best solution. One way to program this is to generate a *row selector matrix* $\mathbf{M}$ that zeros out $n - m$ rows (where $n$ is the number of columns and $m$ is the number of rows). See how little data you can get away with. Note that $\mathbf{M}$ does not need to have all of its zero rows at the bottom. You are more interested in doing some kind of sparse sampling throughout the input data domain.

Analyze your results. You should prepare a well written report, with a clear explanation of how you set the problem up, what paramters you chose, and how you varied your experiments. You should use pictures to illustrate your comments. Your write up *must* do the following:

1. General comparison of different types of regularization for these problems. When does one regularization appear to be superior to the others?
2. Discuss how robust each method is with respect to the regularization parameter $\lambda$. That is, do the results vary wildly for different choices of regularization parameters?
3. Discuss how much data are needed in order to achieve meaningful results.
4. Discuss how each method responds to added noise and/or blur.
5. Discuss the efficiency of the method. Note that this is a bit out of your control because you are using a canned software package.

**Shameless plug:** These problems are regularly solved in imaging and signal processing. The methods are becoming part of the hardware in MRI and other types of medical imaging devices, and are becoming more common in areas such as astronomy and seismology. Projects 1 & 2 on the Math 76 website take into account other issues, such as cross registration of data.