MATH 171B: Numerical Optimization: Nonlinear Problems

Instructor: Michael Holst

Spring Quarter 2015

Homework Assignment #4
Due (See Class Webpage for Due Date)

This homework finishes off the unconstrained optimization part of the course. We will practice some techniques for modifying the Hessian in the Newton iteration when it becomes necessary to do so. You should read pages 94-98 in Professor Gill's notes on this material carefully before you do the homework. The starred exercises are those that require the use of MATLAB. *You must do the* MATLAB *problems to get credit for the homework.*

Exercise 4.1. Let $f(x)$ denote a convex continuously differentiable function. Show that if a stationary point x^* exists, then $f(x^*)$ is a global minimum of f. Also show that if $f(x)$ is actually strictly convex, then x^* is the unique global minimum. Why can uniqueness be lost if the function is not strictly convex? Draw a picture of such a situation when $f : \mathbb{R} \mapsto \mathbb{R}$.

Hint: Use the result that $f(x)$ is convex if and only if $f(y) \geq f(x) + f'(x)(y - x)$ for all x and y, and the result that $f(x)$ is strictly convex if and only if the inequality holds strictly.

Exercise 4.2. This problem requires modifying the Hessian to produce a descent direction. Consider the function $f : \mathbb{R}^3 \to \mathbb{R}$ such that

$$f(x) = x_1^2 + x_2^2 \cos x_3 - e^{x_2} x_3^2 + 4x_3.$$

(a) Derive the gradient $g(x)$ and Hessian $H(x)$ of $f(x)$.

(b) Compute the *spectral decomposition* of $H(x)$ at $\bar{x} = (0,\, 1,\, 0)^T$.

(c) Compute the "pure" Newton direction p^N at $\bar{x}$. Is p^N a descent direction?

(d) Compute a *modified* Newton direction p^M at the same point using the eigenvalue relection technique (the better of the two approaches we discussed in class). Find the directional derivative along p^M at $\bar{x}$.

(e) Find a *direction of negative curvature* at $\bar{x}$. Verify your result numerically.

Exercise 4.3.* Write a MATLAB function `newton.m` that implements a modified Newton with a backtracking linesearch. (This is a fairly simple modification of the routine `steepest.m` that you wrote for the previous homework.) Your function *must* include the following features:

- The specification of `newton.m` must be of the form:

```
function [x,fx,gx] = newton( fname,x0,tol )
% Purpose:  newton(fname,x0,tol) finds a local minimizer of a nonlinear
% function f(x).  fname is a string containing the name of an m-file that
% computes the value of f(x) for a given value of the variables x.  x0 is a
% starting guess, tol is the target Euclidean norm of the gradient of f.  For
% example, newton('my_function',x0,1.0e-5) a function defined in the m-file
% my_function.m.
```

- Use $\mu = \frac{1}{4}$ to define the sufficient-decrease criterion in the backtracking algorithm.
- The minimization must be terminated when either $\|g(x_k)\| \leq 10^{-5}$ or 50 iterations are performed. Any MATLAB "`while`" loop must include a test that will terminate the loop if it is executed more than 20 times.

Now, do the following with the implementation:

(a) Starting at $x_0 = (0,\, -1)^T$, apply the modified Newton method to Rosenbrock's function

$$f(x) = 100\big(x_2 - x_1^2\big)^2 + \big(1 - x_1\big)^2,$$

which has a unique minimizer at $x^* = (1,\, 1)^T$.

(b) Minimize the function

$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4,$$

starting at $x_0 = (\frac{1}{2},\, \frac{1}{2},\, \frac{1}{2},\, \frac{1}{2})^T$. The minimizer lies at $x^* = 0$. Discuss the differences between this run and that of part **(a)**.

In each case, verify that the point you find is a local minimizer.