CSC/MTH 753 Mid Term Exam Spring 2013

Due by Thursday March 7

Please do your own work, individually, without private help from others, or me. You should ask questions directly related to the **problems only in class, so everyone can hear my responses**. That would be fair to your classmates.

- Some of the problems may relate material in the supplementary text, "Convex Optimization", Stephen Boyd and Lieven Vandenberghe, available free online at

  http://www.stanford.edu/~boyd/cvxbook/.

  You might want to read the relevant parts in Chapters 1–4, and 9.

- Again, do your own work, individually, without help from others, or me, outside of class. You can use any books, papers, or internet resources.

- You might want to list any resources you use for help in solving each problem.

- Turn in a neat stapled paper, with answers clearly identified, email submissions accepted "only" from the two remotely located students.

1. Suppose that f is a convex function defined on a convex set in R^n. Show that the set of all global minimizers of f is a convex set.

2. Refer to my notes provided to you earlier by email: **An Overview of Unconstrained Optimization**. Suppose $A \in R^{m\times n}$ has full column rank n. Consider the penalized "nonlinear" least squares problem to minimize over all x:

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2 + \frac{1}{2}\lambda\|x\|_2^2.$$

   Here λ is a positive scalar, normally $0 \leq \lambda < 1$, that determines a balance between the first term, fit-to-data, and the smoothness penalty in the second term. λ is called a regularization parameter. It "preconditions" the problem.

   (a) Give the the gradient, and the Hessian for f.

   (b) Is f convex? Why?

(c) Show that this problem of finding the minimizer x can be converted to a linear least squares problem, by considering the symmetric positive definite normal equations obtained from setting the gradient to the zero vector.

(d) Then construct a simple example as follows. Set $H = hilb(15);$, $A = H(:, 1:13);$ and $b = A * ones(12, 1);$. Thus the vector of all $1's$ solves the optimization problem. Use the Matlab command **cond** to find the condition numbers of H and A. Finally, use any method you choose, "backslash" in Matlab, **chol**, etc., to solve the normal equations with $\lambda = 0$, and then with some experimental positive λ between 0 and 1. Can you get a better approximate solution by a "good" choice of λ? Explain your results.

3. As discussed and shown by demo in class, consider the Rosenbrock (banana) function

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

Below, you will be considering the "banana function" demo and Example 1 in the Matlab description of fminsearch.

(a) Use fmincunc to approximate the minimizer $(1, 1)$. Use optimset. Hint, define f either in an m-file or inline statement, or function handle, and run, as done in class,
$options = optimset('LargeScale', 'off', 'Display', 'iter', 'TolFun',$1e-8);
$x_0 = [0\ 2];$
Depending on how you set up your m-file, you might use
$[x, fval, exitflag, output, grad, hessian] = fminunc(f, x_0, options)$
Use optimtool if you like, but display your iterations.

Try various other starting vectors x0. Why is this problem hard?

(b) Now, define f and the gradient g in an m-file and run fminunc using your gradient. See examples of how to do this in the Optimization Toolbox.

(c) Repeat (a) above using fminsearch. Explain how fminsearch solves these problems, that is describe the algorithms that are used. When would it be advantageous to use fminsearch?

4. As mentioned in class, consider the "Extended" Rosenbrock Function

$$f(x) = \sum_{i=1}^{2k} [\alpha(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2],$$

where α is a parameter you can vary, say $\alpha = 1$ or 100. Repeat 2 (a) above for this function. (Use doc for in Matlab to see how to code f.) Set k = 200, use a starting point $(-1, -1, \cdots, -1)$ and observe the behavior of fminunc. Of course you will **not** set 'LargeScale','off'. Explain how fminunc solves this large-scale problem, by what algorithms, etc. Don't print your solutions for x, too large.