Homework 4

CS 726, Semester I, 2011–12

October 14, 2011

The matlab coding of this assignment should be submitted electronically using the instructions on the course web page. The remaining questions and comments should be handed in during class on October 28. The assignment name is hwk4 and you should electronically hand in two files with the following names: BFGS.m, TNewton.m

We provide a mechanism for accessing nonlinear test problems written in the AMPL modeling language for the remainder of the course. This enables you to avoid writing any more function, gradient and Hessian files.

To use this, you must be running Matlab on Linux on a CS machine (e.g mumble-XX). To ensure that you are working on a linux machine, use the command "hostinfo". If you don't have a matlab directory or a startup.m file, you need to create both of these. To do this, at the UNIX prompt:

```
cd ~
mkdir matlab
cd matlab
cp ~cs726-1/public/cute-start.m startup.m
```

If you already have a startup.m file in the directory ∼/matlab, simply add the following line to that file.

```
path('/p/course/cs726-ferris/public',path);
```

Then logout and log back in again.

To run a particular problem, you first need to copy the file "NAME.nl" to the directory in which you are working, and then can use:

```
[x,l,u,v,cl,cu] = spamfunc('NAME');
```

which sets up files for the problem 'NAME' and generates x as a given starting point; follow this by the use of the "objampl" function. For example, the commands:

```
[x,l,u,v,cl,cu] = spamfunc('rosenbr');
xstart = struct('p',x);
nparams = struct('maxit',1000,'toler',1.0e-4,'method','chol');
[inform,xsol] = Newton(@objampl,xstart,nparams);
```

should find a solution in 25 steps or less.

Note that the "nl" files for each problem are available from the course website. The ampl files that generated these can also be viewed at the same location.

1. Consider the problem of minimizing (locally) the function

$$f(x,y) = (1/2)\{p(x^2+y^2-2x-2y)+(xy-1)^2\},$$

where x and y are real numbers and p is a real parameter. Answer the following questions, justifying your answers. You may use known theorems, but if you do then you must clearly describe them.

   (a) What are values x_0 and y_0 such that f has a stationary point at (x_0, y_0) for *every* value of p?

   (b) For which value(s) of p does (x_0, y_0) satisfy the second-order necessary condition?

   (c) For which value(s) of p does (x_0, y_0) satisfy the second-order sufficient condition?

   (d) For which values(s) of p can you be certain that f is convex in a neighborhood of (x_0, y_0)?

   (e) For which value(s) of p can you be certain that Newton's method, if started sufficiently close to (x_0, y_0), will converge quadratically to (x_0, y_0)?

2. Let $f\colon \mathbf{R}^n \to \mathbf{R}$ be the quadratic function $f(x) = \frac{1}{2}x^TAx - b^Tx + c$, where $A \in \mathbf{R}^{n\times n}$ is symmetric and positive definite. For arbitrary $x_0 \in \mathbf{R}^n$ and linearly independent vectors $p_0, \ldots, p_{k-1}$, let

$$V_k = \left\{ x \in \mathbf{R}^n \mid x = x_0 + \sum_{i=0}^{k-1} \alpha_i p_i, \forall \alpha_i \in \mathbf{R} \right\}$$

   Show that the unique minimizer of the restriction of f to V_k is given by

$$x_0 + P_k(P_k^T A P_k)^{-1} P_k^T (b - Ax_0)$$

   where P_k is the matrix with columns $p_0, \ldots, p_{k-1}$.

   Suppose the conjugate gradient method is applied to minimize f, that is

$$x_{k+1} = x_k - \alpha_k p_k, k = 1, 2, \ldots, n$$

   where

$$\alpha_k = \frac{(Ax_k - b)^T p_k}{(Ap_k)^T p_k}$$

   and $p_0, \ldots, p_{n-1}$ are A–conjugate, that is $(Ap_i)^T p_j = 1$ if $i = j$ and 0 otherwise. Show that

$$f(x_k) = \min\{f(x) \mid x \in V_k\}, k = 1, 2, \ldots, n$$

3. Implement the BFGS method outlined in class. Use the stopping criterion of

$$\frac{\|\nabla f(x)\|}{\min\{1000, 1 + |f(x)|\}} < 10^{-4}$$

   for this (and subsequent) homeworks.

   Experiment on the examples from previous homeworks with this code. You should update the initial approximation of the inverse Hessian as detailed in class using

$$H_0 = \gamma I, \quad \text{where } \gamma = \frac{s_0^T y_0}{y_0^T y_0}.$$

   Test this out on the AMPL problems `brownal`, `watson`, `geodesic`, `cragglvy` and `woods`. Details on running this are in the script file `hwk4.m`.

4. Implement the Truncated Newton method outlined in the course text and during lectures.

   Use $\eta_k = \min(0.5, \sqrt{\|\nabla f(x_k)\|}$ (multiplied by $\|\nabla f(x_k)\|$), the other constants as noted in class or in the text. Test your routine on all four examples using the recommended starting points. Make sure you update the global variables that compute the total number of function, gradient and Hessian evaluations that you use.

5. The following problem:

$$\min f(x) = \sum_{i=1}^{n} \left[(1 - x_{2i-1})^2 + 10 * (x_{2*i} - x_{2i-1}^2)^2\right] + \frac{\gamma}{2}(\sum_{i=1}^{2n} x_i - 1)^2$$

   with $\gamma = 100$ is available on the web as `objg.m`. Test your Newton, BFGS and TNewton codes on the problem using of `hwk4.m`. Comment on the results; (allow the codes to abort if more than 10 minutes have passed without solution). Can you explain the relative speeds of each? You should notice that in this example that with $n = 600$, this code works rather slowly. The purpose of this assignment is to rectify this, and solve the same problem for $n = 1000$, $n = 2000$ and $n = 4000$.

   Comment on what methods you would choose for large scale unconstrained optimization and justify your choices explicitly in your solution. You may wish to experiment also with the geodesic problem from previous homeworks so your comments are not just related to these problems.